In-Class Work: Molecular Dynamics Simulations

1. Numerical Integration

I will first give some short introduction.

1a. For $f(t, y) = Ay$ write a program which uses the Euler-step for integration (and therefore use the flowchart on the white board) to integrate

$$\frac{dy}{dt} = f(t, y) = Ay$$

Use $A = 0.3, t_0 = 0.0, y_0 = 0.6, \Delta t = 0.2, n_{\max} = 100$ (that means $t_{\max} = 20.0$). Print t, y_{new} for every time-step. Save the data in a file, e.g. `out1sim0.2.dat`.

1b. What do you expect for $y(t)$ (we can solve the DE analytically).

1c. Add to your program that the exact solution for $t, y(t)$ is printed for every time-step into another file, e.g. named `out1theory.dat` Look at the comparison of the numerical solution and the theoretical solution with `xmgrace out1theory.dat out1sim0.2.dat`.

1d. Now rerun the program for $\Delta t = 0.1$ and adjust $n_{\max}$ to get the same $t_{\max} = 20.0$ and print into another file, e.g. `out1sim0.1.dat`. Then rerun the program again this time for $\Delta t = 0.01$ and $n_{\max}$ again adjusted. Look at your data with `xmgrace out1theory.dat out1sim*.dat`. When you got this, please get me, I will show you a few tools with `xmgrace`.

2. Newton's Second Law

2a. Read of our textbook pages 12 & 13 (first two pages of Ch 2). Make sure to get me, before you continue with 2b. I will give another intro.

2b. Numerically integrate for $F_x^{\text{net}} = -mg$. Use $g = 9.8, \Delta t = 0.2, t_{\max} = 20.0, x_0 = 5.0, v_0 = 2.3$. Print into a file $t, x(t), v_x(t)$. As above, also determine the analytical solution and rerun the numberical solution also for $\Delta t = 0.1$ and $\Delta t = 0.01$. Look at your comparison as in 1d.

3. Harmonic Oscillator & Surprise

3b. Numerically integrate this time for the harmonic oscillator, so $F_x^{\text{net}} = -kx$. We can also analytically solve this equation. Let's choose $x_0 = 5.0, v_0 = 0.0$, then the theoretical solution is

$$x(t) = 5.0\cos(\omega_0 t) \qquad v_x(t) = -5.0\omega_0 \sin(\omega_0 t)$$

where $\omega_0 = \sqrt{(k/m)}$. So we know the period $T = 2\pi/\omega_0$. Let's choose $\Delta t = T/n_{\text{div}}$. Integrate $F_x^{\text{net}} = -kx$ for $k = m = 1$ and for $n_{\text{div}} = 100$ and $n_{\max} = 10n_{\text{div}}$. Print also the analytical solution and compare. Try also with $n_{\text{div}} = 1000$. What happens? Get me, when you have the results.

4. Euler-Cromer

Read from our textbook page 45 (Ch 3 first page). Change your program from 3b to use the Euler-Cromer step instead of the Euler step. Repeat the integration and compare again with the theoretical solution.

5. Integration Methods

If time is left, read from our textbook Appendix 3A (starts on page 74)

Upcoming Deadlines:

- March 18, W, 9am: First Version of Program (copy into `~/share.dir/` and give read-permission `chmod a+r ~/share.dir/*`
- March 24, T, 9:30am: Second Version of Program (copy into `~/share.dir/` and give readpermission `chmod a+r ~/share.dir/*`

In-Class Work: Molecular Dynamics Simulations

3. Harmonic Oscillator & Surprise

Note: For the following in-class work you may use the solution to last class

```
~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/md2.py
```

3a. Numerically integrate this time for the harmonic oscillator, so $F_x^{\rm net} = -kx$. We can also analytically solve this equation. Let's choose $x_0 = 5.0, v_0 = 0.0$, then the theoretical solution is

$$x(t) = 5.0\cos(\omega_0 t) \qquad\qquad v_x(t) = -5.0\omega_0 \sin(\omega_0 t)$$

where $\omega_0 = \sqrt{(k/m)}$. So we know the period $T = 2\pi/\omega_0$. Let's choose $\Delta t = T/n_{\rm div}$. Integrate $F_x^{\rm net} = -kx$ for $k = m = 1$ and for $n_{\rm div} = 100$ and $n_{\rm max} = 10 n_{\rm div}$. Print also the analytical solution and compare. Try also with $n_{\rm div} = 1000$. What happens?

3b To investigate more, what happens, determine for each time step the total energy

$$E(t) = \frac{1}{2}mv_x^2 + \frac{1}{2}kx^2$$

Determine the total energy for the simulation $E_{\rm sim}$ and also the theoretical prediction $E_{\rm theory}$. In case you printed four columns $t, x(t), v_x(t), E(t)$ into files `out3theory.dat` and `out3sim100.dat` and `out3sim1000.dat`, then you can compare the $E(t)$-data with

```
xmgrace -block out3theory.dat -bxy 1:4 -block out3sim100.dat -bxy 1:4 -block out3sim1000.dat -bxy 1:4
```

You can change the y-axis to be logarithmic, by double-clicking on the y-axis and changing the `scale` from `linear` to `logarithmic`. Get me, when you have the results.

4. Euler-Cromer

Read from our textbook page 45 (Ch 3 first page). Change your program from 3b to use the Euler-Cromer step instead of the Euler step. Repeat the integration and compare again $x(t), v_x(t)$, and $E(t)$ of the simulations for $n_{\rm div} = 100$ and $n_{\rm div} = 1000$ with the theoretical solution.

5. Integration Methods (if time permits)

Read from our textbook Appendix 3A (starts on page 74)

6. Driven Damped Pendulum Intro & Trajectory

6a. Please get me, when you get here. I will give an introduction to the system, i.e. the equation of motion.

6b. So we ended up with the essential equation for the simulation of the driven, damped pendulum to be

$$\frac{d^2\theta}{dt^2} = \tilde{A}\,\cos\left(\tilde{\omega}_{\rm D}\,t\right) - \sin(\theta) - \tilde{\gamma}\,\frac{d\theta}{dt} \tag{2}$$

where we replaced $\tilde{t}$ by t simply for the convenience of notation. In the computer simulation we solve this equation numerically, i.e. our goal is to determine $\theta(t)$ and $\dot{\theta}(t)$.

Using the Euler method as written on the white board, program this driven damped pendulum. Use

$$\theta_0 = 0.0 \qquad \omega_0 = 1.9 \qquad \tilde{A} = 0.9 \qquad \tilde{\omega}_{\rm D} = 2.0/3.0 \tilde{\gamma} = 0.5 \qquad \Delta t = 0.01 \qquad n_{\rm max} = 5000$$

Print only every 10th MD-step $t, \theta(t), \omega(t)$. Look at $\theta(t)$ and $\omega(t)$. You can do this with
`xmgrace -block out6.dat -bxy 1:2 -bxy 1:3`

Upcoming Deadlines:

- March 24, T, 9:30am: Second Version of Program (copy into `~/share.dir/` and give read permission `chmod a+r ~/share.dir/*`
- March 31, T, 9:30am: Results (Figures and Interpretation) as hardcopy

In-Class Work: Molecular Dynamics Simulations

Note: For the following in-class work you may use the solution to last class

`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/md4.py`

6. Driven Damped Pendulum Intro & Trajectory

6a. I will start class with a quick intro to the driven damped pendulum and how to rescale time to get the fewest number of independent parameters.
6b. So we ended up with the essential equation for the simulation of the driven, damped pendulum to be

$$\frac{d^2\theta}{dt^2} = \tilde{A}\cos\left(\tilde{\omega}_{\rm D}\, t\right) - \sin(\theta) - \tilde{\gamma}\,\frac{d\theta}{dt} \tag{3}$$

where we replaced $\tilde{t}$ by t simply for the convenience of notation. In the computer simulation we solve this equation numerically, i.e. our goal is to determine $\theta(t)$ and $\dot{\theta}(t)$.

Using the Euler method as written on the white board, program this driven damped pendulum. Use

$$\theta_0 = 0.0 \quad \omega_0 = 1.9 \quad \tilde{A} = 0.9 \quad \tilde{\omega}_{\rm D} = 2.0/3.0 \quad \tilde{\gamma} = 0.5 \quad \Delta t = 0.005 \quad n_{\rm max} = 5000$$

Print only every 10th MD-step $t, \theta(t), \omega(t)$. (In the following I will call this `nprint=10`.) Look at $\theta(t)$ and $\omega(t)$. If your data are in the file `out6.dat` you can do this with
`xmgrace -block out6.dat -bxy 1:2 -bxy 1:3`

6c Think about what the energy of the driven damped pendulum is, and print and look at $E(t)$.

7. Period Doubling

Use $n_{\rm print} = 10$ and do 100000 MD-steps. To ensure to not plot the transient plot only after 20000 MD-steps.

7a. Look at $\theta(t), \omega(t)$ and $E(t)$ for $\tilde{A} = 1.046$.
7b. Look at $\theta(t), \omega(t)$ and $E(t)$ for $\tilde{A} = 1.052$.
7c. Look at $\theta(t), \omega(t)$ and $E(t)$ for $\tilde{A} = 1.054$.
7d. Look at $\theta(t), \omega(t)$ and $E(t)$ for $\tilde{A} = 1.078$.
7e. Get me when you got all results for 7a-7d. (Get them all on the screen, so that your class members can see them also.)

8. Poincaré Plot

8a. Incorporate periodic boundary conditions for θ, i.e. ensure that `thetanew` satisfies

$$-\pi < \theta \leq \pi$$

8b. To get $\omega(\theta)$ measured in phase with T_{D} determine Δt as $\Delta t = (2\pi/\tilde{\omega}_{\mathrm{D}})/n_{\mathrm{print}}$. Use $n_{\mathrm{print}} = 200$ and do 100000 MD-steps. To ensure to not plot the transient plot only after 20000 MD-steps. Look at the Poincaré plot $\omega(\theta)$ for the $\tilde{A}$ values of the above 7a-7d. Get me, when you have the results.

9. Bifurcation Diagram

Now you are ready to make a pretty cool plot, the bifurcation diagram. Add to your program a loop over $\tilde{A}$. Use $0.9 \leq \tilde{A} \leq 1.08$ in steps of $\Delta\tilde{A} = 0.002$. Remember to initialize t_0, θ_0, and ω_0 within the $\tilde{A}$-loop. Print $\tilde{A}, \theta_{\mathrm{PC}}, \omega_{\mathrm{PC}}$, where $\theta_{\mathrm{PC}}, \omega_{\mathrm{PC}}$ are the values from the Poincaré-plot of step 8.

10. Feigenbaum Number

If there is time left, read about the Feigenbaum Number and try to determine it for our simulation.

In-Class Work: Molecular Dynamics Simulations

Note: Unless you had finished in last class step 8. (Poincaré Plot) please use for the following in-class work the solution to last class

```
~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/md8.py
```

$$\frac{d^2\theta}{dt^2} = \tilde{A}\,\cos\left(\tilde{\omega}_{\rm D}\,t\right) - \sin(\theta) - \tilde{\gamma}\,\frac{d\theta}{dt} \tag{4}$$

9. Bifurcation Diagram

Now you are ready to make a pretty cool plot, the bifurcation diagram. Add to your program a loop over $\tilde{A}$. Use $0.9 \leq \tilde{A} \leq 1.08$ in steps of $\Delta\tilde{A} = 0.002$. Remember to initialize t_0, θ_0, and ω_0 within the $\tilde{A}$-loop. Print $\tilde{A}, \theta_{\rm PC}, \omega_{\rm PC}$, where $\theta_{\rm PC}, \omega_{\rm PC}$ are the values from the Poincaré-plot of step 8. (We used $\theta(t=0) = 0.0$, $\omega(t=0) = 1.9$, $\tilde{\omega}_{\rm D} = 2.0/3.0$ and $\tilde{\gamma} = 0.5$.)

In-Class Work: Work on Your Main Project

Today you will use the remaining class-time to work and get help on your main project. Take 10 – 15 min to plan which quantities you would like to plot (e.g. Sarah might plot for the DLA model the fractal dimension as function of a sticking probability for a random walker being next to an already grown cluster site, or as function of the parameter which governs how much more likely it is to stick to a site with more cluster member neighbors. Joe might want to plot the bifurcation diagram for θ_1 and one for θ_2 for fixed damping but varied $\tilde{A}$ or for fixed $\tilde{A}$ and varied damping. Or Joe might draw the bifurcation diagram for fixed $\tilde{A}$ and fixed damping but for varied initial condition $\omega(t=0)$ or a phase-diagram which indicates which initial conditions give period-2, which period-4, chaos, etc. Or Pete might plot the time it takes an earth-quake triggered in the center of the lattice to travel to the border of the lattice as function of the trigger amplitude or other parameters which specify the material of the earth.

Be creative with your planning of plots which would allow you to answer interesting questions. Start with a (if you wish way too long) wish list. Then take 10 min to sketch with a flow chart how to implement your first wish of a plot. Get as far as possible with your results. Get me for any help on any step in this process.

Upcoming Deadlines:

- March 25 (yesterday): 2nd Version of Program (graded as 1st Version)
- March 31: Results (Figures and Interpretation) as hardcopy
- April 7: Final Program (graded)
- April 9, Results section of second paper
- April 9, Results (graded)

In-Class Work: Figures for Scientific Paper & Talk

1. Figure-Requirements: Practice Example

I will walk you through how to use xmgrace to fullfill the following list of requirements for figures for scientific papers and talks. To show you how to do each step in xmgrace we will all start with the same data, the output-files from `md8.py`. I will show you how to make figures as on our webpage (see Guidelines for Main Project).

Copy into your working directory

```
~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/out7_A*.dat
```

Get the first data-set into xmgrace with

```
xmgrace out7_A1.078.dat
```

I will show to you how to get in a second set of data $\omega(t)$ (and how you could also show error bars if available).

Requirements for Figures:

- no title
- axes:
  - label axes (large enough, neat font-tools via clicks and via commands, location of axis label)
  - axis width thick enough
  - number of tick marks large enough
  - tick label size large enough
  - tick marks width and size large enough
  - choose wanted x-range and y-range (main features visible)
- legend (or equivalent with labels) large enough and each set should be labeled (or clear trend of which parameter was varied and in which range) and should not cover data
- label for major parameter large enough (in talk in figure, in paper if not in figure then in figure caption)
- symbols large enough and distinguishable and lines thick enough and distiguishable (keep in mind potential color blind person in audience) and in case of error bars thick enough error bars (labeled: see legend)
- in paper figure caption for each figure

Further xmgrace-tools:

- extra cool fonts: italics, greek, boldface, superscript, shift
- position & size of figure (so that in paper no white frame)
- storing info in file.xmgr (highly recommended)
- printing eps-file (prepare and print; and epstopdf)
- pull in further data-set via block-data
- arrows and labels etc. (drawing objects)
- how to recycle figure via deleting data and replacing with new data or via change of xmgr-file
- symbols: filled and open symbols
- if time: insets
- (not xmgrace but useful: keep logfile for how you made data and where they are)

2. Your Figures

For the next 15 min work on the figures of your main project results. Please ask if you need any further tools, because I could show you several further xmgrace-tools.

In-Class Work: Molecular Dynamics Simulations

10. Forces

We are going to simulate a monatomic liquid. We use the Lennard Potential, that means that the interaction potential V_{ij} of particle i at position $\mathbf{r}_i$ and particle j at position $\mathbf{r}_j$ is

$$V_{ij} = 4\,\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right) \tag{5}$$

where $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$. In the simulation we will need forces $\mathbf{F}_i$ on each particle $i = 1, \ldots, N$. We use the relation

$$\mathbf{F}_i = -\nabla_i V = -\nabla_i \sum_{i}^{N-1} \sum_{\substack{j>i \\ j\neq i}}^{N} V_{ij}$$

As preparation for the program determine explicitly the x-component of the force on particle i

$$F_{ix} = \sum_{\substack{j=1 \\ j\neq i}}^{N} -\frac{d}{dx_i} V_{ij}(r_{ij})$$

Upcoming Deadlines:

- April 7: Final Program (graded) $\longrightarrow$ keep working on your project
- April 9, Results section of second paper
- April 9, Results (graded)

In-Class Work: Molecular Dynamics Simulations

10. Forces

We are going to simulate a monatomic liquid. We use the Lennard Potential, that means that the interaction potential V_{ij} of particle i at position $\mathbf{r}_i$ and particle j at position $\mathbf{r}_j$ is

$$V_{ij}(r_{ij} = 4\,\epsilon\,\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right) \tag{6}$$

where $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$. In the simulation we will need forces $\mathbf{F}_i$ on each particle $i = 1, \ldots, N$. We use the relation

$$\mathbf{F}_i = -\nabla_i V$$

where

$$V = \sum_{i}^{N-1} \sum_{\substack{j>i \\ j\neq i}}^{N} V_{ij}$$

10a. As preparation for the program determine explicitly the x-component of the force on particle i

$$F_{ix} = \sum_{\substack{j=1 \\ j\neq i}}^{N} -\frac{d}{dx_i} V_{ij}(r_{ij})$$

and similarly determine F_{iy} and F_{iz}.

10b. When you get here, let me know so that (i) we confirm your F_{ix} etc. and (ii) I will introduce the "minimum image convention". Make a flow chart for the determination of the acceleration of all particles $\{\mathbf{a}_i\}$. We will use as length unit σ, as energy unit ϵ and as mass unit m. You therefore may set $\sigma = \epsilon = m = 1$. Use also Newton's third law that $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$.

10c. Use the results for F_{ix}, F_{iy} and F_{iz} and use also the sample program
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/md10_sample.py`
and add the determination of $\{\mathbf{a}_i\}$ ($= \mathbf{F}_i$ in Lennard-Jones units) for all particles i in the function `acceleration`. Compare your result in the file `out10.dat` with the file
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/out10_comp.dat`
(The unix command `diff file1 file2` helps you to check if `file1` and `file2` are identical.)

11. Molecular Dynamics Step

You are almost ready to add the molecular dynamics (MD) step to your program. We will use the velocity Verlet algorithm

$$\begin{aligned} \mathbf{r}_i(t+\Delta t) &= \mathbf{x}_i(t) + \mathbf{v}_i(t)\,\Delta t + 0.5\,\mathbf{a}_i(t)\,(\Delta t)^2 \\ \mathbf{v}_i(t+\Delta t) &= \mathbf{v}_i(t) + 0.5\,[\mathbf{a}_i(t) + \mathbf{a}_i(t+\Delta t)]\,\Delta t \end{aligned} \tag{7}$$

11a. Draw a flow chart for the MD program using equations (10). Include in your flow chart the periodic boundary conditions

$$\begin{aligned} -L/2 &\leq x_i < L/2 \\ -L/2 &\leq y_i < L/2 \\ -L/2 &\leq z_i < L/2 \end{aligned} \tag{8}$$

11b. Using your flow chart, add to your program from step 10 the loop over MD steps and the MD step including periodic boundary conditions. Print every `nprint` steps $t, x_0(t), y_0(t), z_0(t), x_6(t), y_6(t), z_6(t)$. Use $\Delta t = 0.001$, `nprint=5`, and run your program for `maxmdsteps=500` MDsteps (as already in sample program for 10.). Compare your results with
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/out11traj.dat`
You also find the final configuration $(x_i, y_i, z_i, v_{x,i}, v_{y,i}, v_{z,i})$ in
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/out11finalconf.dat`

Upcoming Deadlines:

- April 7: Final Program (graded) $\longrightarrow$ keep working on your project
- April 9, Results section of second paper
- April 9, Results (graded)
- April 14, Mini-Project II in class (graded) and abstract

In-Class Work: Molecular Dynamics Simulations

11. Molecular Dynamics Step In last class you determined that the force in x-direction on particle i is

$$F_{ix} = 48\,\epsilon \sum_{\substack{j=1 \\ j\neq i}}^{N} \left(\frac{\sigma^{12}}{r_{ij}^{14}} - 0.5\frac{\sigma^{6}}{r_{ij}^{8}} \right) (x_i - x_j) \tag{9}$$

where r_{ij} is the distance between particles i and j $r_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$. Similarly you obtain F_{iy} and F_{iz}.
We change now this force slightly (for a faster program), by setting $\mathbf{F} = 0$ for $r_{ij} > 2.5\sigma$.
I will start class with explaining the program
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/md11_start.py`
which includes already the determination of $\{\mathbf{a}_i\}$ ($= \mathbf{F}_i$ in Lennard-Jones units) for all particles i in the function `acceleration`.

You are almost ready to add the molecular dynamics (MD) step to your program. We will use the velocity Verlet algorithm

$$\begin{aligned} \mathbf{r}_i(t+\Delta t) &= \mathbf{x}_i(t) + \mathbf{v}_i(t)\,\Delta t + 0.5\,\mathbf{a}_i(t)\,(\Delta t)^2 \\ \mathbf{v}_i(t+\Delta t) &= \mathbf{v}_i(t) + 0.5\,[\mathbf{a}_i(t) + \mathbf{a}_i(t+\Delta t)]\,\Delta t \end{aligned} \tag{10}$$

11a. Draw a flow chart for the MD program using equations (10). Include in your flow chart the periodic boundary conditions

$$\begin{aligned} -L/2 &\leq x_i < L/2 \\ -L/2 &\leq y_i < L/2 \\ -L/2 &\leq z_i < L/2 \end{aligned} \tag{11}$$

11b. Using your flow chart, add to `md11_start.py` the loop over MD steps and the MD step including periodic boundary conditions. Print every `nprint` steps $t, x_0(t), y_0(t), z_0(t), x_6(t), y_6(t), z_6(t)$. Use $\Delta t = 0.001$, `nprint=5`, and run your program for `maxmdsteps=500` MDsteps. Compare your results with
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/out11traj.dat`
You also find the final configuration $(x_i, y_i, z_i, v_{x,i}, v_{y,i}, v_{z,i})$ in
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/out11finalconf.dat`

11c. Use xmgrace (or other graphics) to look at your results. Interpret them.

12. Potential Energy

Add to the MD program a function which determines the potential energy per particle

$$V/N = \frac{1}{N}\sum_{i=1}^{N}\sum_{j>i} V_{\text{cutshift}} \tag{12}$$

where for $r_{ij} < 2.5$ $V_{\text{cutshift}} = V_{ij} - V_{ij}(2.5\sigma)$, $V_{ij} = 4\,\epsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right)$ and $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$. For $r_{ij} \geq 2.5$ $V_{\text{cutshift}} = 0$. Include the minimum image convention similar to how it is used for the determination of the acceleration. Run the simulation for 500 MD steps. Print also V/N every `nprint=5` steps. Let us together have a look at your result for $V(t)$.

13. Kinetic Energy and Temperature

13a. Add to your program from 12. a function which determines the kinetic energy $E_{\text{kin}} = 0.5\sum_{i=1}^{N}\mathbf{v}_i^2$. Print E_{kin} also every `nprint=5` MD steps. Look at your result.

13b. Temperature T is related to the kinetic Energy. How? Determine and look at $T(t)$.

Upcoming Deadlines:

- TODAY: April 7: Final Program (graded)
- April 9, Results section of second paper (Results will be graded)
- April 14, Mini-Project II in class (graded) and abstract

In-Class Work: Molecular Dynamics Simulations

12. Potential Energy

If you had finished the in-class work 11b, you may use your program, if not, please start with
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/md12_start.py`
Add to the MD program a function which determines the potential energy per particle

$$V/N = \frac{1}{N}\sum_{i=1}^{N}\sum_{j>i} V_{\text{cutshift}} \tag{13}$$

where for $r_{ij} < 2.5$ $V_{\text{cutshift}} = V_{ij} - V_{ij}(2.5\sigma)$, $V_{ij} = 4\,\epsilon\,\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right)$ and $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$. For $r_{ij} \geq 2.5$ $V_{\text{cutshift}} = 0$. Include the minimum image convention similar to how it is used for the determination of the acceleration. Run the simulation for 500 MD steps. Print also V/N every `nprint=5` steps. Let us together have a look at your result for $V(t)$.

13. Kinetic Energy and Temperature

13a. Add to your program from 12. a function which determines the kinetic energy $E_{\text{kin}} = 0.5\sum_{i=1}^{N}\mathbf{v}_i^2$. Print E_{kin} also every `nprint=5` MD steps. Look at your result.

13b. Temperature T is related to the kinetic Energy. How? Determine and look at $T(t)$. Get me, when you found the answer.

14. Maxwell-Boltzmann Distribution

Copy into your working directory
`~kvollmay/classes.dir/phys338.dir/phys338_s2015.dir/md.dir/maxboltest_start.py`

14a. Run the program and look at the resulting $P(v_x)$. Add to the program the determination of the expected Maxwell-Boltmann distribution

$$G(v_x) = \frac{1.0}{\sqrt{2\pi}\sigma}\exp\left(-v_x^2/(2\sigma^2)\right)$$

where

$$\sigma = \sqrt{\langle v_x^2\rangle} = \sqrt{\frac{kT}{m}}$$

and compare $P(v_x)$ and $G(v_x)$.

14b. Play with each of the constants, how they change $P(v_x)$ and the comparison with $G(v_x)$.

15. Temperature Bath

Add to your program from 13b a constant temperature bath by periodically redrawing all velocities from the Maxwell-Boltzmann distribution of the desired temperature. Look at your results and play with the bath temperature.

Upcoming Deadlines:

- April 10 (extended): Results section of second paper (Results will be graded)
- April 14: Mini-Project II (in class & graded)
  and title and abstract of main projectpaper
- April 21, 23, 28: 5, 5, 3 Talks
- April 21: First Version of Second Main Project Paper
- May 6 (our final): Final Version of Second Main Project Paper