ASSIGNMENT 1+2

Issued on 20 September 2019

Simulation and analysis of the Ising Model

Complete the following in your workbook in the style and according to the criteria which are set out in the separate outline.

Process: Just before hand-in of the intermediate Assignment 1, you should please stick in a copy of whatever you have programmed up to that point into your workbook, even if it is incomplete. Before handing in your final workbook as Assignment 2, you should again stick a copy of the program code in your workbook and provide me with an electronic copy. No graphics or other output need be provided electronically: print whatever graphics and/or data you want me to consider for marking and stick it into your workbook. Consult me regarding possible inclusion of special software or utilities.

Marks allocated and deadlines: Assignments 1 and 2 each count 25% of your final module mark. The two hand-in dates for these assignments will be determined in class.

Plagiarism will be penalised as set out in the course outline. Write your own code and generate your own results even if you look what other students are doing. **Cite your sources** including books, literature, internet links etc.

Content: The list below starts very prescriptively and becomes progressively more vague and open. The idea is that you **must** take the first few steps as a matter of survival, while in the later stages there is increasing scope for different choices, viewpoints and options.

Approach to Ising model simulation

(a) Read at least the following sections of Newman in the course of the next few weeks: 3.1, 3.2, 3.3 (without 3.3.2) and 3.7.

(b) Understand the theory and practice of the spin-flip algorithm as implemented in the Newman C-code fragment which is provided to you.

(c) Set up your program to use random numbers generated from a choice of fixed seed or time-dependent seed. Ensure your random number generator is properly initialised.

(d) Start your programming with a **fixed** dimensionless temperature kT/J, which we just call "T" from now on, between values approximately 1.0 and 4.0 which you specify when your code starts to run, and a lattice size parameter L which is fixed for the time being.

(e) Implement different initialisations of the lattice for the cases $T < T_c$ and $T > T_c$ and the calculation of the initial energy and magnetisation. Check a few configurations by hand.

(f) Flip *one* spin and ensure that your energy updating works correctly. Only then should you implement a full set of sweeps.

(g) Get the approach to equilibrium under control by looking at the time development of the magnetisation and plan for capturing τ_{eq} for various scenarios.

(h) Look at the dependence of the equilibration time on the lattice size. First consider a few temperatures far from T_c and then concentrate on temperatures near T_c.

(i) Briefly study the behaviour of the Ising model for lattice size $L \leq 20$ as a function of time for a number of different temperatures.

(j) Look at a few temperatures for $L = 100$ or more if your code can handle it. Look at temperatures above and below T_c.

(k) Consult the Physics 344 lecture notes and Newman on the **time autocorrelation** of the magnetisation M, and plan in your book how to code this for the case where the number of datapoints is too large to store in an array. Then code it and use it to find correlation times τ for given L, T.

Then continue along the following lines while remembering that you can try own ideas. Discuss ideas not listed below with me and others beforehand to prevent time-expensive wild goose chases.

(a) Choose a fixed lattice size (100 or more, unless you run python or other slow software) and a set of temperatures at which you want to sample the magnetisation phase diagram. Find and implement realistic values of the equilibration time τ_{eq} for each of your chosen temperatures.

(b) Find realistic correlation times τ for each chosen temperature and implement them in your code.

(c) Write code to accumulate a statistical sample of *independent* magnetisations and energies with the help of τ_{eq} and τ for a given temperature, then proceed to find e.g. sample means and standard errors.

(d) *Once the above is working*, do a full analysis of the **phase diagram of the lattice magnetisation**, concentrating on temperatures around the phase transition.

(e) **Project work in the last phase of the course:** There are many possible questions and tasks to explore and document. Details and choice of topic are intentionally left to you. You may want to start with relatively simple extensions to your existing code. Spend some time familiarising yourself with the options and their possible risks and rewards before jumping into coding. Apart from doing your own thinking, among the many places to look for inspiration are the following:

   i. **For various temperatures** including near the phase transition:
      - Visualisation of the lattice as it changes
      - histograms of relevant quantities;
      - specific heat, entropy and heat capacity (e.g. Newman 3.5)
      - spatial correlation functions (e.g. Newman pp.13–14, 3.6)

   ii. **Near the phase transition:**
      - Switch over to a logarithmic scale of $|T - T_c|$
      - More stringent tests of equilibration.
      - critical exponents (e.g. Newman 4.1 and 8.3.1)
      - cluster flipping and the Wolff algorithm (e.g. Newman 3.7.1, 4.2, 4.3)
      - finite-size scaling (e.g. Newman 8.3)

   iii. **Larger lattices** are very interesting but also expensive in terms of CPU time. Remember to again go through all the preparatory steps.

   iv. **Related topics:**
      1. Also see Schroeder's *Thermal Physics*, Section 8.2
      2. Potts models (e.g. Newman 4.5.1)
      3. Different lattice topologies, dimensions, boundary conditions.