

Final Project Proposal

molecular-dynamics

Pavan Ravindra (Uni: phr2114)

Algorithm description: My general project idea is to implement a molecular dynamics simulation code to simulate the motion of simple particles. Molecular dynamics is a popular type of physical simulation. The idea is that we represent “molecules” as a set of interacting point particles. Before we start the simulation, we decide on a “force field” that dictates how the particles interact with each other. The force field essentially describes how the particles push and pull on each other. I will further elaborate on the choice of force field below.

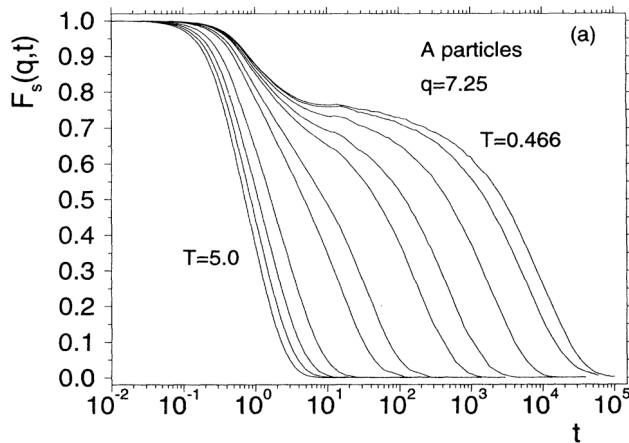
The procedure of a molecular dynamics simulation is roughly as follows:

1. Choose a force field F that determines how your particles interact with each other.
2. Choose the initial positions of your N interacting particles.
3. Repeat until you reach your simulation time of interest:
 - (a) Calculate the forces on each particle. For each of the N particles, they interact with all of the other $N - 1$ particles (in principle), so the runtime of this step is $O(N^2)$ where N is the number of particles.
 - (b) Update the positions of each particle based on these forces. The runtime of this is just $O(N)$.

Hence, the overall runtime of a molecular dynamics simulation is $O(N^2t)$, where N is the number of particles, and t is the simulation length that you want. However, we cannot parallelize along the time dimension, since we need to calculate each timestep before we can calculate the next timestep. This means that the major parallelization effort will come from parallelizing the calculation of the forces. This means that if I choose too few particles to simulate, I won't really see any parallelization benefits. My rough plan is to keep increasing the number of particles N until I see a major parallelization benefit.

The main objective for my project will be to reproduce Figure 2 from Ref. [1], shown at the top of the next page. This seminal work from 1995 was one of the first molecular dynamics calculations. It is extremely relevant to my own PhD research in Chemical Physics, which is why I chose this project topic.

I'll try to give a rough idea of what this figure means, just for fun. The different curves shown comes from simulations at different temperatures. We



generally use T to denote temperature. Low temperatures correspond to slower moving particles, which higher temperatures correspond to faster moving particles. As you go from left to right, the curves go from hotter temperatures ($T = 5.0$) to colder temperatures ($T = 0.466$). The x-axis is time, and the y-axis can be thought of as a measure of how far particles have moved from their starting positions. As particles move further from their starting positions, the $F_s(q, t)$ curves decrease from 1.0 to 0.0. Hence, the main message of this figure is that particles move away from their starting positions faster at high temperatures than at low temperatures. The interesting behavior though is the emergence of this intermediate plateau at the lower temperatures. This is a hallmark of so-called glassy dynamics, which is the topic of my PhD research.

The simulation protocol will be taken directly from Ref. [1]. There are two ways that I can increase the computational demand of the project in order to emphasize the benefits from parallelization. Since the compute-intensive step comes from calculating the $\sim N^2$ interactions between pairs of particles, I can always increase the number of particles in my simulation to further display the benefits of parallelization. Similarly, I could also go to lower temperatures, which would require longer simulations to observe the intended decay (as explained in the above paragraph).

Broad plan for parallelization: In terms of parallelizing my code, my main plan is to use the Par monad to parallelize the calculation of the $O(N^2)$ forces at each simulation timestep (refer to the above discussion). There's also a caveat with all of this, which is that molecular dynamics simulations often use a cutoff distance for particle interactions. The general idea is that particles that are further than some distance away from each other rarely impose a significant interaction, so sometimes we just ignore particle interactions beyond some distance determined by the force field. I'm going to mess around with this once I actually start coding things up with the Par monad to see what strategy makes the most sense.

I also suspect that I might be able to use REPA to make things faster, since I'll be using some kind of array to store current positions (and velocities) of each particle. However, I actually think this might not really speed things up because I'll only ever be doing linear in-order traversals through these arrays, but I'll have to think about this more.

Input Data: The only two pieces of input data I need are the choice of force field and the initial positions of the particles. For the choice of force field, I'll be using the same force field as in Ref. [1]. For two particles at a distance r away from each other, the Lennard-Jones interaction potential has the form:

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

The σ and ϵ variables are just parameters that are specified in Ref. [1]. Things are complicated a little by the fact that there are two different particle types, but this is just a technical detail.

For the initial positions, I'll use a common trick that at high temperatures, particle positions "equilibrate" (or stabilize) very quickly. So I can just use a high temperature simulation to get a reasonable set of initial positions, and then I can just lower the temperature from there. The exact details of this are probably too technical to warrant going into here, but I'd be happy to elaborate on this if needed.

Reason for wanting to work alone: I know that we are expected to provide a reason for wanting to work alone if we choose to do so. Since I'm coming from the Chemical Physics department, I really wanted to work on a project that is more closely related to my research, since that seemed most interesting to me. I didn't really think anyone else in the class would enjoy working on something like this, so I figured working on my own would be best.

References

- [1] Walter Kob and Hans C Andersen. Testing mode-coupling theory for a supercooled binary lennard-jones mixture. ii. intermediate scattering function and dynamic susceptibility. *Physical Review E*, 52(4):4134, 1995.