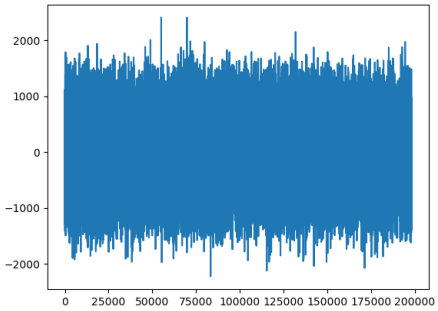
HEXANE WATER RESULTS LOG SUMMER 2018 (AUGUST-PRESENT)

KIRK SWANSON

1. 8/6/2018

(1) Downloaded pressure_scalar-press-NPT-1bead-water-lammps to dash_work/interface on laptop. Used pressure_analysis_UPDATE.py leaving out first 2000 records and used 198 blocks 5 ps to get average pressure of 1.41, SE 0.94, SD 511.08:



(2) Because we accidentally used restart/traj every 1,000 for fulldata-NPT-298 and then restart/traj 10,000 for fulldata-NPT-298-restart1, we will re-do this calculation again using the updated interface file collecting restart/traj every 1,000.
(3) Submitted simulation no. 1.
(4) To check pressures in a non-vacuum interfacial system, in DASH we are going to run simulation no. 2, which does NPT equilibration for 200,000 steps and then does an NVT simulation of the interface. Filename press-nonexpanded.
(5) Submitted simulation no. 2.
(6) We want to compare the above simulation to LAMMPS, so we will first run the equilibration step in LAMMPS as simulation no. 3. Note, however, that the mixing rules currently used in the LAMMPS file are arithmetic. Also NOTE: to use depablo-tc, must type module load openmpi into terminal first, even if it is already in submit script. After this, we need to run the NVT portion of the simulation for 6M steps.
(7) Running simulation no. 3.
(8) Now, let's turn to how pressure is typically computed in DASH. In general, an ideal gas mechanical equation of state is given by

$$P = \frac{Nk_BT}{V} = \rho k_B T \tag{1.1}$$

Date: October 4, 2018.

This assumes that molecules in the gas are point-like and do not interact with each other. However, if interactions between the particles and a finite particle volume is allowed. The virial expansion is a perturbation theory around the ideal gas used when the interactions in the real gas are dominated by two-body interactions. One uses a virial expansion in powers of the density, keeping the temperature dependence of the coefficients. We can then apply manipulations of the time derivative of the classical virial to arrive at

$$P = \rho k_B T - \left[\frac{1}{3V} \left\langle \sum_{i<j}^{N} r_{ij} F_{ij} \right\rangle \right] \tag{1.2}$$

where the second term is called the virial correction.

(9) In DASH, it appears as though DataComputerPressure.cu is responsible for scalar pressure values. On line 92, we have

$$pressureScalar = (tempScalar\_loc * ndf\_loc * boltz + sumVirial)/(dim * volume) * state- > units \tag{1.3}$$

tempScalar_loc appears to be the temperature of the system, boltz is the boltzmann coefficient, sumVirial is that second virial correction, dim is the dimension of the system. ndf I calculated to be 3 per atom, including M sites for the water model (printed values using interface file). So, it does make sense to divide by dim, i.e. divide by 3, because that gets us to N. However, we are also including M-sites in the pressure value. So, we are at the very least overcounting the ideal gas term for the pressure. The amount of overcorrection for 3650 water molecules is given by:

$$P_c = \frac{Nk_BT}{V} = \frac{3650 \times 1.38 \times 10^{-23} \times 298}{50 \times 10^{-10} \times 50 \times 10^{-10} \times 150 \times 10^{-10}} \tag{1.4}$$

Actually, would it even be incorrect to have four sites per atom? Maybe that's actually the correct ideal model here?

2. 8/7/2018

(1) First, we will begin by compiling the most recent version of DASH, which should contain Mike's edits to the TIP4P/F fix for accurate pressure computation (avoiding M-site in ideal gas term). Compiling version in folder DASH-8-7-2018.

(2) DASH pressure computations:

NVT with vacuum: run_UPDATE_8-7-2018.sh and interface_UPDATE_8-7-2018.py, z length 149.2717, x length 58.45467, y length 58.67825, 3650 TIP4P/F water molecules, 500 hexane molecules, waldman-hagler mixing between the water and hexane molecules, NVT with Andersen thermostat with parameters nu = 0.01 (a parameter describing the collision frequency of the system with the heat bath) and applying every 10 steps, 298 K, 1M steps equilibration and 1M steps production, PI false, restart/traj every 10,000, -20/+20 for z change as accounted above, filename newpress-expanded, tensor information commented out in input script. This is simulation no. 4.

We will do the same as above, except we will not expand the box by 20 A in each z direction, and we will apply NPT equilibration for 1M steps and then 1M steps of NVT. This is simulation no. 6.

(3) LAMMPS pressure computations:

NVT with vacuum: interface.sbatch and in.taffi_tip4pF_waldman, x length 149.2717, y length 58.67825, z length 58.45467, 3650 TIP4P/F water molcules, 500 hexane molecules, waldman-hagler mixing, NVT with NoseHoover, 298 K, 1M steps equilibration and 1M steps production, PI false, restart/traj every 10,000, filename newpress-expanded-lammps. We will start with 500,000 steps and restart from there, which is simulation no. 5.

Same as above, except we will not expand the box by 20 A in each z direction, and we will apply NPT equilibration for 1M steps and then 1M steps of NVT. The first 500,000 steps will be simulation no. 7.

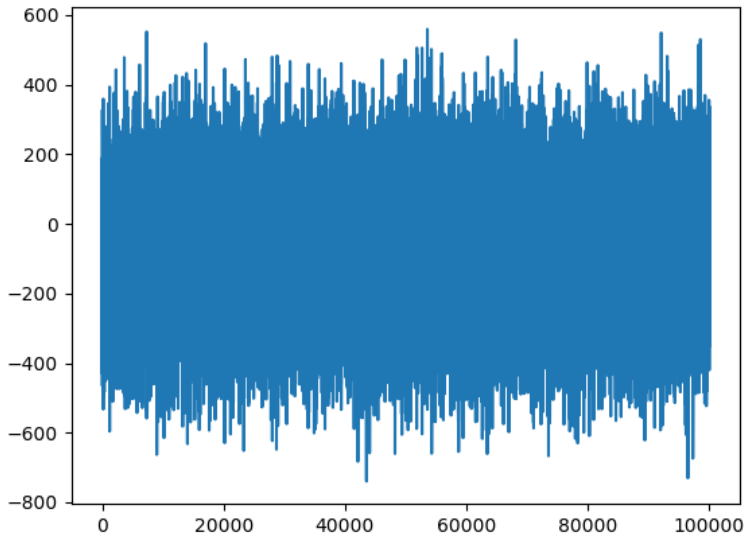
Simulation no. 8 is filename newpress-expanded-lammps-restart1, restarting from newpress-expanded-lammps.restart for 1.5M steps

Simulation no. 9 is filename newpress-nonexpanded-lammps-restart1, restarting from newpress-nonexpanded-lammps.restart for 500K steps

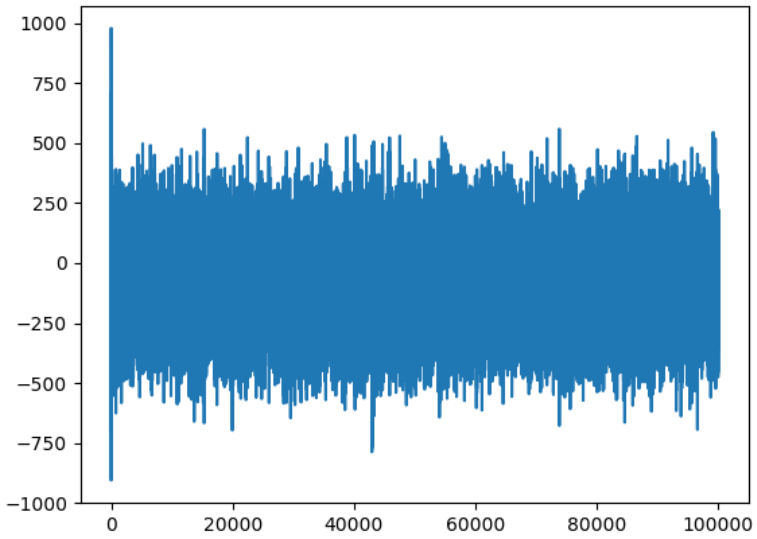
Simulation no. 10 is filename newpress-nonexpanded-lammps-restart2, restarting from newpress-nonexpanded-lammps-restart1.restart, and going for 1M NVT steps.

3. 8/8/2018

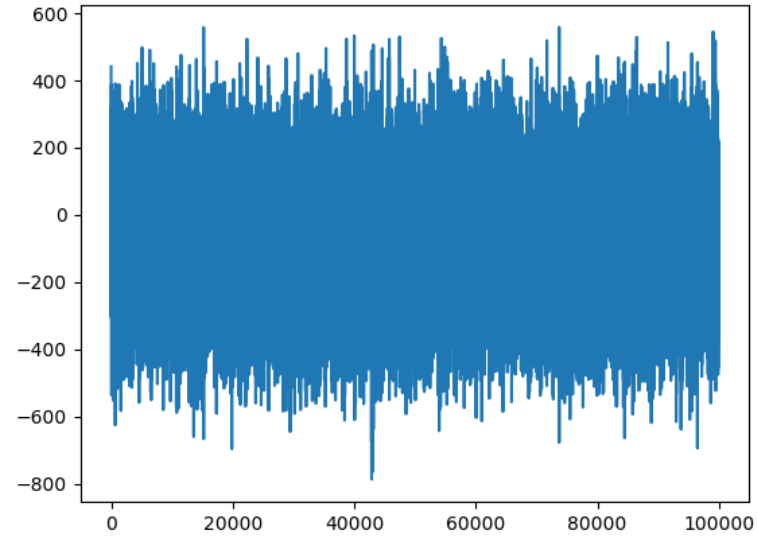
(1) From simulation no. 8, we use pressure_analysis_UPDATE.py on laptop in dash_work/interface folder to analyze pressures_scalar-newpress-expanded-lammps-restart.txt. We use indices 50,000 to 150,000 and 100 SE blocks of size 5 ps. This gives us a mean pressure of -83.443 atm, SE of the mean 1.578 atm, and SD of the distribution 154.699 atm. Here is a plot of these values over time:



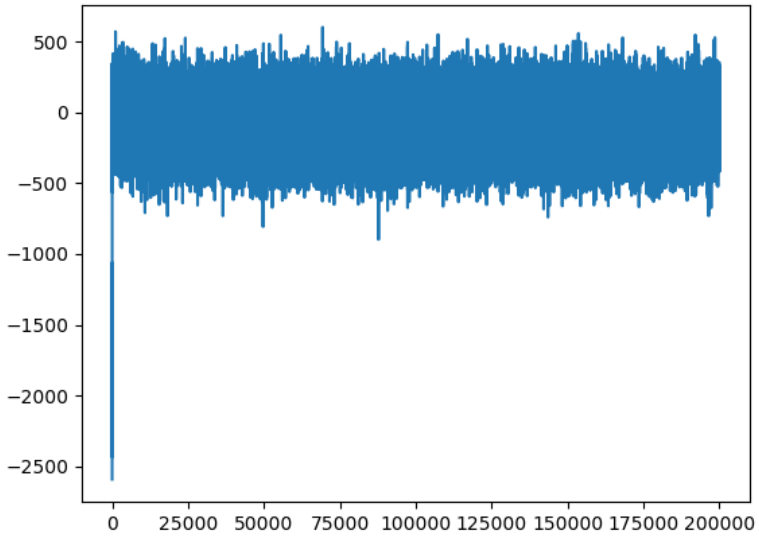
(2) From simulation no. 4, we use pressure_analysis_UPDATE.py on laptop in dash_work/interface to analyze pressure_scalar-newpress-expanded.txt. We use indices 100,000 to 200,000 and 100 SE blocks of size 5 ps. This gives us a mean pressure of -70.603 atm, SE of the mean 1.487 atm, and SD of the pressures 154.854. Here is a plot:



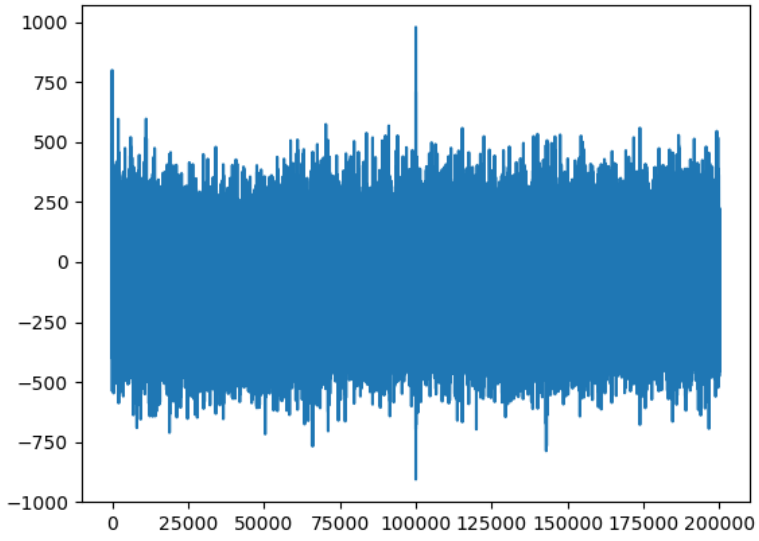
Removing the first 100 values for better comparison to the above plot, we have:



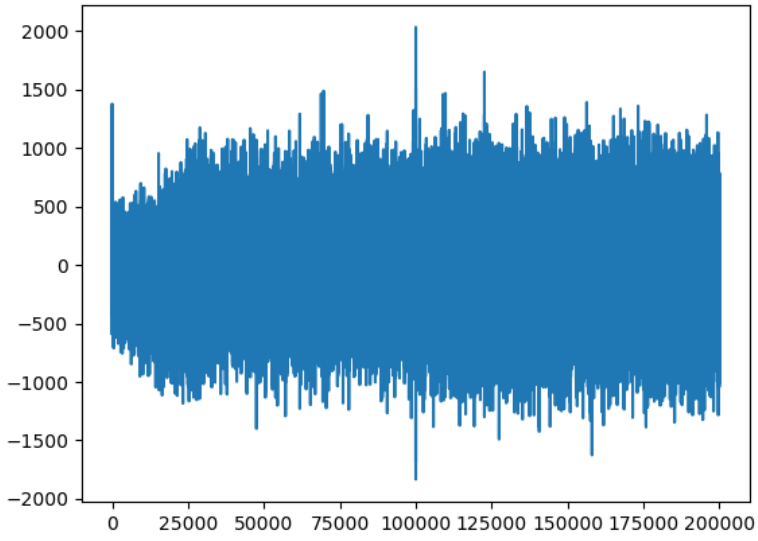
(3) If we combine pressures_scalar-newpress-expanded-lammps.txt and pressures_scalar-newpress-expanded-lammps-restart1.txt, then we have all 2M steps covered. We call this file pressures_scalar-newpress-expanded-lammps-combined.txt. Analyzing all points in this file, using 200 blocks of size 5 ps, we get an average pressure of -80.945, SE of the mean 1.237, and SD of values 156.056:



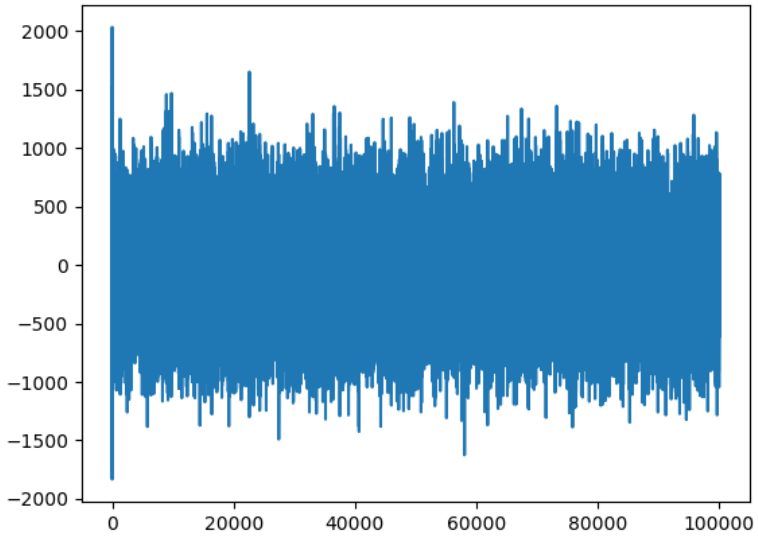
(4) Using all values from pressures_scalar-newpress-expanded.txt from DASH, using 200 blocks as well, we get a mean pressure of -77.716, SE of the mean 1.337, and SD of values 155.610:



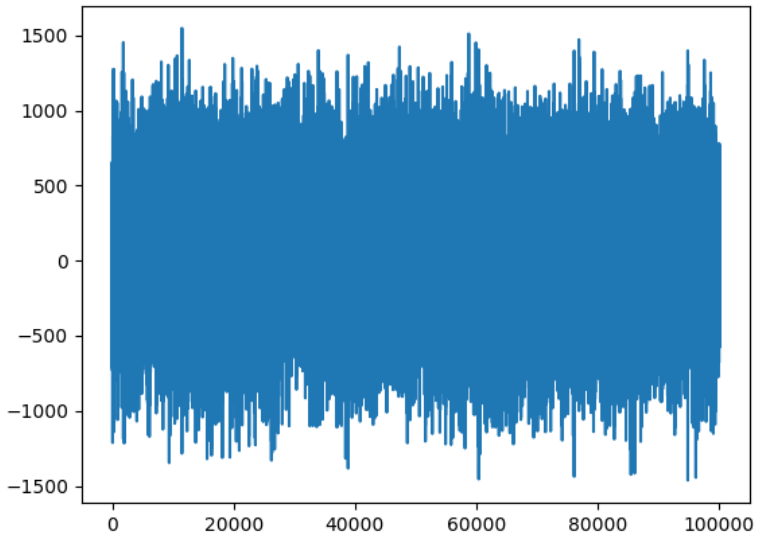
(5) Andersen Thermostat: Rescales velocities at regular timesteps to control the temperature
(6) NoseHoover Thermostat: Integrates equations of motion derived from a Hamiltonian that has an extra degree of freedom for the heat bath
(7) From simulation no. 6, in DASH, we did 1M NPT steps followed by 1M NVT steps so that we can analyze pressure from the NVT which will not have vacuum, in theory. The box dimensions we get from this simulation are: x length 48.875, y length 49.061, z length 91.3641. Downloading pressure_scalar-newpress-nonexpanded.txt to dash_work/interface on laptop for analysis as above. Using the 1M NPT plus 1M NVT data and 200 blocks, we have a mean pressure of -31.943, SE of the mean 3.174, and SD of values 333.460:



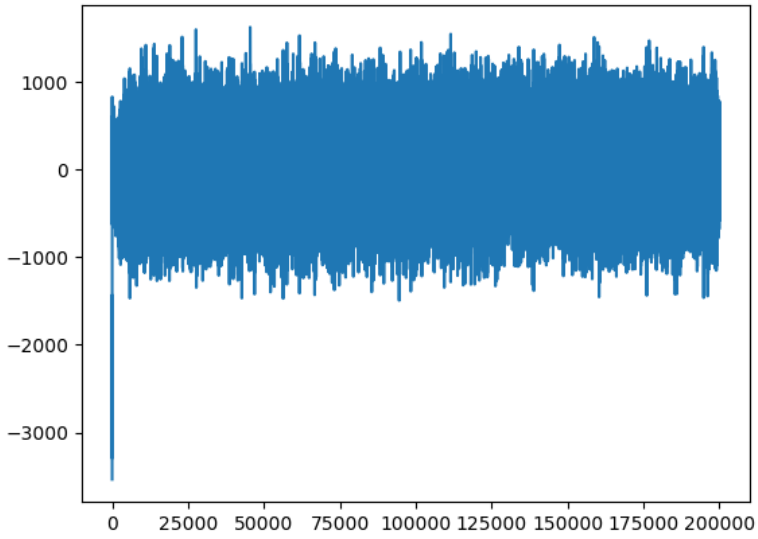
Using only the 1M NVT steps, and 100 blocks, we have a mean pressure of -32.501, SE of the mean 3.592, SD of values 364.244:



(8) From simulation no. 10, in LAMMPS, we did 1M NVT steps after 1M NPT equilibration. The box dimensions are: x length 90.498, y length 48.597, z length 48.412. These lengths are slightly different from the above DASH simulation. The NVT component of the simulation, from pressures_scalar-newpress-nonexpanded-lammps-restart2.txt, we get a mean pressure of 54.504, SE of the mean 4.47, and SD of values 375.729, using 100 blocks:

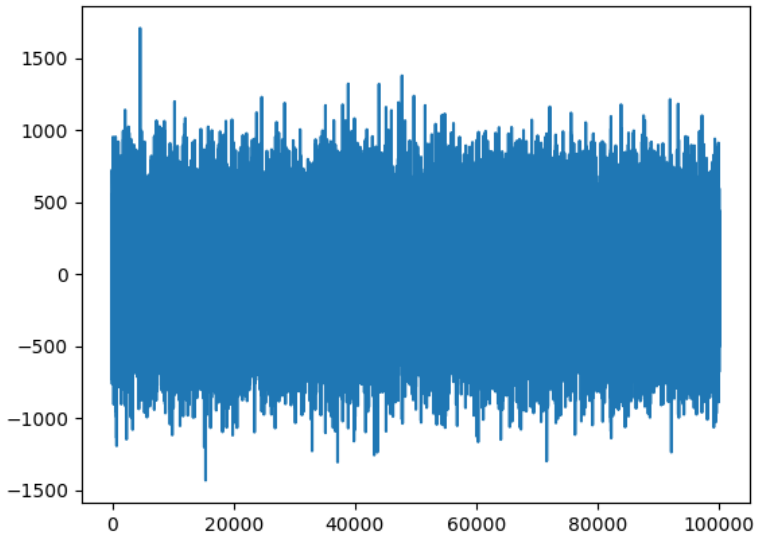


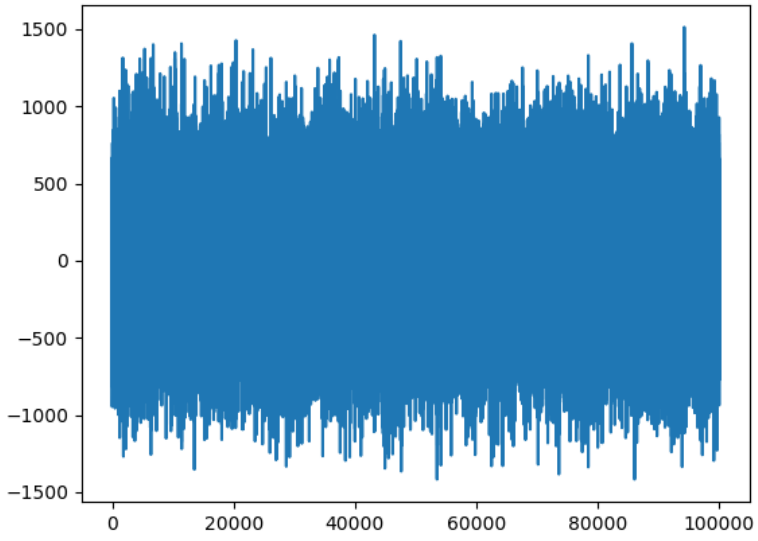
Combining the NPT steps into pressures_scalar-newpress-nonexpanded-lammps-combined.txt, we get a mean pressure of 25.947, SE 3.152, and SD 374.379 using 200 blocks:



(9) It is possible that we have differing values here because the volumes are slightly different. To test the hypothesis that the vacuum is responsible for the negative pressure, let's try decreasing the volume by only putting 10 Angstroms on each side of the simulation box.
(10) Simulation no. 11 in DASH will take care of this filename newpress-smallexpanded, and simulation no. 12 in LAMMPS will do the same.
(11) Simulation no. 13 will be DASH 2M NPT of the interface
(12) Simulation no. 14 will be LAMMPS NPT of the interface
(13) Simulation no. 15 will be DASH 2M NVT at 0.997 of TIP4P/F 3650 molecules
(14) Simulation no. 16 will be 2M NPT at 1 atm of 3650 molecules in DASH

4. 8/9/2018





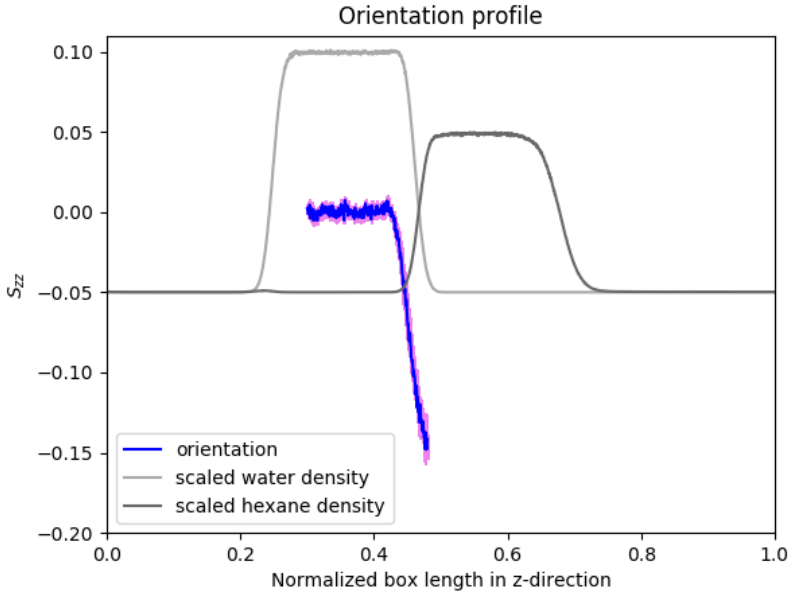
(1)

5. 8/11/2018

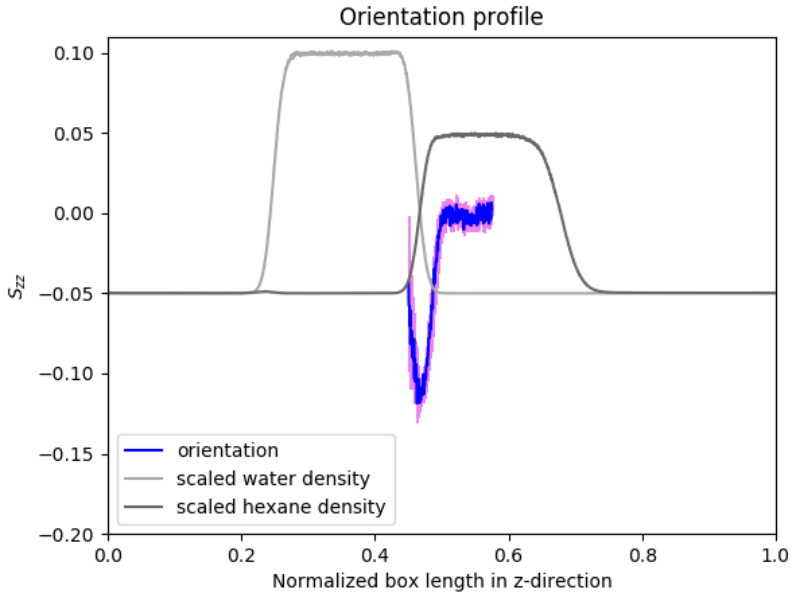
(1) First, we need to re-do molecular orientation plots. The file molecular_orientation_water_postproduc contains the unwrap update that is necessary for accurate analysis. In this file, we correctly unwrap molecules based on the location of the oxygen atom. In

dash_work/interface on Midway, we will use orientation_postproduction.sbatch to do water orientation postproduction profiles from the 1bead trajectory, full-298-1bead, and the 32 bead trajectory, full-298-32bead-combined.xyz. The filename for 1bead is orientations_water_unwrap.txt, which once downloaded to laptop was copied as orientations_water_full-298-1bead-orientations-water.txt. The filename for 32bead is orientations_water_32bead-water-unwrap.txt.

(2) For full-298-1bead, we used indices 600 to 960:



(3) We also corrected molecular_orientation_hexane_postproduction_UPDATE.py to have the appropriate unwrapping procedure. We use this in orientation_postproduction_hexane.sba to analyze the same trajectory files as above for hexane. The 1bead file is orientations_hexane_full-298-1bead-orientations-hexane-corrected.txt. We download this onto laptop and use indices 900 to 1150 to obtain:



We are also running the 32 bead version, named as above, orientations_hexane_full-298-32bead-orientations-hexane-corrected.txt.

(4) We recently resolved the negative pressure issue. Let's start from Section 2.5 of The Molecular Theory of Capillarity, by J. S. Rowlinson and B. Widom. The assumption we begin with is that it is possible to consistently define local values of the thermodynamic fields p, T, μ even in an inhomogeneous system. At a planar fluid-fluid interface, confied to a cube of sidelength l and interfacial area A = l^2, these fields and densities are functions only of the z dimension, defined as normal to the interfacial plane. We call the pressure in either bulk phase the scalar quantity p. Near the interface the pressure is a tensor since its tangential components, parallel with a dividing surface, include the tension of the interface, and so differ from the normal component. Mechanical stability requires that the gradient of this tensor is zero everywhere in the fluid, $\nabla \cdot \vec{p} = 0$, and the symmetry of the surface requires that $\vec{p}$ is a diagonal tensor, $\vec{p}(\vec{r}) = p_{xx}(\vec{r})e_xe_x + p_{yy}(\vec{r})e_ye_y + p_{zz}(\vec{r})e_ze_z$ with $p_{xx}(\vec{r}) = p_{yy}(\vec{r})$. Substitution of this equation into the gradiant equation yields $p_{zz}(z) = p_{yy}(z) = p_T(z)$ and $p_{zz}(z) = p_N(z) = p$, where p_N and p_T are the normal and transverse components of the pressure. If a side-wall of the cube is displaced isothermally and reversibly so as to increase the area by δA, then the tengential work done on the system is $\delta W_T = -\delta A \int_{-l/2}^{l/2} p_T(z)dz$. A similar displacement of the top or bottom wall by a distance $\delta A/l$ serves to maintain the volume constant, and requires further (normal) work $\delta W_N = A(\delta A/l)p = l\delta Ap$. Thus, the total work done, and hence the increase in free energy, is $\delta F = \delta W_T + \delta W_N = \delta A \int_{l/2}^{l/2}[p - p_T(z)]dz$. Using $dF = -SdT - pdV + \gamma dA + \mu dN$. So, we can associate this integral with the interfacial tension.

(5) This brings us to the following. We define the interfacial tension as $\gamma = \int_0^{L_z}[p_N(z) - p_T(z)]dz$. Since at a stable planar interface the gradiant of the pressure tensor must be zero, the tensor must be symmetric, and the bulk pressures have scalar value p, we can define $p_N(z) = p_{zz} = p$ and $p_T(z) = \frac{p_{xx}(z)+p_{yy}(z)}{2}$. Using the mean value theorem, we can write the average pressure tensor elements obtained from an MD simulation, $< p_{xx} >= p_{xx}$ and $< p_{yy} >= p_{yy}$ as $p_{xx} = \frac{1}{L_z}\int_0^{L_z} p_{xx}(z)dz$ and $p_{yy} = \frac{1}{L_z}\int_0^{L_z} p_{yy}(z)dz$. We can then carry out the integration that defines γ and rewrite the interfacial tension as $\gamma = (p - \frac{p_{xx}+p_{yy}}{2})$, which is the standard formula for computing interfacial tension in molecular simulations. In our simulation box, we have two interfaces, so the interfacial tension associated with a single interface can be given by dividing by two: $\gamma = \frac{1}{2}(p - \frac{p_{xx}+p_{yy}}{2})$. We assume that the bulk phases are at their equilibrium densities corresponding to $p \approx 1$ atm , either by measuring the bulk densities in an NVT simulation or by applying a barostat in the z dimension to maintain $p \approx 1$ atm . For hexane-water, the experimental interfacial tension is 50 $\frac{mN}{m} = 4,900$atm·Å. Our simulation box is approximately 100 Åin the z dimension, so, we have 4,900 atm · Å= (1 atm - $\frac{p_{xx}+p_{yy}}{2}$)· 100 Å. We can solve this to get $p_{xx} + p_{yy} = -194$. Since the scalar pressure in an MD simulation is computed as $P_{tot} = \frac{p_{xx}+p_{yy}+p_{zz}}{3}$, we have $P_{tot} = -64.33$. What happens to these negative scalar pressures in the thermodynamic limit? In ambient conditions, γ would still be 50 $\frac{mN}{m}$ and $p = 1$ atm , but $L_z \to \infty$, so $\lim_{L_z\to\infty}\frac{\gamma}{L_z} = 0$, which implies that $\frac{p_{xx}+p_{yy}}{2}$ converges to p and $P_{tot} \to 1$.

$$\gamma = \int_0^{L_z}[p_N(z) - p_T(z)]dz \tag{5.1}$$

$$p_N(z) = p_{zz} = p \tag{5.2}$$

$$p_T(z) = \frac{p_{xx}(z) + p_{yy}(z)}{2} \tag{5.3}$$

$$= \frac{1}{2}\left(p - \frac{p_{xx} + p_{yy}}{2}\right)L_z \tag{5.4}$$

$$p_{xx} = \langle p_{xx}(z)\rangle = \frac{1}{L_z}\int_{-L_z/2}^{L_z/2} p_{xx}(z)dz \tag{5.5}$$

$$4,900 \text{ atm Å} \approx \frac{1}{2}(1 \text{ atm } - \frac{p_{xx} + p_{yy}}{2}) \cdot 100\text{Å} \tag{5.6}$$

$$p_{xx} + p_{yy} \approx -194 \tag{5.7}$$

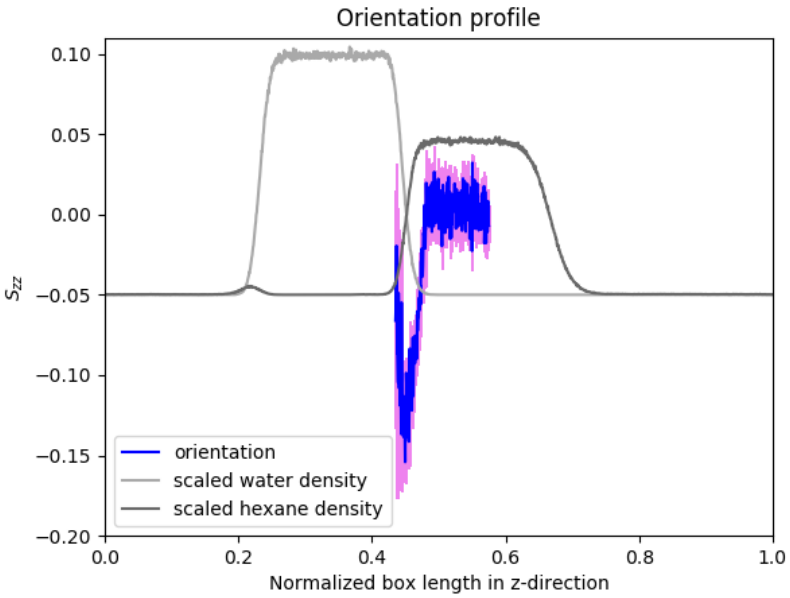
$$P_{\text{tot, MD sim.}} = \frac{p_{xx} + p_{yy} + p_{zz}}{3} \approx -65 \text{ atm} \tag{5.8}$$

$$\lim_{L_z\to\infty}\frac{\gamma}{L_z} \to = 0 \tag{5.9}$$

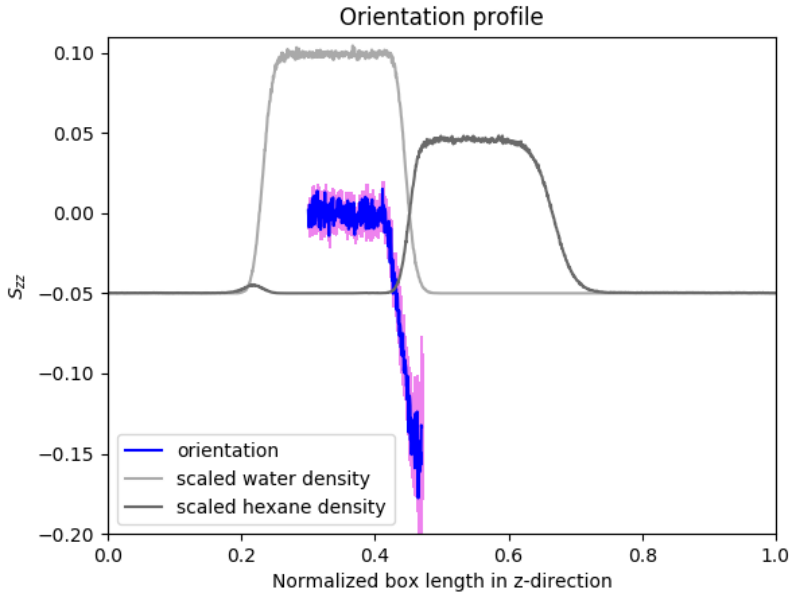
$$\frac{p_{xx} + p_{yy}}{2} \rightarrow p \tag{5.10}$$

6. 8/12/2018

(1) Working on slides explaining capillary theory contributions for candidacy ppt.
(2) We download orientations_hexane_full-298-32bead-hexane-orientations-corrected.txt and analyzed:



(3) Download orientations_water_32bead-water-unwrap.txt. Had issues with nan in orientation_profile_analysis_UPDATE for some strange reason, nans in the SE measurements, so used nan_policy='omit' for stats.sem. Finally, it works: used zlo 600 zhi 940:



7. 8/13/2018

(1) PIMC barostat acceptance criterion:

$$A(V'/V) = \min[1, e^{-\beta P(V'-V)} e^{-\beta[(U(r')-U(r))/nBeads]} e^{N\ln(V'/V)}] \tag{7.1}$$

(2) Developing ppt slide on density fluctuations:

$$\begin{aligned} \rho_W(z) &= \frac{1}{2}\rho_W + \frac{1}{2}\rho_W \text{erf}\left(\frac{z-\langle h_W\rangle}{\sqrt{2}\sigma_c}\right) \\ \rho_H(z) &= \frac{1}{2}\rho_H - \frac{1}{2}\rho_H \text{erf}\left(\frac{z-\langle h_H\rangle}{\sqrt{2}\sigma_c}\right) \end{aligned} \tag{7.2}$$

$$\begin{aligned} \sigma_0 &= \langle h_H\rangle - \langle h_W\rangle \\ \sigma_c^2 &= \frac{k_B T}{2\pi\gamma}\ln\left(\frac{L_{||}}{l_b}\right) \end{aligned} \tag{7.3}$$

8. 8/14/2018

(1)

$$\leq \tag{8.1}$$

9. 8/23/2018

(1) Test 1: P_16 on midway-l13b-21, printing to test.xyz but not output right away. About 175,000 printings to .xyz (about 10,000 steps) but none to output. Started printing to output after about 4:00 minutes. Stopped printing and said "Turn 3271000 15.86 per" without finishing line. At about 8 minutes it printed again to output file, catching up...and immediately got stuck again.
(2) Test2: P_16 on midway2-gpu06, printing to output right away, no problem.
(3) Test 3: P_16 on midway-l13b-21, printing to test.xyz but not output right away.

10. 9/26/2018

(1) According to Professor de Pablo, the priority item for the path integral research project is to fix the path integral monte carlo barostat in DASH. So, we will start by investigating the code that Mike has already written for this barostat on the DASH github.
(2) First, let's compile the most recent version of DASH on Midway. Doing "git clone https://github.com/dreid1991/md_engine.git". Saving this compiled version on Midway 1 in DASH-9-26-2018.
(3) Information on BoostPython: https://wiki.python.org/moin/boost.python/GettingStarted.
    The BoostPython library binds C++ and Python in a mostly-seamless fashion. It is just one member of the boost C++ library collection. Use the Boost-Python library to quickly and easily export C++ to python such that the python interface is very similar to the C++ interface. Boost.Python bindings are written in pure C++, using no tools other than your editor and your C++ compiler.

11. 10/1/2018

(1) Created tip4pF_9-26-2018.py to run q-TIP4P/F water model in DASH 9/26/2018 version.
(2) Continuing to work on developing this basic test code, checking all possible options to make sure python script is correct.

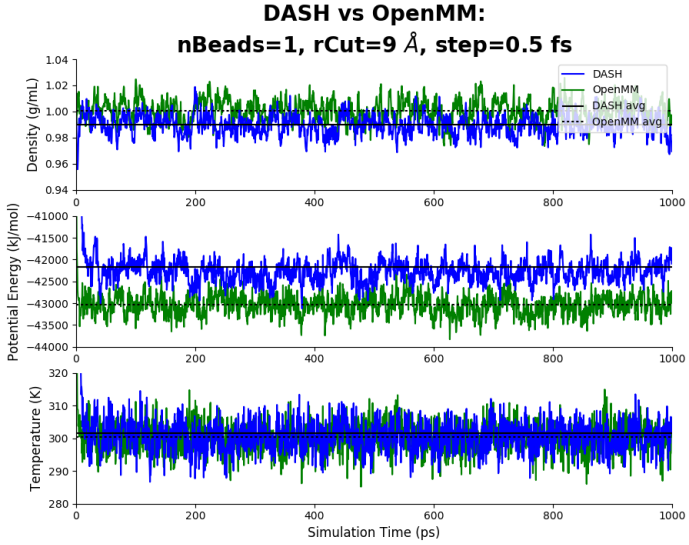
12. 10/3/2018

(1) We will begin by running a series of tests on q-TIP4P/F water. We will use the following system: 1000 water molecules, 1 ns, initial density of 0.997, temperature of 298 K, pressure 1 atm, printing data every 1000 steps. We are using run_9-26-2018.sh and tip4pF_9-26-2018.py to run DASH version from 9/26/2018.
(2) On /home/swansonk1, Midway 1, go into openmm folder. On depablo-gpu, module load Anaconda3, source activate openmm-env. Downloaded pimd.py from Markland, called it pimd_original.py, and created new file pimd_modified.py. In the modified file, we change the nonbondedCutoff to 0.9*nanometers, comment out the platform and properties assignments (so that we use OpenCL by default), and change the Simulation() to get rid of platform and properties arguments.
(3) Searching for OpenMM code here: http://docs.openmm.org/development/api-python/
(4) Found RPMDMonteCarloBarostat. "This class is very similar to MonteCarloBarostat, but it is specifically designed for use with RPMDIntegrator. For each trial move, it scales all copies of the system by the same amount, then accepts or rejects the move based on the change to the total energy of the ring polymer."
(5) Having trouble understanding the kinetic energy units in DASH...not sure what is going on here. I just want to compute the total kinetic energy. For now we will leave it out then.
    Tests using the Isotropic Monte Carlo Barostat:
    Time Step: 0.5 fs, rCut: 9 A, nBeads: 1, jobname MC-1/B-1/MM-1

Time Step: 0.5, rCut: 9, nBeads: 16, jobname MC-2/B-2/MM-2
Time Step: 0.5, rCut: 9, nBeads: 32, jobname MC-3/B-3/MM-3
Time Step: 0.5, rCut: 12, nBeads: 1, jobname MC-4/B-4/MM-4
Time Step: 0.5, rCut: 12, nBeads: 16, jobname MC-5/B-5/MM-5
Time Step: 0.5, rCut: 12, nBeads: 32, jobname MC-6/B-6/MM-6
Time Step: 0.25, rCut: 9, nBeads: 1, jobname MC-7/B-7/MM-7
Time Step: 0.25, rCut: 9, nBeads: 16, jobname MC-8/B-8/MM-8
Time Step: 0.25, rCut: 9, nBeads: 32, jobname MC-9/B-9/MM-9
Time Step: 1.0, rCut: 9, nBeads: 1, jobname MC-10/B-10/MM-10
Time Step: 1.0, rCut: 9, nBeads: 16, jobname MC-11/B-11/MM-11
Time Step: 1.0, rCut: 9, nBeads: 32, jobname MC-12/B-12/MM-12
Time Step: 0.5 fs, rCut: 9A, nBeads: 1, NO BAROSTAT, jobname dash-noP/open-noP (compare energies at same density)
Time Step: 0.5 fs, rCut: 9A, nBeads: 64, jobname MC-13/B-13/MM-13
Time Step: 0.5 fs, rCut: 9A, nBeads: 128, jobname MC-13/B-13/MM-13

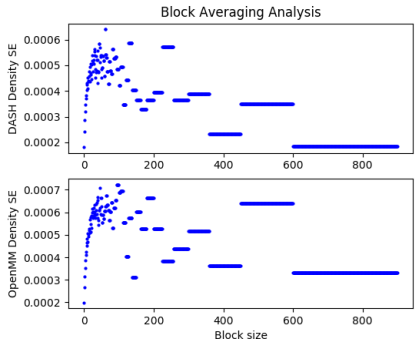
13. 10/4/2018

(1) MC-1 thru 3 and MM-1 thru 3 finished running. Let's now analyze the results.
(2) Wrote data_analysis.py in Path Integrals/hexane-water/dash_work/water folder for analysis of this data.
(3) Here is an initial result for Time Step: 0.5 fs, rCut: 9 A, nBeads: 1:

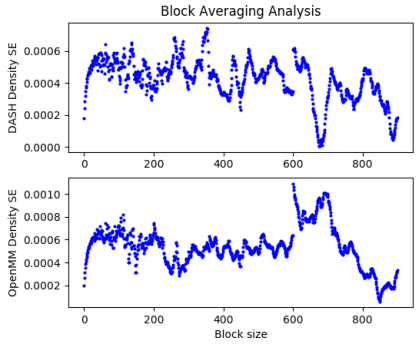


(4) Realized that we should have an equilibration period before comparing averages, since the beginning of the simulation seems to screw things up. We should also report average values with SE values.
(5) Using simulations MC-1 and MM-1, we can look at the standard error of the density for various block sizes to choose an appropriate number of blocks for

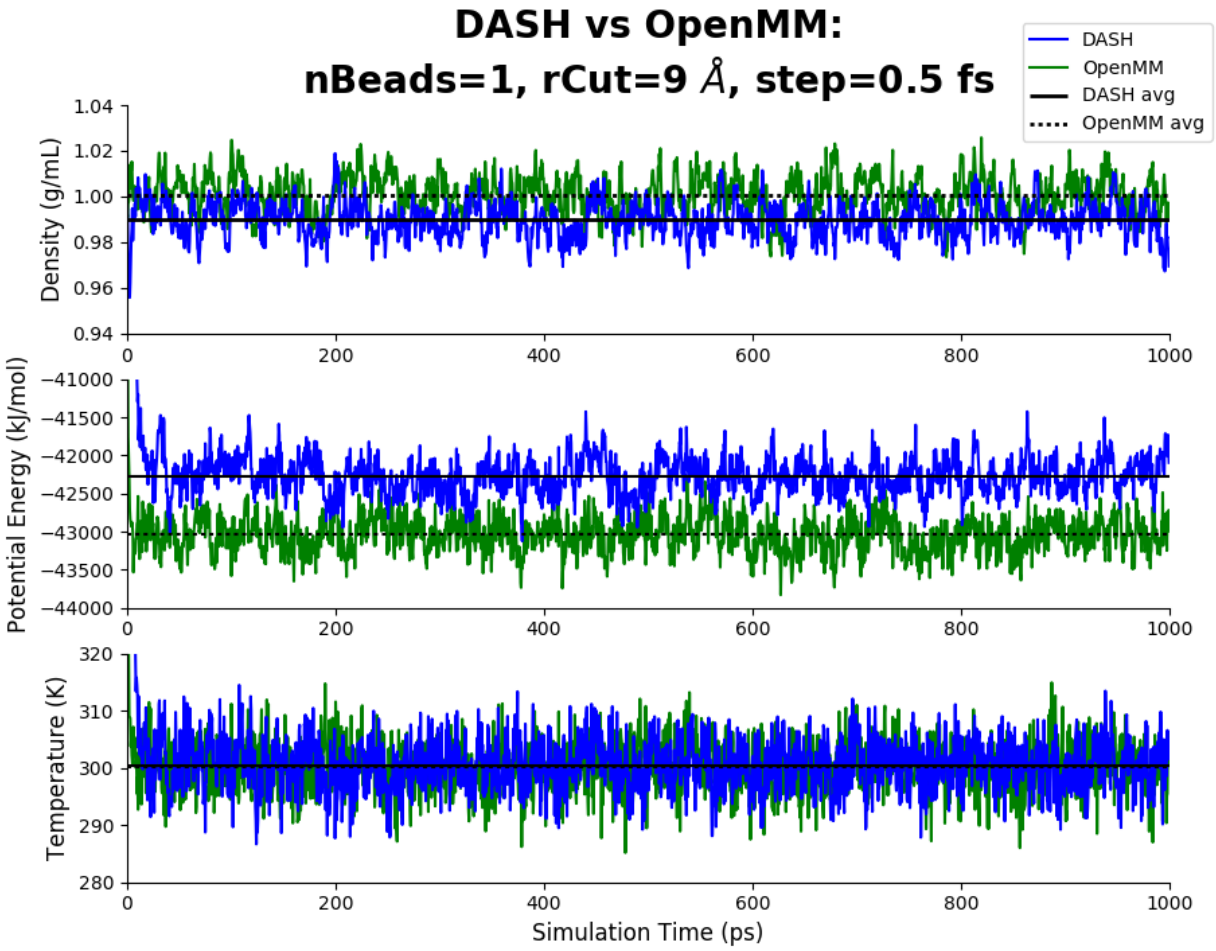
block averaging. Wrote the file SE_analysis.py in the same folder as above. We get the following:



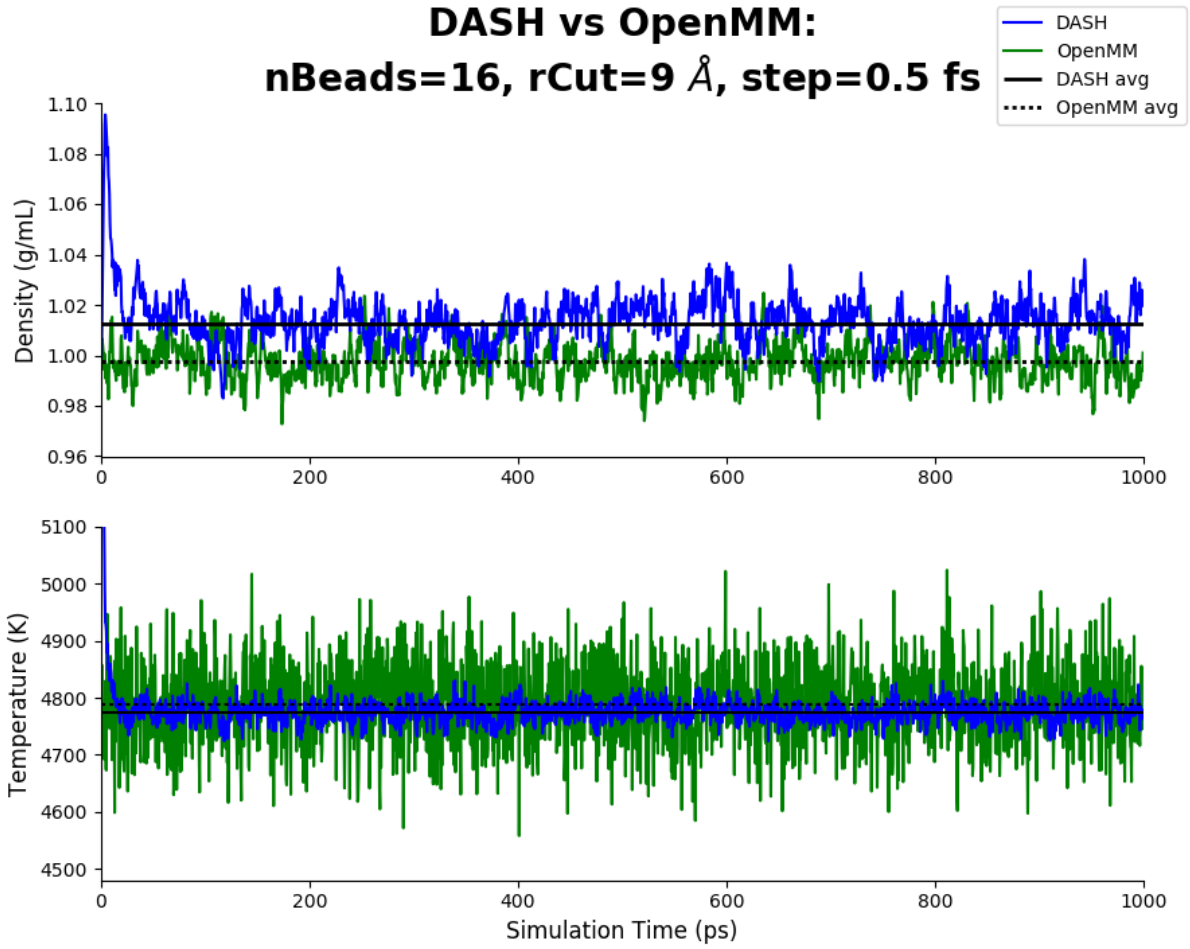
(6) Based on this figure, we will choose 100 ps for an appropriate block size. Since we are reserving the first 100 ps for equilibration, this gives us 9 blocks of size 100 ps.
(7) We also tried a different format allowing for variable block sizes:



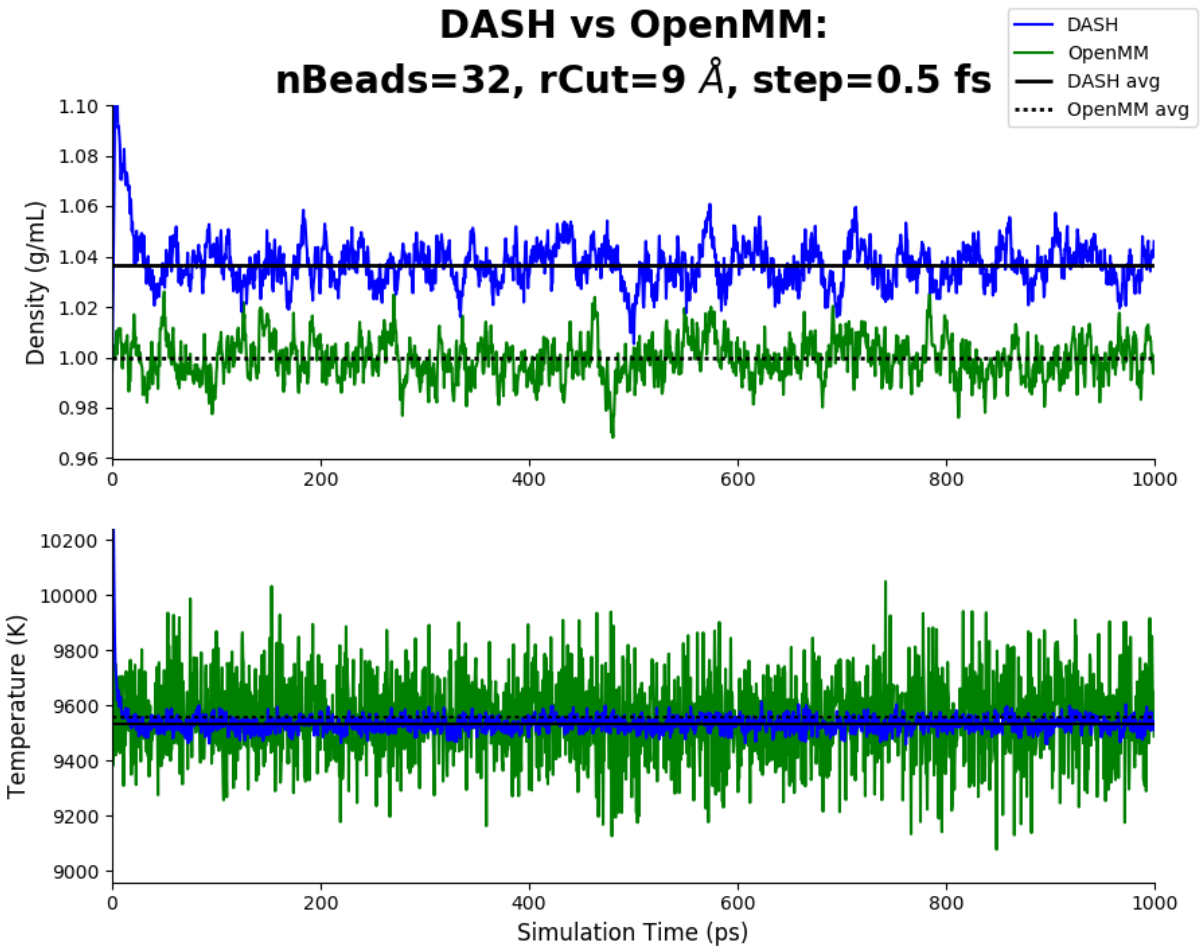
(8) The If the number of blocks is 2.5, then it leaves out the last 0.5 of data. There could be other ways of doing this...but for now, this is fine. Result is still the same: 100 ps is a good block size.
(9) Using 9 blocks of size 100 ps, we then have (USING +/- 2*SE):
    DASH Density: 0.98974 $\pm$ 0.00072 g/mL
    OpenMM Density: 1.00073 $\pm$ 0.00132 g/mL
    DASH Potential Energy: -42276.85961 $\pm$ 35.38552 kJ/mol
    OpenMM Potential Energy: -43041.14116 $\pm$ 33.45556 kJ/mol
    DASH Temperature: 300.30567 $\pm$ 0.27942 K
    OpenMM Temperature: 300.09525 $\pm$ 0.386 K



(10) For Time Step: 0.5 fs, rCut: 9 A, nBeads 16:
(11) Using 9 blocks of size 100 ps, we then have:
DASH Density: 1.01268 ± 0.00176 g/mL
OpenMM Density: 0.99751 ± 0.00104 g/mL
DASH Temperature: 4774.04631 ± 1.91432 K
OpenMM Temperature: 4788.12303 ± 2.5978 K



(12) For Time Step: 0.5 fs, rCut: 9 A, nBeads 32:
(13) Using 9 blocks of size 100 ps, we then have:
DASH Density: 1.03649 $\pm$ 0.00082 g/mL
OpenMM Density: 0.9995 $\pm$ 0.00176 g/mL
DASH Temperature: 9532.01452 $\pm$ 2.57316 K
OpenMM Temperature: 9555.22163 $\pm$ 6.96436 K



(14) It would be interesting to see what happens for 64 beads. Let's run both of these under the same above conditions. These will be simulations no. 23 and 24.
(15) We also want to run the Berendsen barostat in DASH for 0.5 fs, rCut 9, and nBeads 1, 16, and 32. These will be simulations 25, 26, 27.
(16) It would also be interesting to run with no PI beads for 0.5 fs, rCut 9, which will be simulation 28 and 29 for Berendsen and MonteCarlo, respectively.
(17) We also want to run the simulations that compare different rCut, so we will run DASH MC barostat using 0.5 fs, rCut 12 A, and nBeads 1, 16, 32, which will be simulations 30, 31, 32 and job names MC-4, MC-5, MC-6. We will also do this in OpenMM, jobs MM-4, MM-5, MM-6 and simulations 33, 34, 35.

E-mail address: swansonk1@uchicago.edu

INSTITUTE FOR MOLECULAR ENGINEERING, UNIVERSITY OF CHICAGO, 5640 S ELLIS AVE, CHICAGO, IL 60637