AA 2021/2022 Computational Methods
Final projects

Mario Galante[*], Almaz Khabibrakhmanov[†], Alexandre Tkatchenko

Theoretical Chemical Physics group, FSTM,
Campus Limpertsberg, University of Luxembourg

[*] mario.galante@uni.lu

[†] almaz.khabibrakhmanov@uni.lu

Molecular dynamics simulations of hard spheres in a box

The hard spheres model is one of the first approaches that was used to perform molecular dynamics simulations of atomic gases. The main idea of the method is to evolve each particle in a linear uniform motion until a collision occurs. The velocity of the particles involved in the collision is then updated assuming purely elastic scattering. This operation is repeated until the desired end time of the simulation, as shown in Algorithm 1.

The key step of the algorithm is the determination of which is the first collision that happens in the system starting from a time t_0. Given a pair of particles, i, j, their collision time is given by

$$t_{ij} = \frac{-b_{ij} - \sqrt{b_{ij}^2 - v_{ij}^2(r_{ij}^2 - d^2)}}{v_{ij}^2}, \tag{1}$$

where $r_{ij} = |\boldsymbol{r}_i - \boldsymbol{r}_j|$, $v_{ij} = |\boldsymbol{v}_i - \boldsymbol{v}_j|$ and $b_{ij} = \boldsymbol{r}_{ij} \cdot \boldsymbol{v}_{ij}$. Assuming that the smallest (positive) collision time t_{ij} is found for particles i_0, j_0, the velocities of such particles are to be updated at $t = t_{i_0j_0}$ by imposing momentum conservation:

$$\boldsymbol{v}_{i_0} = \boldsymbol{v}_{i_0} + \Delta\boldsymbol{v}, \quad \boldsymbol{v}_{j_0} = \boldsymbol{v}_{j_0} - \Delta\boldsymbol{v} \quad \text{where,} \quad \Delta\boldsymbol{v} = \frac{\boldsymbol{r}_{i_0j_0}}{d^2} b_{i_0j_0}. \tag{2}$$

Algorithm 1 Hard spheres MD

```
Given $N$ particles of diameter $d$ and equal mass in a 2D box of length $L$
Set the total number of collisions, Nc, and the maximum simulation time
for c = 1, . . . ,Nc+1 do
    calculate collision times, $t_{ij}$, for all pairs
    $t_{i_0j_0} = min\{t_{ij}\}$
    $\boldsymbol{r} = \boldsymbol{r} + \boldsymbol{v}t_{i_0j_0}$                    ▷ Evolve all particles until $t_{i_0j_0}$
    update the velocities of $i_0$ and $j_0$ according to Eq. (2)
end for
```

1. Consider two hard spheres in 2D of diameter $d = 0.1$ with initial conditions $\boldsymbol{r}_1 = (0.25, 0.5)$, $\boldsymbol{r}_2 = (0.75, 0.5)$, $\boldsymbol{v}_1 = (0.5, 0.)$ and $\boldsymbol{v}_2 = (-0.1, 0.)$. Use Algorithm 1 to calculate their dynamics and plot the results (note that step 5 is here not necessary).

2. Now consider 4 hard spheres positioned at $(\pm0.25, \pm0.25)$. Select initial velocities with directions pointing towards the center of the square. Choose the same magnitude for particles on the same diagonal of the square but different for the different diagonals (so that you'll have one scattering at a time). Plot the resulting dynamics.

3. In practice, all gases are confined within a delimited space. To include the presence of the walls of a squared container of length L we must consider (elastic) particle-wall and particle-particle collisions simultaneously. Modify your code accordingly and use the same setup as point 1 to test your implementation. Stop the simulation after 10 collisions.

4. Run simulations for 4 spheres in a box of linear size $L = 1, 2, 5, 10$ for 300 collisions. Compare the average distance that a particle travels without collisions for the different box size.

Simple pendulum

The Newton equation of motion for a weight suspended from a pivot through a massless rod of fixed length, L, is one of the most studied in physics,

$$\frac{d^2\vartheta}{dt^2} = -\frac{g}{L}\sin(\vartheta), \quad \frac{d\vartheta}{dt}(0) = 0, \quad \vartheta(0) = \vartheta_0. \tag{3}$$

For small angles, the linear approximation can be taken, namely

$$\frac{d^2\vartheta}{dt^2} \approx -\frac{g}{L}\vartheta, \quad \frac{d\vartheta}{dt}(0) = 0, \quad \vartheta(0) = \vartheta_0. \tag{4}$$

1. Use the Midpoint method to integrate the two differential equations and determine numerically the critical value of ϑ_0 that results in the breakdown of the linear approximation. Set $L = 1$ m.

2. Implement the integration method displayed in Algorithm 1 and compare its error with the Midpoint method for both the differential equations.

3. In Eq. (4), the only driving is given by gravity. If an additional periodic force is added, one obtains the equation,

   $$\frac{d^2\vartheta}{dt^2} = -\frac{g}{L}\big(1 + h\cos(\omega t)\big)\vartheta. \tag{5}$$

   Compare the time evolution of Eqs. (4) and (5), calculated using Algorithm 1. Set $h = 0.5$ and compare the different results obtained for $\omega = 0.6, 1.6, 3, 6.3, 15.5$ Hz.

Algorithm 2

```
Given the equation $x'' = f(x, v, t)$
set the total number of steps, Nt, and the increment dt
for i = 1, ... ,Nt do
    $k_1 = dt \cdot v_i$
    $j_1 = dt \cdot f(x_i, v_i, t_i)$
    $k_2 = dt \cdot (v_i + j_1/2)$
    $j_2 = dt \cdot f(x_i + k_1/2, v_i + j_1/2, t_i + dt/2)$
    $k_3 = dt \cdot (v_i + j_2/2)$
    $j_3 = dt \cdot f(x_i + k_2/2, v_i + j_2/2, t_i + dt/2)$
    $k_4 = dt \cdot (v_i + j_3)$
    $j_4 = dt \cdot f(x_i + k_3, v_i + j_3, t_i + dt)$
    $x_{i+1} = x_i + (k_1 + 2k_2 + 2k_3 + k_4)/6$
    $v_{i+1} = v_i + (j_1 + 2j_2 + 2j_3 + j_4)/6$
end for
```

Growth model

Growth models are commonly employed to analyze the formation of clusters of bacteria and other active matter. The starting point is generally a single particle, called a *seed*, at the center of a grid. Then additional particles are progressively added in order to analyze the shape of the cluster depending on the chosen approach to simulate the growth.

Algorithm 3 Growth model

```
Given an $N$ dimensional grid centered around the seed (initial occupied site)
for step = $1, \dots, Ns$ do
    Calculate which are the unoccupied sites neighbouring any occupied site $nn_{free}$
    Add a particle in a site chosen randomly among $nn_{free}$ ▷ A new particle is added at each step
end for
```

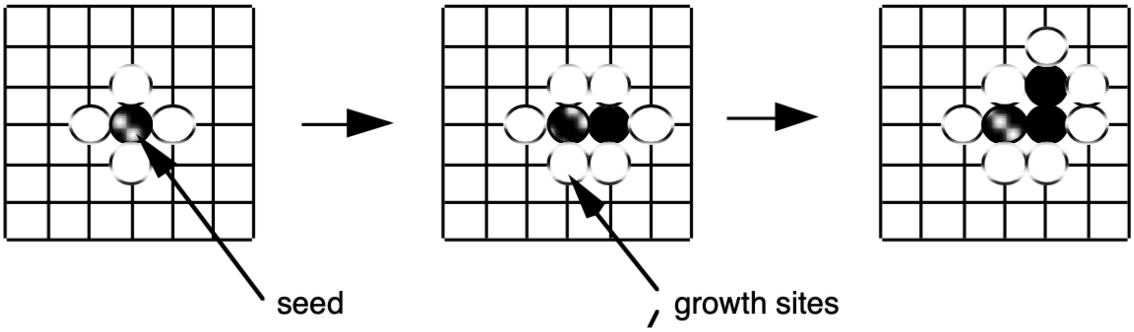

Figure 1: Representation of the first two evolution of the growth model.

1. Implement the growth model described in Algorithm 1 for $N = 2$ (i. e. use a 2D grid). Plot the position of the particles after 10, 100, 1000, 10000 and 100000 steps.

2. The nature of the clusters can be quantified by calculating their fractal dimension, d_f, defined as

$$m(r) \propto r^{d_f} \tag{6}$$

   where $m(r)$ is the mass of the cluster contained within a radius r from the seed. Calculate d_f for the clusters plotted as per point 1 and discuss the differences.

3. So far, the probability of populating a new site was independent of the number of occupied neighbours. Repeat the calculation including a higher population probability for sites with more occupied neighbours, n. Set such probability to $n/6$ and use a Metropolis acceptance probability to accept or reject the new particle accordingly.

Schrödinger dynamics: Scattering on a potential barrier

The task is to computationally study the scattering of a wave packet on a potential barrier. You need to solve the time-dependent Schrödinger equation (TDSE):

$$i\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2} \tag{7}$$

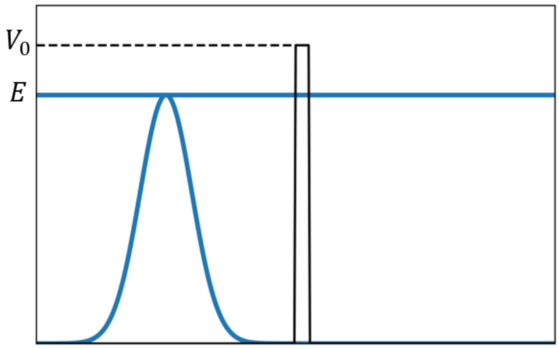

Figure 2: Schematic illustration of the wave packet and potential barrier.

The initial wave function is given by the normalized Gaussian wave packet:

$$\psi_0(x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{4}}} \cdot e^{-\frac{(x-x_0)^2}{4\sigma^2}+ikx} \tag{8}$$

The average momentum and energy of the wave packet are then given by:

$$\bar{p} = \hbar k, \qquad E = \frac{\bar{p}^2}{2m} = \frac{\hbar^2 k^2}{2m} \tag{9}$$

For convenience, adopt the "natural units": $\hbar = m = 1$. The potential is given by $V(x)$:

$$V(x) \;=\; \begin{cases} 0, & x < 0,\ x > a \\ V_0, & 0 \le x \le a \end{cases} \tag{10}$$

1. Implement a finite difference scheme to solve TDSE on the interval $[-L, L]$:

$$\psi_k^{m+1} = \psi_k^m + \frac{i}{2}\frac{\Delta t}{(\Delta x)^2}(\psi_{k+1}^m - 2\psi_k^m + \psi_{k-1}^m) - i\Delta t V(x_k)\psi_k^m \tag{11}$$

   Keep in mind that solving TDSE implies dealing with complex numbers. You can read about their representation in Python elsewhere on the web. However, for our purposes using `numpy` would be still enough, just pay attention to the data types of the variables.

2. Initial and boundary conditions are:

$$\psi(0,x) = \psi_0(x), \qquad \psi(t,-L) = \psi(t,L) = 0 \tag{12}$$

   Parameters of the problem: $\sigma = 2, x_0 = -10, k = 5, \Delta t = 2\cdot 10^{-5}, \Delta x = 0.1, a = 1, L = 20, t_{max} = 5$.

3. Study the scattering of a Gaussian wave packet on a potential barrier for $V_0 = 4,\ 8,\ 12,\ 13,\ 15$. Plot $|\psi(t,x)|^2$ as a function of x at $t = 0, 1, 2, 3, 4$. To visualize your solutions you may want to use the `Animator` function which you already know from our classes. If you want to use it, modify it to draw only every n-th frame to have ∼100 frames in total (otherwise, it will take too long).

4. Continue all runs until the solutions clearly describes two well-defined reflected and transmitted parts (but before they hit the walls of simulation box). At this time, determine the transmission and reflection probabilities, defined respectively as:

$$T = \int_a^L |\psi|^2 dx, \qquad R = \int_{-L}^0 |\psi|^2 dx \tag{13}$$

   Use your favorite method of numerical integration (e.g. trapezoid method) for this. Check whether the equality $T + R = 1$ holds true in your calculations.

5. Plot the analytical expression for T as a function of $\varepsilon = E/V_0$ (check your favorite Quantum Mechanics textbook for the formula). Add to the plot 5 points corresponding to the runs you have just completed and compare the results.

6. Write a report summarizing your work. Include all plots and data that you find relevant to make the report understandable. Analyze the results obtained from the physical perspective. What would we observe for a classical particle?

2D Ising model

Consider a square lattice of $N \times N$ spins σ_{ij}, each of them taking the value 1 or -1 marking if the spin at site (i, j) is up or down. The nearest-neighbour Ising model assumes that interactions only exists for the adjacent spins; the energy of a particular spin configuration $\{\sigma_{ij}\}$ is then given by:

$$E(\{\sigma\}) = -J \sum_{i,j}{}' \sigma_{ij}\sigma_{i'j'}, \qquad J > 0, \tag{14}$$

where a prime near the sum indicates that summation goes only over the nearest neighbors (in other words, $|i - i'| \leq 1, |j - j'| \leq 1$). The total magnetization of the lattice is given by a sum over the lattice:

$$M(\{\sigma\}) = \sum_{i,j}^{N} \sigma_{ij}. \tag{15}$$

In contrast to the 1D case, in two dimensions Ising model predicts spontaneous phase transition to the ferromagnetic state below the critical temperature T_c:

$$T_c = \frac{2J}{ln(1 + \sqrt{2})} \approx 2.269J. \tag{16}$$

That is why 2D Ising model is more important and interesting to study.

1. Implement the Metropolis algorithm solving 2D Ising model. The algorithm was explained in Lesson 9. Assume that our lattice is under periodic boundary conditions (PBC) in both i and j directions.

2. Put $J = 1, N = 100$ and run dynamics for the temperatures $T = 0.5T_c,\ T_c,\ 1.5T_c$. You should continue your runs until your system is equilibrated. To track this, calculate the energy per particle E/N^2 and magnetization per particle M/N^2 at every step. Then plot these quantities vs step number to see the equilibration of at least one of them.

3. Repeat dynamics 5 times for every temperature and visualize the equilibrated spin distributions (for that you may want to use the `meshgrid` function from `matplotlib`). Average equilibrium values of E/N^2 over these 5 runs to give you $\langle E/N^2 \rangle$.

4. Include the figures with the most typical spin distributions (1-2 for each temperature) to your report and describe qualitative differences in the behavior of model observed for the three T considered. Make connections between the spin distributions you are obtaining and the physical phenomena to which they correspond.

5. Repeat 2-3) also for $T = 0.25T_c,\ 0.7T_c,\ 0.9T_c,\ 1.1T_c,\ 1.3T_c,\ 2T_c$. Based on your findings, plot the dependence of $\langle E/N^2 \rangle$ on temperature T.

Random walk vs diffusion equation

Random walk is the simplest model for stochastic processes. An elementary example is the random walk along the x axis which starts at 0, and at each step a "walker" moves +1 or -1 with equal probability.

1. Implement this 1D random walk model. Run $n = 1000$ steps for each of $N = 100$ walkers. Then average trajectories over all N walkers and ensure that $\langle x \rangle \approx 0$ and $\langle x^2 \rangle \sim t$, where t is the "time" of simulation.

2. Random walk model can be applied to simulate stochastic processes such as diffusion. Consider two "swarms" consisting of $M = 2000$ non-interacting particles each. The size of the simulation box is $[-L, L]$ with $L = 200$. All particles in the first swarm are initially located at $x = -50$, while the on in the second are all at $x = 50$. We ask you to model simultaneous diffusion of these two swarms by random walks (1 particle = 1 random walker). As before, at every step a particle can move +1 or -1 with equal probability. Run for $t_{max} = 2000$ steps and compute the distribution of particles along x (e.g. using `numpy.histogram`). Plot the computed distributions after $n = 0, 100, 500, 1000, 2000$ steps.

3. The same physical process can be described at another level by solving the equivalent diffusion equation for concentrations:

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}, \quad c(0,x) = c_0(x), \quad c(t,-L) = c(t,L) = 0 \tag{17}$$

$$c_0(x) \;=\; \begin{cases} 0, & x \neq \pm 50 \\ M, & x = \pm 50 \end{cases} \tag{18}$$

   Using your knowledge from our course, implement the finite differences scheme to solve this equation. Take diffusion constant $D = 0.5$. You may use the same integer grids for time and x as in 2).

4. Solve the equation and take snapshots of $c(t,x)$ as a function of x at $t = 0, 100, 500, 1000, 2000$. Compare with the respective random walk snapshots from 2). For this, make plots containing both random walk and diffusion equation snapshots at the same t.

5. Analyze the results obtained and draw your conclusions. Which method performed better and why?

Shooting method for the quntum harmonic oscillator

Consider the time-independent Schrödinger equation (TISE) for the one-dimensional harmonic oscillator:

$$-\frac{\hbar^2}{2m}\frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x), \tag{19}$$

with the quadratic potential

$$V(x) = \frac{1}{2}m\omega^2 x^2. \tag{20}$$

The boundary conditions in this case are $\psi(x \to \pm\infty) = 0$. The analytical solution gives the following energetic spectrum:

$$E_k = \hbar\omega(k + 1/2), \quad k = 0, 1, 2, \ldots \tag{21}$$

For simplicity, set $\hbar = m = \omega = 1$.

1. Since the potential is symmetric, the shooting method can be used to solve the Eq. (1) in the same way as it was done for infinite well. Set the initial conditions $\psi(0) = 1$ and $\psi'(0) = 0$ (even solutions) and use the shooting method to evolve the solution until $x = L > 0$ such that $\psi(L) \approx 0$. Use a small but positive value of E as an initial guess, for example $E = 0.1$. L should be far enough to provide that $V(L) \gg E$.

2. Compute the first eigenvalue with a precision of $\varepsilon = 10^{-5}$ for several values of L. You should observe that there is a region of the space x in which the solution rapidly decays to zero, and then it diverges to positive or negative values. However, in between $x = 0$ and $x = L$ you should have the correct eigenfunction $\psi(x)$. Based on these considerations, choose the best value for L. Compare your E obtained with this L to the analytical eigenvalue for the ground state.

3. Now try to obtain the solutions by shooting method for $k = 1, 2, 3$. Keep in mind that for different k you will have different parities, so that you have to choose the proper (even or odd) initial conditions. Make a table that compares the exact eigenvalues with those obtained numerically. Do your values of E_k correctly reproduce the energetic spectrum of QHO?

4. The wave functions you have calculated are not yet normalized. Use your favorite numerical integration method (e.g., trapezoid method) to normalize your numerical wave functions for the first three states. Plot them along with the analytical solution on one plot (separate plot for each k). For analytical wave functions you may check e.g. your favorite Quantum Mechanics textbook.