Project Report

Computer Modelling

Due: 16:00 Thursday, Week 10, Semester 2

1 Aims

Based on the N-body code you co-wrote, you should run a series of representative simulations to obtain data. You should contrast your results with the expected outcome and, where possible, compare them with the literature. Based on these results, you will write and submit a *"computational report"* of up to 1600 words[1].

2 Tasks

The goal of a scientific report is to communicate the results of an experiment or a simulation, a development of a theory, or a hypothesis; with enough detail and information so others can independently judge the (de)merits of your work. It is important to include enough information so your results can be independently reproduced. In a computational physics report, one would state the main maths/physics behind the simulation, the main physical and numerical approximations made, as well as the numerical parameters of all relevant simulations presented.

Good quality computational physics requires *convergence*. A simulation has converged if more accurate simulations will not meaningfully change results. In exercise 2 you computed the frequency of a Morse dimer with various dt. In the report you will show that the results of a *representative simulation* have converged.

You shall assume the reader is a competent coder, who is familiar with the main algorithms used, and who can represent math and algorithms as code. Therefore, the main report should *include no code*. Typically, reports over the word limit are too detailed in their code description.

Lastly, good reports will condense the main result in an abstract, set the background of the physics in the introduction in an attractive way, and have a clear conclusion.

[1] Shorter reports will not be penalised, as long as all key points are discussed

2.1 Project Report

Prepare your report according to the following guidelines:

- Indicate at the top which project you have attempted
- Your report should contain the following sections:
  - An **abstract**, listing the physics of what you have done, and the main result,
  - an **introduction** of the physics at play, together with a *physical*, not programmatical, description of your code,
  - a description of novel, non-trival algorithms you may have devised,
  - main **results**, including (see section 3 for details):
    - proof of convergence,
    - verification of the code,
    - one mini-task from Section 3 and,
    - a brief discussion of your results. And finally,
  - **conclusions** and post-mortem. State the main conclusions you have arrived at, and whether (and how) your simulation was satisfactory. Finally, briefly reflect on project: were there any problems? Was the design document useful[2]? Is there something you would now do differently?
- Your code may need substantial fixes after the source code submission. If that is your case, document important changes in an Appendix.
- Include screenshots of your simulations and tables and graphs of relevant quantities wherever they are helpful to underpin your discussion.
- Length limit is 1600 words, not counting abstract, tables, references or appendices (if applicable).

The report should read similar to a lab report, and be less technical than the Design Document. You should not need to write *any* code outside critical changes in Appendix A. We aim for replicability, and detailing the physics and the maths behind the simulation. For physicists, code is a tool; it should not be given top priority.

3 Project-specific Instructions

3.1 Project A

Your report should touch upon the following questions:

[2] *"no"* is a valid answer, but must be justified.

- If your code allows it, simulate a *toy* solar system (the Sun, Mercury, the Earth and the Moon for about 10 years). Test the convergence of the observables (apsides, periods) with respect to the timestep.
- Infer, from the toy system, the largest timestep you can use to qualitatively simulate the Solar System. What are the limiting factors influencing your value?
- Now show your simulation is a reasonable representation of the actual solar system by comparing your apsides and periods to observational data.

Your results should show now your simulation is representative, accurate and is a reliable predictive tool. Therefore, explore ***one*** of the following issues:

- Verify Kepler's third law for the planets using *your* results. If you replace Jupiter by a *"super-Jupiter"* 20 times heavier, is Kepler's law still true?
  Hint: Power laws are often well-suited to log-log graphs. All fits should be shown with an estimate of their quality.
- A 300–year simulation will show a Halley's comet following an open trajectory. Note how close Halley comes to Venus in 2061. Is Halley's open orbit an artefact of your simulation or is it caused by other planets? Which planets cause the largest perturbation of the orbit? You should remove planets from your simulation at will in order to reach a conclusion.
- *If your code is unsuccessful*: Discuss from a physics point of view, what may be wrong with the simulation, and what potential steps you could follow to fix it.

3.2 Project B

For your report, you should first analyse the convergence of the results with respect to dt (RDF, MSD, energy conservation) for the following *representative* systems:

- Solid: 32 atoms, $T = 0.1\ \epsilon$, $\rho = 1.0\ \sigma^{-3}$ and $r_c = 2.5$, for 10 $[t]$[3].
- Gas: 30 atoms, $T = 1.0\ \epsilon$, $\rho = 0.05\ \sigma^{-3}$ and $r_c = 3.5$, for 10 $[t]$.

A reasonable dt set would be 0.05, 0.01 and 0.005 for the solid, and 0.01, 0.005 and 0.001 for the gas. Based on those results, you should choose a timestep for the solid & liquid, and a timestep for the gas for the next simulations[4]. Now show your code can roughly reproduce the Argon phase diagram on more accurate calculations:

- Solid: number of atoms $N = 108$; LJ cutoff distance $r_c = 3.5$; 100 [t].
- Fluid: $N = 90$; $r_c = 3.5$; 100 [t].
- Gas: $N = 90$; $r_c = 3.5$; 100 [t].

[3]E.g. 1000 steps for $dt = 0.01$

[4]You may need to increase the initial temperature of the liquid by $\sim 20\%$

For each simulation, choose appropriate temperatures T^* and densitites ρ^* based on Argon's phase diagram given in the project instructions. Estimate the maximum timestep possible used to accurately simulate each system, and what your criterion is. For each phase, you should

- Plot the potential, kinetic and total energies as a function of timestep;
- Plot the MSD as function of timestep;
- Plot the RDF; discuss in terms of what is expected for each system.

Your results should show now your simulation is representative, accurate and is a reliable predictive tool. Therefore, explore ***one*** of the following issues:

- *Specific Heat.* At the same density, simulate the solid at several initial temperatures, ideally from close to 0 ϵ to about 1.0 ϵ. Plot now the total energy against temperature. Remember that $E(0) = V(t = 0)$. Compute the specific heat using

$$C_v \approx \frac{1}{N}\frac{\Delta E}{\Delta T}$$

  and discuss your results.

- *Melting Temperature.* Fixing the density at $\rho^* = 0.85$, run a series of simulations at $T/\epsilon = 1.0,\ 1.1,\ 1.2\ldots$. Melting can be seen by a sudden destabilisation of the MSD. At what initial temperature does your system melt? Melting should also be accompanied by a significant decrease of the kinetic energy. Can you relate the final average kinetic energy to the temperature? Discuss.

- *If your code is unsuccessful*: Discuss from a physics point of view, what may be wrong with the simulation, and what potential steps you could follow to fix it.

3.3 VMD tips

VMD is a powerful tool to analyse N-body simulations. Some of the functionality can help you analyse your XYZ trajectory for a variety of purposes, including complimenting missing functionality from your code.

- **Apsides:** VMD can measure distances from one body to another. For this press '2' (Mouse → Label → Bonds) and then click on two "particles". The distance at the current frame will be shown. Monitoring this will allow you to measure apsides
- **Periods** On the "Points" graphical representation, plot a trajectory of N steps (Representation → Trajectory → Draw multiple frames, input `0:N`). Knowing how much time is elapsed per XYZ step, you can estimate the period by seeing how many frames are needed to close an orbit.
- **System trajectory.** Plotting a full trajectory following the steps above will clearly show if your Lennard-Jones system is solid or not.

- **RDF.** Using the box length, VMD can effortlessly calculate the RDF for your trajectory. Click on Extensions → Analysis → Pair Radial Distribution Function. Select your trajectory as the *"molecule"*. On "Util", click on *Set Unit Cell Dimensions*, and input the relevant value. Then input `all` in both Selection boxes and play with the other parameters. Finally, click on *"Compute $g(r)$"*.

4 Submission

Submit your report through the Turnitin Link on the course LEARN page, by **16:00 on Thursday, week 10 of semester 2**. This is an individual task, to be completed by every student independently.

5 Marking Scheme

This assignment counts for 40% of your total course mark. It is made up of the following components, marked according to the rubric visible on the Turnitin assignment:

1. Abstract & Introduction [10]
2. Convergence tests [5]
3. Results. [20]
4. Discussion. [20]
5. Conclusions and post-mortem [10]
6. Report layout, language, graphics quality and usefulness [15]

Total: 80 points.