Part II Computational Physics Exercises marking guide

David Buscher <db106@cam.ac.uk>

v4.0 — 2024/05/16

1 Introduction

There are marks of 6, 6, and 8 respectively for exercises 1, 2, and 3. In the marking, these have been doubled to 12, 12 and 16, to allow greater discretion and reducing the need to use half-marks.

In the first two exercises, we are not marking for code quality, just for successful completion of the tasks. In the last exercise the code quality should enter into the marks as indicated later.

2 Marking levels

The following table gives guidelines for the marking levels you are aiming for:

	Ex1/12	Ex2/12	Ex3/16	Total/40
mean	9	9	11	29
std deviation	1.5	1.5	3	6
range (min–max)	6–12	6–12	6–16	18–40

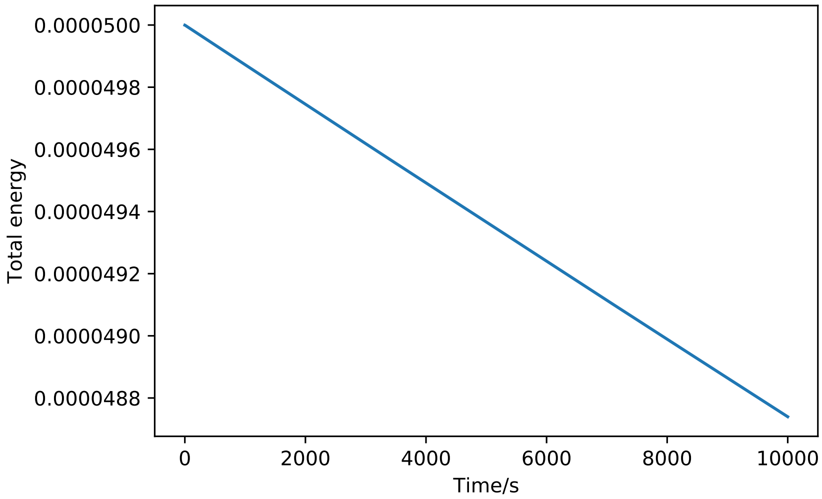
Clearly these are approximate, and you will expect some deviation due to sampling statistics. The range especially is indicative since these are not Gaussian distributions.

3 Exercise 1: pendulum

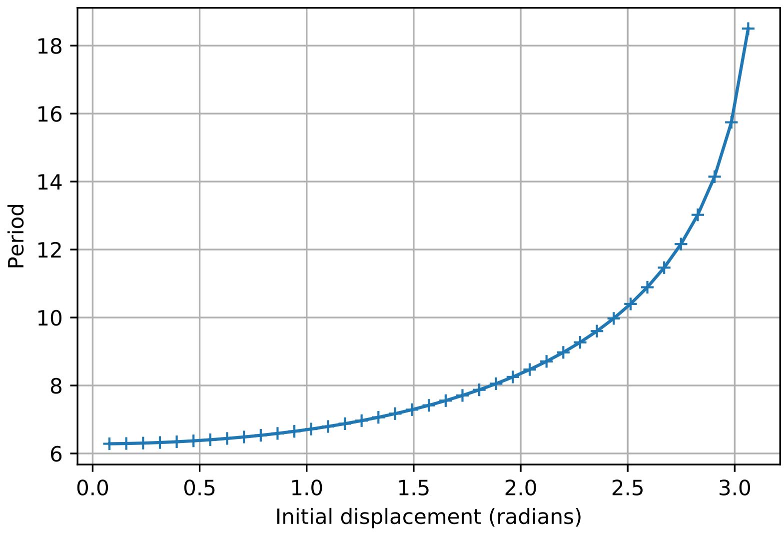
There are 12 marks for this question.

3.1 Core task 1

The core task is to get the code running correctly. Depending on the integrator, the energy plot should be stable to sub-percent values or better over long periods:



The plot of period against amplitude looks like:

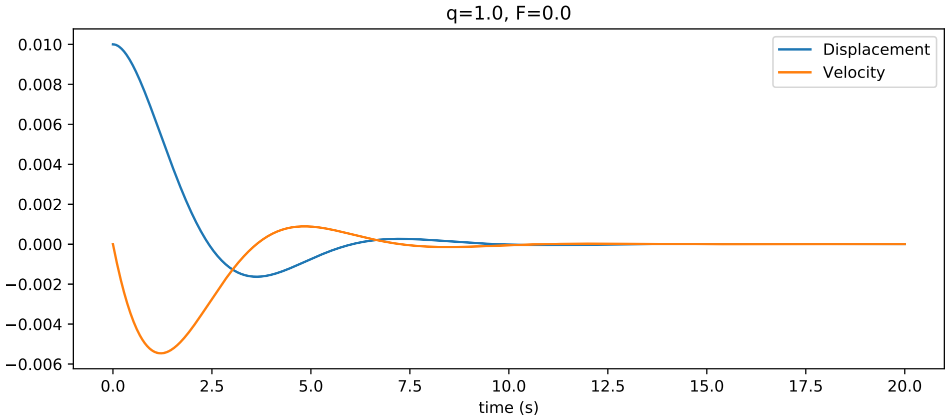


The period required is 7.42 seconds.

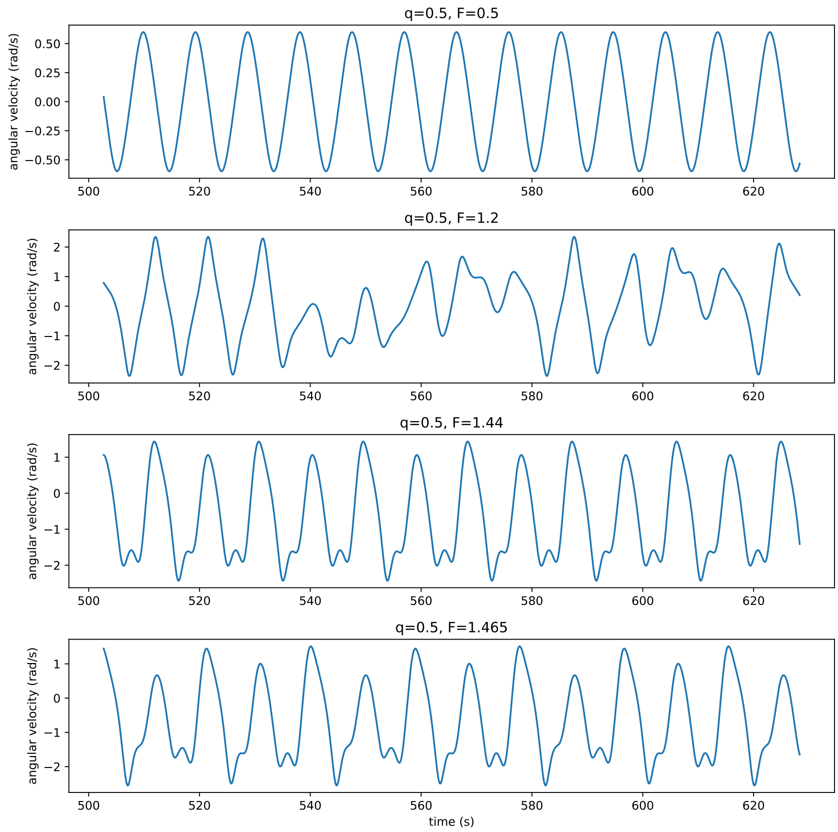
4 marks for a good job at this task.

3.2 Core task 2

The damped, unforced system should look as predicted from linear SHM, e.g.



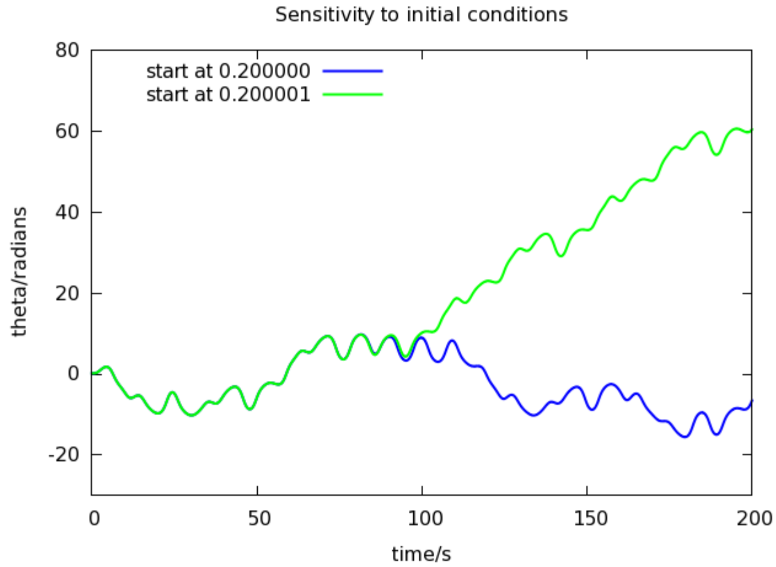
With forcing, we see chaotic behaviour for $F = 1.2$ and period doubling and quadrupling for $F = 1.44$ and $F = 1.465$ — another characteristic feature of chaos. Most people won't explicity spot the period doubling/quadrupling, but give generous credit for worthwhile attempts and doing something useful and accurate.



4 marks for a good job at this task.

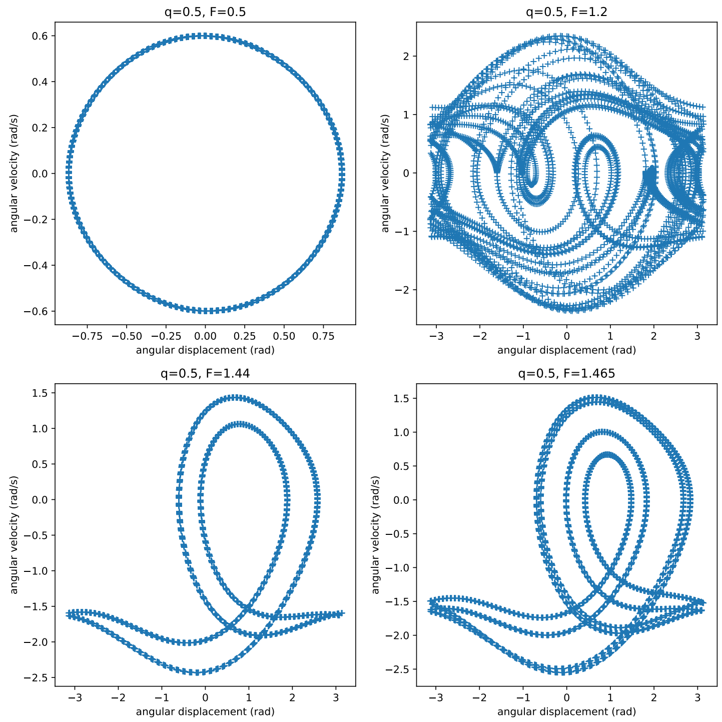
3.3 Supplementary Task 1

Here we see sensitivity to initial conditions



3.4 Supplementary Task 2

Phase-space plots give further evidence for period doubling and quadrupling.



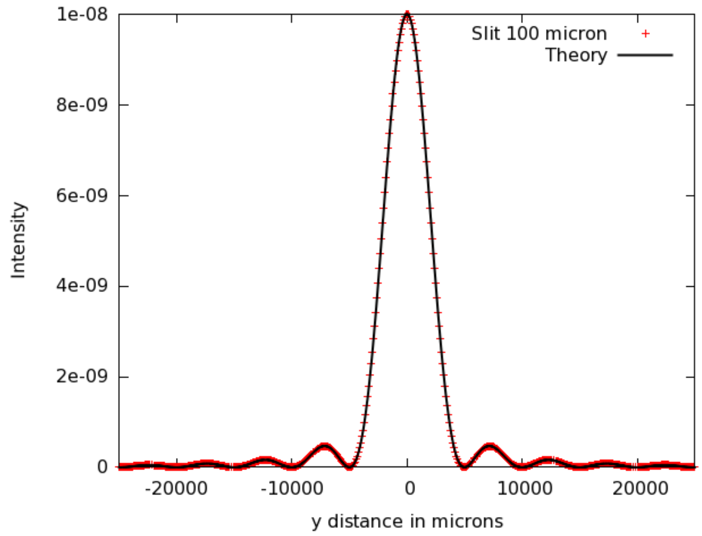
4 marks for the supplementary tasks: the student won't have to do all of the above to get full credit.

4 Exercise 2: diffraction

There are 12 marks for this question.

4.1 Core task 1

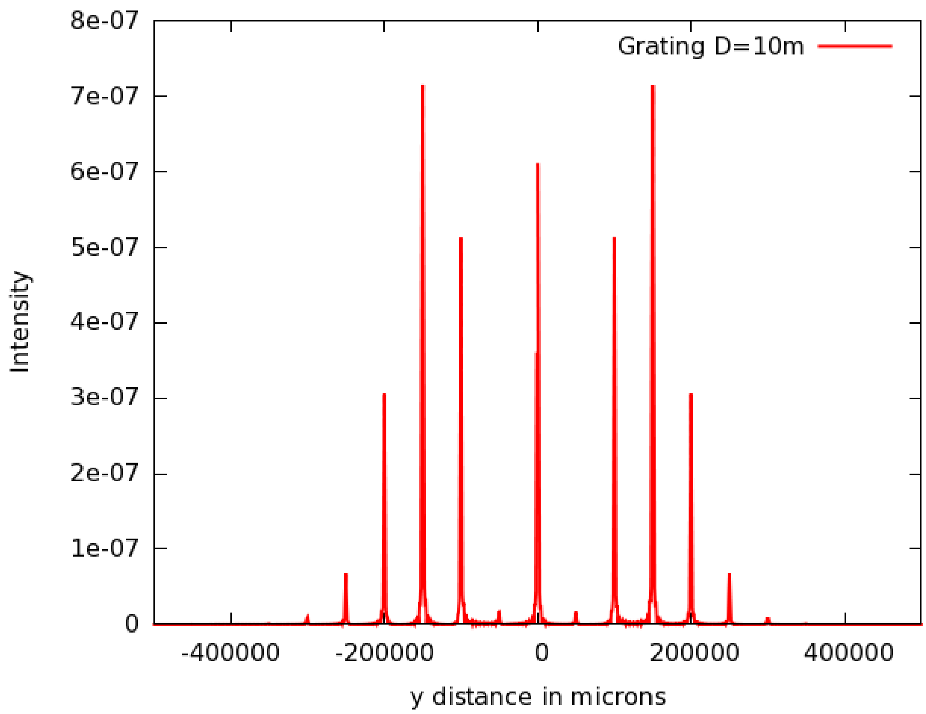
Core task 1 is to find this single slit pattern:



2 marks for a good job at this task.

4.2 Core task 2

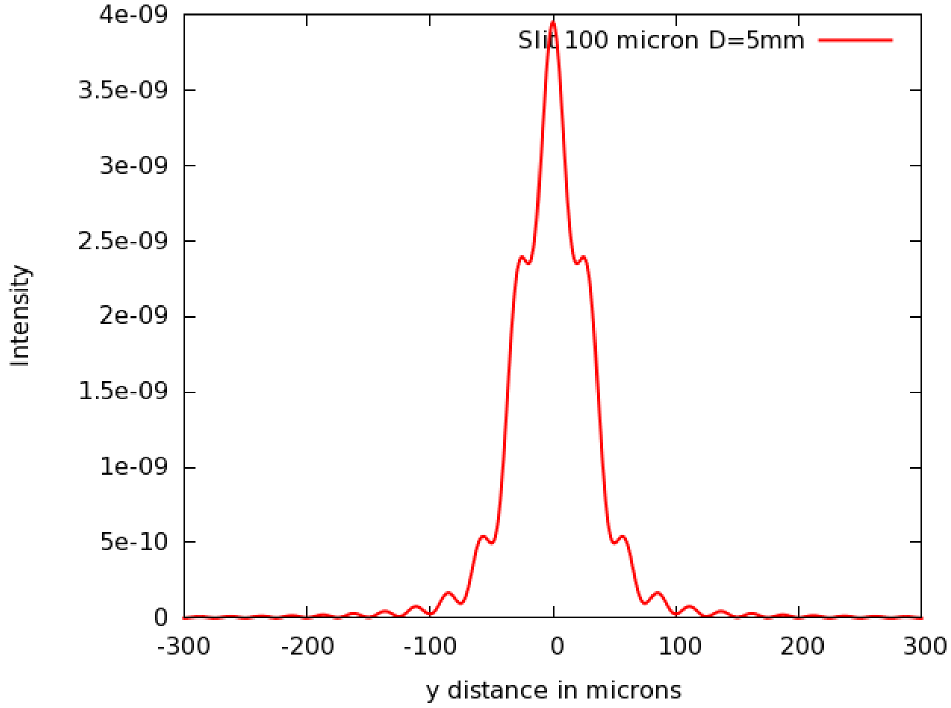
Core task 2 is the pattern of a sinusoidal grating: note the missing diffraction order.



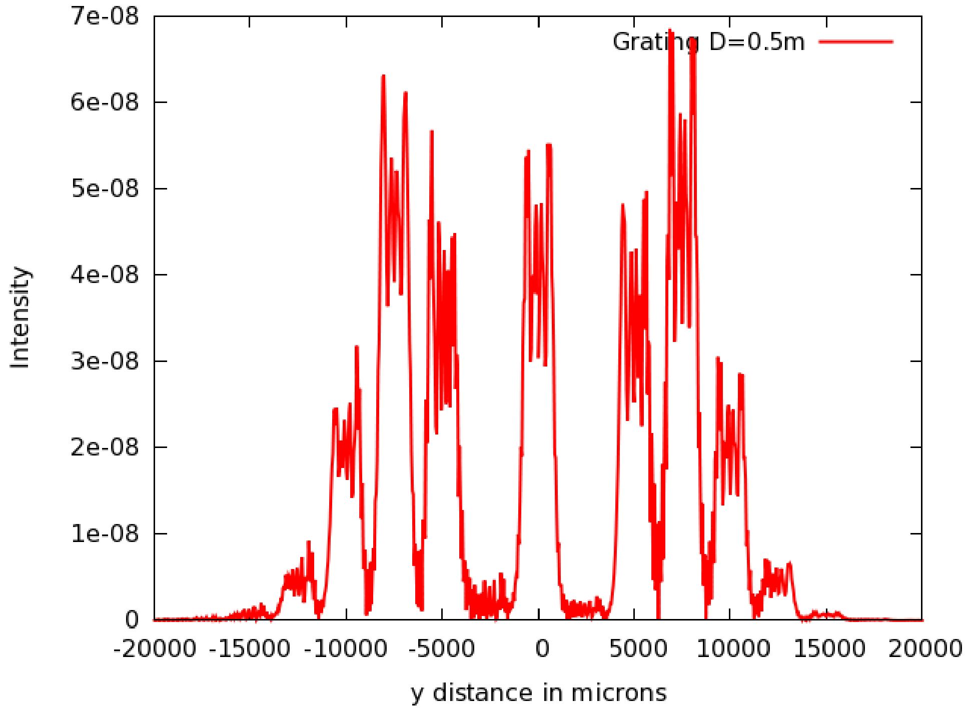
3 marks for a good job at this task.

4.3 Core Task 3

In the near field, these become



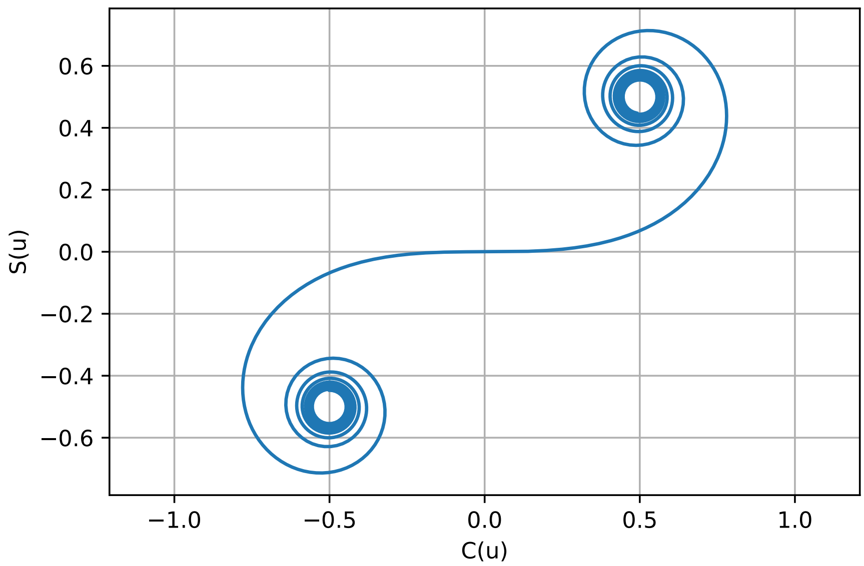
and



3 marks for a good job at this task.

4.4 Supplementary Task 1

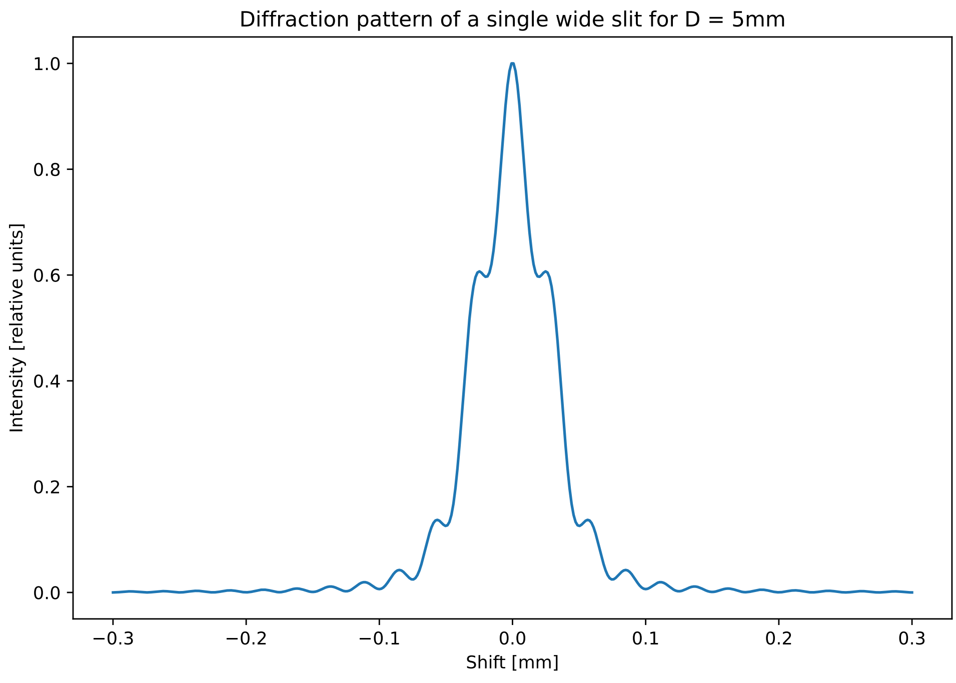
This task should produce plots something like this for the Cornu spiral.



2 marks for a good job at this task.

4.5 Supplementary Task 2

This task should produce a result in good agreement with the result from Core Task 3



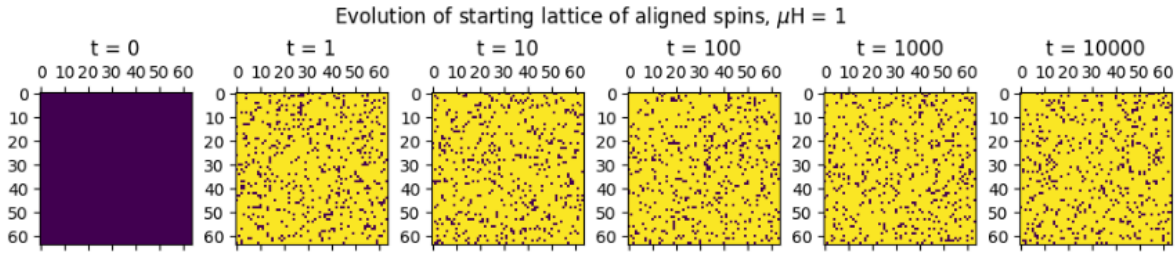
2 marks for a good job at this task.

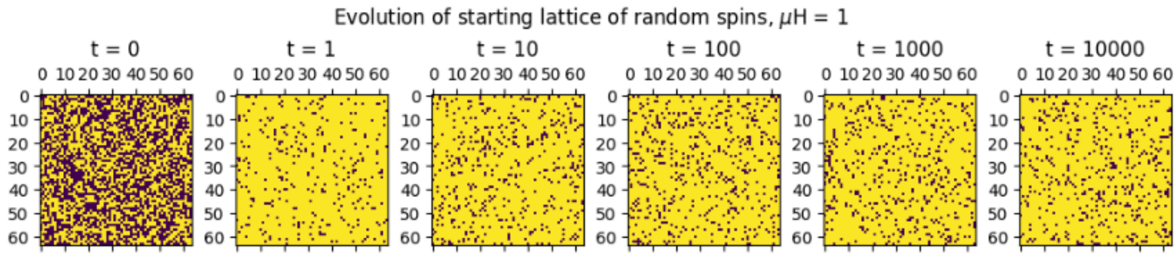
5 Exercise 3: Ising model

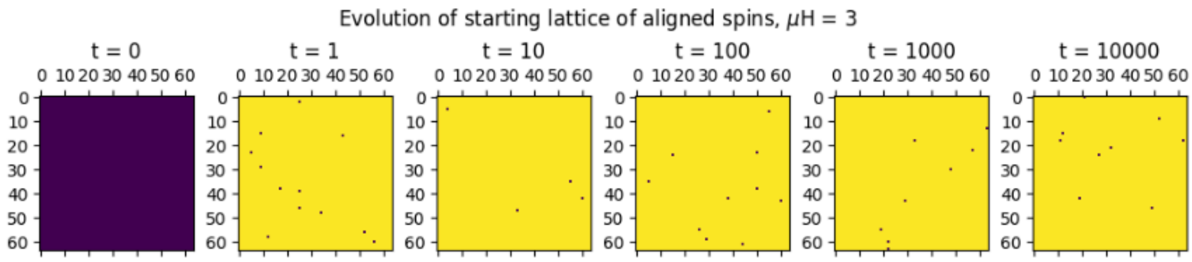
There are **16 marks** in total for this question. Note that this includes marks for code quality — see the last section.

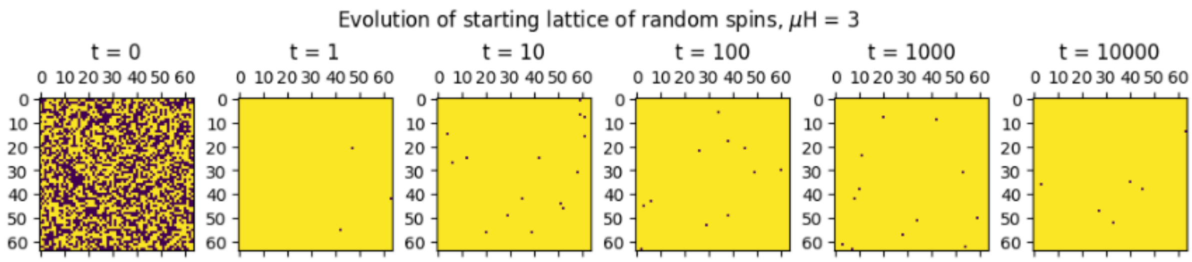
5.1 Core task 1: No spin coupling

The with no coupling the equilibration of the lattice is rapid, independent of the starting conditions.

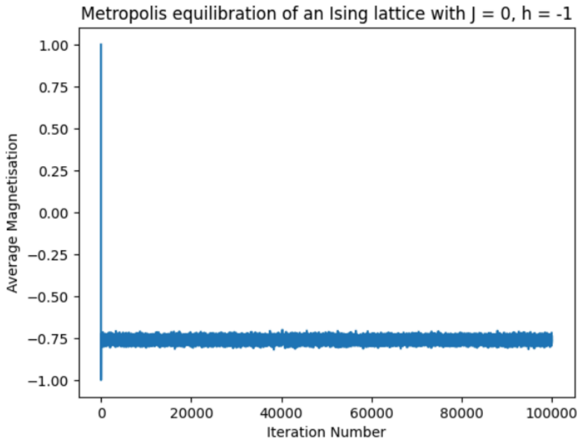




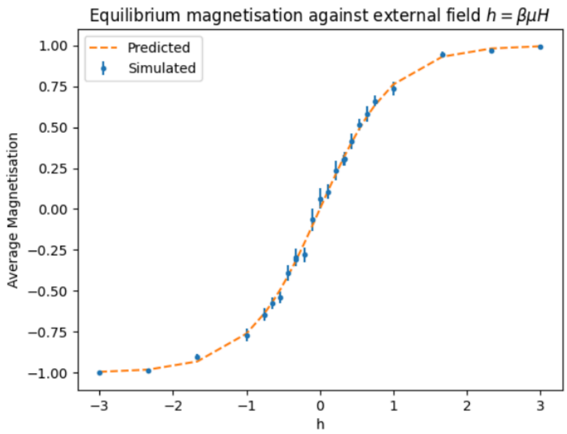




The mean magnetisation stabilises rapidly and then fluctuates around the mean for the rest of the evolution.



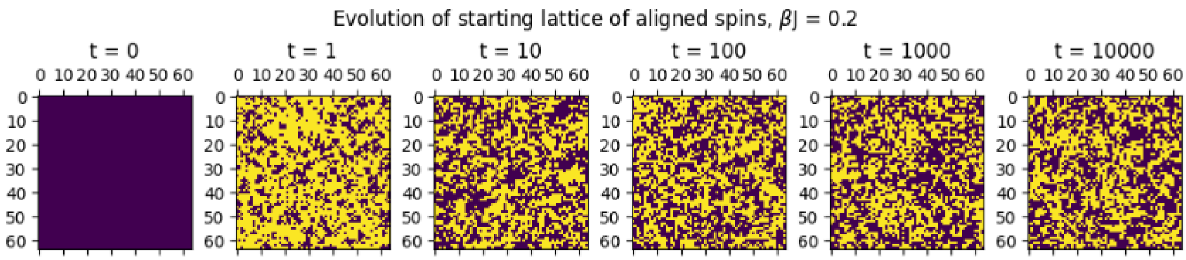
This leads to good agreement with theory for the magnetisation vs field curve with relatively small random errors:

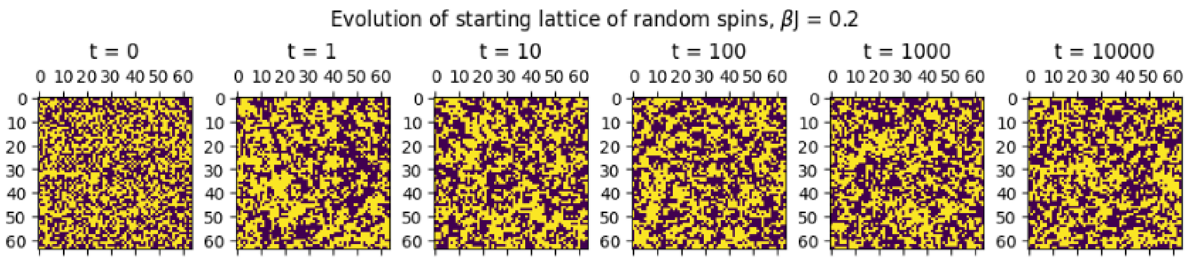


4 marks for a good job at this task, including commenting on their results.

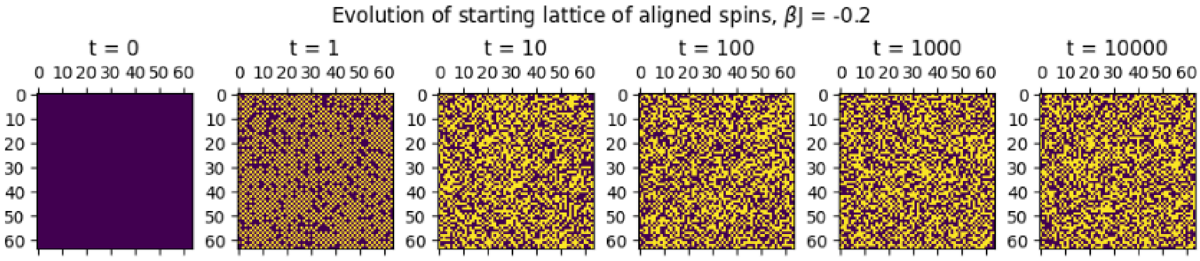
5.2 Core task 2: With spin coupling

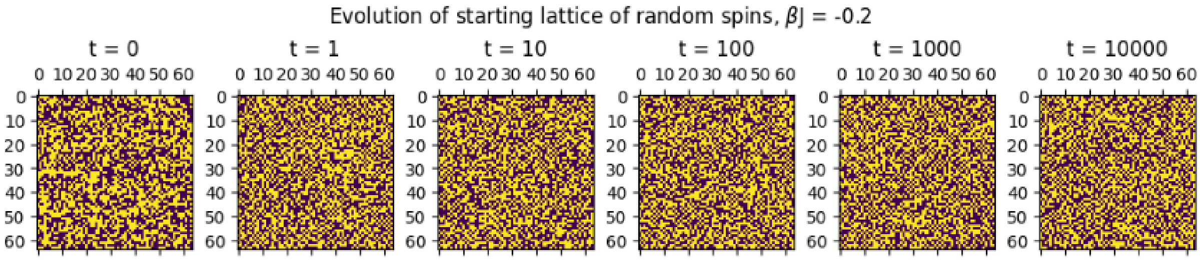
With βJ positive (ferromagnetic) then there is a tendency for spins to align and so spins cluster together in clumps:





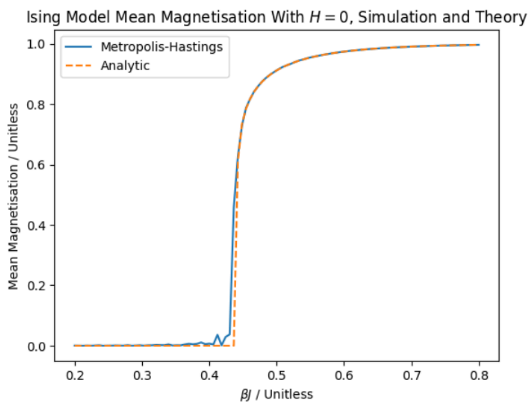
With βJ negative (anti-ferromagnetic) then there is a tendency for the spins to anticorrelate, giving a fine-grained pattern with little clumping:

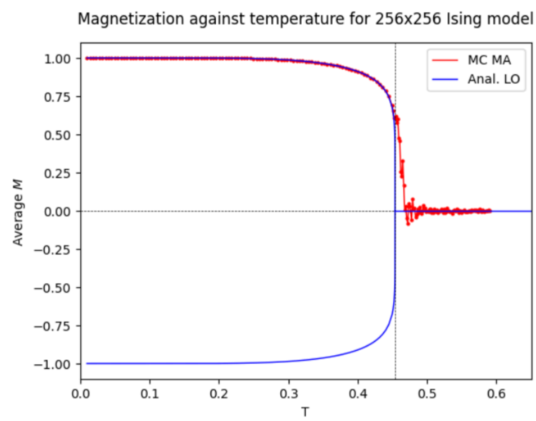




The mean magnetisation follows the Onsager formula except close to the transition temperature (the agreement is likely to get worse for smaller lattices).

The Onsager formula bifurcates at the critical temperature so it is possible to get both positive and negative solutions for the magnetisation, dependent on the starting conditions. The solution is noisy close to the critical temperature, likely because of this possibility to "flip" between the positive and negative magnetisation.



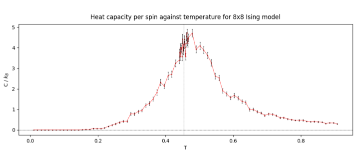


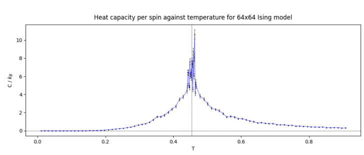
(Note that one of the above graphs shows β as the x-coordinate, the other uses T — either is acceptable).

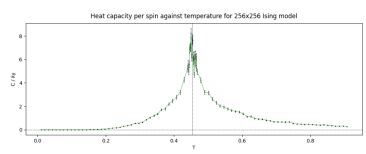
4 marks for a good job at this task, including commenting on their results.

5.3 Supplementary task: heat capacity

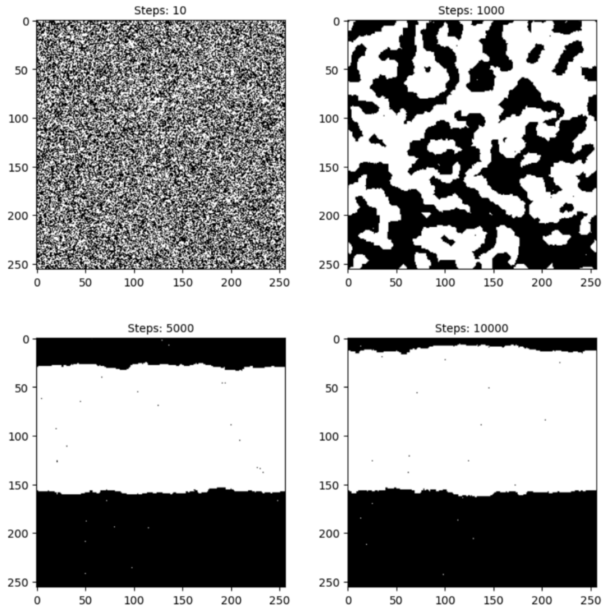
The heat capacity peaks near the transition temperature and the peak gets sharper and closer to the transition temperature predicted for an infinite lattice as the lattice size gets larger.







The graph tends to get noisy close to the transition temperature because the spins form very large and long-lived "clumps" close to the phase transition; this causes large fluctuations in the measured quantities which average out slowly (students don't necessarily need to notice this to get full marks, it just helps to explain why you may see some very noisy graphs).



4 marks for a good job at this task, including commenting on their results.

5.4 Code quality and optimisation

Have a brief look at the source code and award **2 marks** for high-quality code. High-quality code should include at minimum:

- Structured code: tasks broken down into sensible functions;
- Meaningful function names;
- Meaningful variable names (this is less important than for naming functions: single-letter variable names can be meaningful if the meaning can be inferred from context, e.g. loop counters);
- Appropriate levels of commenting (at minimum identifying what each function does, preferably using doc-strings);
- Sensible use of whitespace to indicate code structure (somewhat obligatory in Python, but e.g. using line spacing to separate functions is something to look for).

The key to getting accurate answers is to make the code fast, through vectorisation and other techniques such as parallelisation. Award up to **2 marks** for indications that some thought has been put into such optimisations to allow running of larger lattices and averaging over larger numbers of steps.

4 marks total for these two aspects of the code