Molecular Dynamics Simulation

Elijah Flenner

2016
What is Classical Molecular Dynamics Simulation?

Numerical method based on Newton’s equations of motion to calculate thermodynamic quantities.

- Specific Heat.
- Compressibility.
- Free energy differences.
- Diffusion coefficients.
- Used to test theories.

Provides a means to connect microscopic behavior of atoms with experimental results.
Molecular Dynamics Basics

- Potentials (force fields) used to determine accelerations.
  - \( \mathbf{F}_n = -\nabla_n U(r^N) \)
  - When \( U(r^N) \) is a minimum, \( \mathbf{F}_n \) is zero.

- Use Newton’s law, \( \mathbf{F} = ma \) to determine accelerations.
- Use the accelerations to update the velocities.
- Use the velocities to update the positions.
- Calculate new forces at the new positions, and continue.
- Calculate quantities of interest from the particle positions and velocities.
- Can be thought of as a numerical experiment.
- Use of conservative forces means that Newton’s equations of motion conserve total Energy.
  - Kinetic energy \( K = \frac{1}{2} \sum_n m_n v_n^2 \).
  - Potential energy \( U(r^N) \).
  - Total energy \( E = K + U \).
Setting Up a Simulation

- Start with initial conditions.
  - Initial positions.
    - Periodic structure.
    - Complicated systems requires care.
  - Initial velocities $v$.
    - Usually chosen from a Boltzmann distribution.

$$f(v) = \left( \frac{m}{2\pi k_B T} \right)^{(1/2)} e^{-mv^2/(2k_B T)}$$

![Probability distribution of velocities](image.png)
Setting Up a Simulation

- May need to minimize the energy to remove overlaps.
  - Find energy minimum, thus $F \approx 0$.
  - Initial displacement is only due to velocities.
- Equilibrate system to desired temperature and pressure.
  - Temperature equilibration with a thermostat.
  - Pressure equilibration with a barostat.
- Run production runs and calculate averages.
  - Production runs designed to answer a specific question.
- Use periodic boundary conditions to simulate large system.
- We will use benzene as an example.
Open VMD.

Load structure.
1. File -> New Molecule
2. Browse for file benz.prmtop.
3. Make sure file type is correct (AMBER7 Parm) and load.
4. Browse for file benz.mdcrd and load as AMBER coordinates with periodic boundary conditions.

Examine structure.
1. Graphics -> Representations
2. Change Coloring Method and Drawing Method (play around a little).

We will be using the Amber simulation package.
Energy Minimization

If particles are too close and the forces too large, the system may explode. Find a local potential energy minimum so initial forces are almost zero.

Parameter and Topology file, benz.prmtop.
- Contains information about the force field and identifies the atoms in coordinate file.

First molecular dynamics script with the following format:

1. Comment line.
2. &cntrl
3. Lines describing the simulation.
   - Lines are "keyword=number,"
   - Number may be an integer flag or floating point floating point number.
   - Don’t forget the comma.
   - Many options. See manual for details.
4. Finishing "/".
Benzene Minimization
&cntrl
imin=1,
ntx=1,
irest=0,
maxcyc=12000,
ncyc=6000,
ntpr=100,
ntwx=1000,
cut=12.0,
ioutfm=0,
/

- **imin** = Flag to perform energy minimization: 1=perform minimization.
- **ntx** = Flag to read in coordinates and velocities: 1=coordinates, no velocities.
- **irest** = Flag to run as new simulation: 0=new simulation.
- **maxcyc** = Number of iterations for steepest descent minimization.
- **ncyc** = Flag for energy minimization procedure: 1=Steepest descent and conjugate gradient.
- **ntpr** = Every number of steps to print output to mdout and mdinfo.
- **ntwx** = Every number of steps to save coordinates.
- **cut** = length of non-bonded cutoff.
- **ioutfm** = Flag for the format of the output file. 0=ASCII human readable.
Website for submitting a gpu job.

http://wp.natsci.colostate.edu/fixman/class/running-amber-cluster/
Run the Minimization

pmemd -O -i benz.min.in -o benz.min.log -p benz.prmtop -c benz.inpcrd -r benz.min.rst -x benz.min.mdcrd -inf benz.min.inf
pmemd.gpu -O -i benz.min.in -o benz.min.log -p benz.prmtop -c benz.inpcrd -r benz.min.rst -x benz.min.mdcrd -inf benz.min.inf

- **-O** = Overwrite files.
- **-i benz.min.in** = input script.
- **-o benz.min.log** = output log file. Printed to every ntpcr steps.
- **-p benz.prmtop** = parameter and topology file.
- **-c benz.inpcrd** = input coordinate file.
- **-r benz.min.rst** = final restart file.
- **-x benz.min.mdcrd** = output coordinate file.
- **-inf benz.min.inf** = information file written every ntpcr steps.

Load the trajectory, benz.min.mdcrd, and visualize the results. Do no load with periodic boundary conditions.
Heating the Liquid: benz.heat.in

Heat from T=0K to T=298K
&cntrl imin=0, irest=0, ntx=1, ntb=1, cut=12.0, ntc=2, ntf=2, tempi=0.0, temp0=298.0, ntt=3, gamma_In=2, nstlim=25000, dt=0.002, ntpri=100, ntwx=1000, ntwr=1000, ig=-1, ioutfm=0, /

- No minimization (imin=0).
- Run as new simulation (irest=0).
- Read coordinates and not velocities (ntx=1).
- ntb = Flag for periodic boundary conditions. 1=periodic and fixed.
- ntc = Flag for constrained bonds. 2=Constrain hydrogen bonds.
- ntf = Flag for bond force calculations. 2=omit hydrogen bonds.
- tempi = initial temperature: temp0=final temperature.
- ntt = Flag for temperature coupling. 3=Langevin dynamics.
- gamma_In = Parameter for temperature coupling.
Heating the Liquid: benz.heat.in

Heat from T=0K to T=298K
&cntrl imin=0, irest=0, ntx=1, ntb=1, cut=12.0, ntc=2, ntf=2, tempi=0.0, temp0=298.0, ntt=3, gamma_ln=2, nstlim=25000, dt=0.002, ntpr=100, ntwx=1000, ntwr=1000, ig=-1, ioutfm=0, /

- nstlim = Number of simulation steps.
- dt = Timestep.
- Printing to mdout and mdinfo every 100 steps and saving coordinates every 1000 steps.
- ntwr = Number of steps between each save a restart file.
- ig = Random number seed. -1 means use date/time.
- Output is ASCII
Run Heating Simulation


Starting from the final condition of the minimization (-c benz.min.rst).

Load results, benz.heat.mdcrd, in VMD and visualize the trajectory.

Examine Graphics->Representations and looks at periodic images in x and y direction.
Equilibrate to Ambient Conditions (298K and 1ATM)

&cntrl
imin=0, irest=1, ntx=5,
ntpr=100, ntwr=100, ntwx=100,
nstlim=5000,
dt=0.002,
ntt=3, temp0=298.0, gamma_ln=2,
ig=-1,
ntp=1,
barostat=2,
pres0=1.0,
ntc=2, ntf=2, ntb=2,
cut=12.0,
ioutfm=0,
iwrap=0,
/

- No minimization (imin=0).
- Run as a continuation (irest=1).
- Read coordinates and velocities (ntx=5).
- Langevin temperature coupling, T=298K.
- ntp = Flag for constant pressure simulation.
  1=Isotropic pressure scaling.
- barostat = Type of constant pressure simulation. 2=Monte Carlo.
- pres0 = Pressure, P=1.0 atm.
- SHAKE algorithm with fixed hydrogen bonds.
- iwrap = Flag for putting particles back in box. 0 = Do not translate particles into box.
- Output is ASCII
Equilibration

```
pmemd -O -i benz.equil1.in -o benz.equil1.log -p benz.prmtop -c benz.heat.rst -r benz.equil1.rst -x benz.equil1.mdcrd -inf benz.equil1.inf

pmemd.gpu -O -i benz.equil1.in -o benz.equil1.log -p benz.prmtop -c benz.heat.rst -r benz.equil1.rst -x benz.equil1.mdcrd -inf benz.equil1.inf
```

1. Examine benz.equil.log and benz.equil.inf.
2. Look at three files from running "process_mdout.perl benz.equil.log".
3. Plot summary.EKTOT and summary.TEMP with the the second column multiplied by 5.9616.
   - Temperature is determined from the Virial Theorem and the Kinetic Energy.
   - \( \langle \sum_n m_n v_n^2 \rangle = (3/2)Nk_BT \).  
   - \( \langle \cdot \rangle \) indicates an average.
   - Not the only method to determine temperature.
   - Most used method (by far) in molecular dynamics simulations.
4. Plot summary.ETOT
5. Plot summary.TEMP
6. Plot summary.VOLUME
Want potential energy and volume to fluctuate around an average value.
This is the first condition for equilibrium.

We need more equilibration!

Create input file to save trajectory and output information every 1000 time steps.
Run simulation for 1.5 million time steps.
What are we doing?

Langevin equation of motion.

\[ \ddot{r}(t) = -\gamma \dot{r} + F + R(t) \]

- \( R(t) \) simulates collisions with a heat bath.
- \( \gamma \) makes sure system does not heat.
- \( R(t) \) depends on \( \gamma \) and \( T \).
- \( \gamma \) given in input script.
- Poor choices of \( \gamma \) will equilibrare the system slowly.
What are we doing? Monte Carlo Barostat

- Tries to change the volume by some amount.
- Keep and reject volume changes according to proper thermodynamic ensemble.
After equilibration runs, then production runs are performed. Production runs are designed to produce configurations according to desired calculations. We will run constant energy as an example.

- This is just numerically solving Newton’s equations of motion.
- Energy slowly drifts due to numerical issues.
- Carefully chosen constant temperature can stop energy drift.
- Need to understand what the constant temperature simulation is doing.
Analysis

Use positions and velocities to calculate averages of interest.
We can calculate average values.

\[
\langle A(r, p) \rangle = \frac{1}{N_{\text{conf}}} \sum_{n=1}^{N_{\text{conf}}} A(r, p)
\]

\[
= \lim_{\tau \to \infty} \int_{0}^{\tau} A[r(t'), q(t')] dt'
\]

Static correlation functions.

\[
\langle A(r, p)B(r, p) \rangle = \lim_{\tau \to \infty} \int_{0}^{\tau} A[r(t'), p(t')] B[r(t'), p(t')] dt'
\]

\[
\langle A(t)B(t) \rangle = \lim_{\tau \to \infty} \int_{0}^{\tau} A(t')B(t') dt'
\]

Time correlation functions.

\[
\langle A(t_0)B(t_0 + t) \rangle = \lim_{\tau \to \infty} \int_{0}^{\tau} A(t')B(t' + t) dt'
\]
Example is the pair correlation function.

\[
\rho g(r) = \frac{1}{N} \left\langle \sum_n \sum_{m \neq n} \delta[\mathbf{r} - (\mathbf{r}_m - \mathbf{r}_n)] \right\rangle
\]

\[
g(r) = \frac{1}{N \rho 4\pi r^2} \left\langle \sum_n \sum_{m \neq n} \delta(r - |\mathbf{r}_m - \mathbf{r}_n|) \right\rangle
\]

- \(\rho\) is the density.
- \(\delta(x)\) is the Dirac delta function (distribution).
- Used to characterize the structure.
Calculating $g(r)$

1. Divide radial intervals from $r_{min}$ to $r_{max}$ in steps of $\delta r$.
2. Find the distance $r = |r_n - r_m|$ between each point $n$ and $m$.
   - Points can be atoms, center of mass, etc.
3. Create a histogram $h(i)$ of the points where $i = \text{integer}(r/\delta r)$.
4. Do this for $N_c$ configurations.

$$g(r_i) = \frac{h(i)}{4\pi \rho r^2 \delta r N_c}$$

1. $g(r) = 1$ for large $r$.
2. $4\pi \rho \int_0^r g(r')dr'$ is the number of points (atoms, molecules, etc.) found around the origin of $g(r)$. 
What about periodic boundary conditions?

The saved position of a particle could be in the main simulation box, or any of the periodic images. Furthermore, the particles could be close to the edge of a box. How do we figure out the distance between particles?

1. Find the distances $\delta x_{ij} = x_i - x_j$, $\delta y_{ij} = y_i - y_j$, and $\delta z_{ij} = z_i - z_j$ for particles in the main simulation box or any of the periodic images.

2. The $x$ separation between particles is given by $\delta x_{ij} = L_x \text{nint}(\delta x_{ij}/L_x)$, where $L_x$ is the length of the simulation box in the $x$-direction.

3. Similar for the $y$ and $z$ directions.

4. $\text{nint}()$ is the nearest integer function.

5. In Python $\text{nint}(.) = \text{int}(\text{round}(.))$.
Velocity autocorrelation function.

\[ Z(t) = \frac{1}{N} \left< \sum_n v_n(0) \cdot v_n(t) \right> \]

Mean square displacement.

\[ \left< \delta r^2(t) \right> = \frac{1}{N} \left< \sum_n |r_n(t) - r_n(0)|^2 \right> \]

Energy correlation function.

\[ \left< (E(0) - \langle E \rangle)(E(t) - \langle E \rangle) \right> = \langle E(0)E(t) \rangle - \langle E \rangle^2 \]
Calculating $\langle \delta r^2(t) \rangle$

1. Sum through time origins $t_0$ and obtain $r(t_0)$.
2. Sum through time displacements $t$ and obtain $r(t_0 + t)$.
3. Accumulate averages of $|r(t_0) - r(t_0 + t)|$.
4. Normalize and print output.

$$
\langle \delta r^2(t) \rangle = \frac{1}{N_{\text{part}}} \frac{1}{N_t} \sum_{t_0}^{N_t} \sum_{n}^{N_{\text{part}}} |r_n(t_0) - r_n(t_0 + t)|^2 = \frac{1}{N_{\text{part}}} \frac{1}{N_t} \sum_{t_0}^{N_t} \sum_{n}^{N_{\text{part}}} [x_n(t_0) - x_n(t_0 + t)]^2
$$

$$
+ \frac{1}{N_{\text{part}}} \frac{1}{N_t} \sum_{t_0}^{N_t} \sum_{n}^{N_{\text{part}}} [y_n(t_0) - y_n(t_0 + t)]^2
$$

$$
+ \frac{1}{N_{\text{part}}} \frac{1}{N_t} \sum_{t_0}^{N_t} \sum_{n}^{N_{\text{part}}} [z_n(t_0) - z_n(t_0 + t)]^2
$$
Long Time Limit

Diffusion Coefficient $D$

$$D = \lim_{t \to \infty} \frac{\langle \delta r^2(t) \rangle}{6t}$$

(3)
1. Have a trajectory.
2. Know what we want to calculate.
3. How do we do the calculation?

We will use MDAnalysis toolkit. Python tool to read, write, and analyze Molecular Dynamics trajectories.
MDAnalysis basics

What to import MDAnalysis and numpy.
Other useful python additions are scipy and math.

```python
import numpy as np
import MDAnalysis
import math
import scipy

Create a universe.

universe = MDAnalysis.Universe("topology_file","trajectory_file")

Make a selection of atoms.

selection = universe.select_atoms("atom selection")
```
Loop through trajectory.
Obtain coordinates and other needed information.

for ts in universe.trajectory:
    coord = ts.positions
    box = ts.dimensions
    coord[i][j] = \textit{j position (x=0,y=1,z=2) of particle i}
    box[j] = \textit{Length of simulation box in j-direction}

Perform calculation.
Print results.
Write coordinates of the geometric center of the first benzene.

```python
universe = MDAnalysis("benz200.prmtop","benz200.run2.mdcrd")
selection = universe.select_atoms("all")
print "Number of atoms in selection:", selection.n_atoms
print "Length of trajectory:", universe.trajectory.n_frames

for ts in universe.trajectory:
    for i in range(12):
        center_x = center_x + coord[i][0]
        center_y = center_y + coord[i][1]
        center_z = center_z + coord[i][2]
    center_x = center_x/12.0
    center_y = center_y/12.0
    center_z = center_z/12.0
    test_file.write("%f %f %f \n" %(center_x,center_y,center_z))```
Minimize and equilibrate the benzene simulation.
Run a constant energy production run of 300,000 steps.
Compute the pair correlation functions for the geometric center of benzene molecules.
Compute the mean square displacement for the geometric center of benzene molecules.
Compute the diffusion coefficient of the benzene molecule.