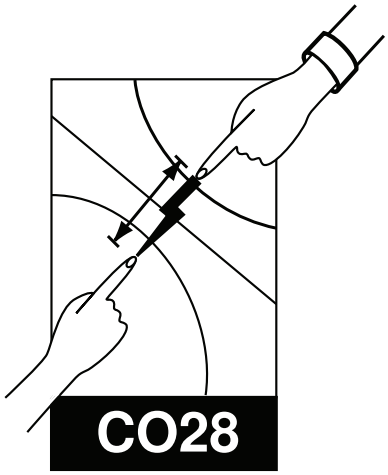
Ferromagnetism



revised EG, MK, NC October 2020

Preface

It is common practice in modern software engineering to write programs in a modular and standardised way. To help you get started, appendix A provides you with one or more specific function headers which you **MUST** use to write the functions around which your program should be built and in order to receive a satisfactory mark. Your comments in the header of each function must include:

- author and date,
- purpose (a brief description of what the function does),
- inputs and outputs,
- and, if appropriate, any constraints or limitations of use.

Do not forget that you will also need to keep good records of your progress during this practical in your logbook. If you make plots and/or write any notes electronically, ideally, you would print them and affix them into the pages of your logbook. You should have comments in function and script headers of your code, as well as comments within your code. These aspects may all be considered at marking time.

The template provided is for the default language of the Lab: MATLAB. If you have checked with a demonstrator about using another language and they have authorised you to do so, you must use the functional equivalent of the appendix A function headers in that language.

Assessment

Please see the Part A Guide in Canvas for Computing Lab deadlines as well as availability (schedule) of Computing Lab Demonstrators who mark your work as well as offer help and advice. Before you meet with a demonstrator for marking, **you must** upload your work (both your code and your report based on AD34 — *the art of scientific report writing*[1]) into WebLearn (also described in Canvas).

1 Introduction

A ferromagnet is a piece of material containing a regular array of elementary magnets or 'spins'; each spin interacts with its neighbours in such a way that the energy is reduced if the spins are parallel. The result of this interaction is that below a 'critical temperature', T_c, the material has a spontaneous magnetic moment even when there is no magnetic field present. The material is not magnetic when the temperature, T, is above T_c. This is an example of a phase change (see [2]).

The Ising model is a simplified model of a ferromagnet. Its properties, such as the magnetic moment and the transition temperature, can be studied using a method of random sampling of the configurations. The aim of this computing practical is to study some aspects of the Ising model in two dimensions.

[1] `www-teaching.physics.ox.ac.uk/practical_course/Admin/AD34.pdf`

2 Physical description

The two-dimensional Ising model consists of an array of spins at the points of a square lattice. Each spin can only point 'up' or 'down' so we give the spin at the ith site one of the two values $S_i = +1$ or -1. The energy of a configuration is taken to be

$$E = -J \sum_{i,j} S_i S_j + B \sum_i S_i \tag{1}$$

where the first sum is over pairs of *neighbouring* sites, and gives the effect of the spin-spin interaction, and the second sum is over *all* sites and represents the effect of an applied magnetic field. For ferromagnetism $J > 0$. The calculation is made for a rectangular piece of the lattice, and periodic boundary conditions are applied as described below in both directions.

According to statistical mechanics, the physical properties of a system can be found by taking a weighted average over the configurations, (each configuration is a particular arrangement of 'spin-up' and 'spin-down' sites). The weight for each configuration is proportional to the Boltzmann factor, $\exp\left(-E/kT\right)$, where E is the energy of that particular configuration and k is Boltzmann's constant.

A famous piece of work [1] found an analytical solution for the Ising model on an infinite square lattice. There is a critical value, $J/kT \approx 0.44$, above which the equilibrium state is ordered at $B = 0$. The use of a finite sample of the lattice, even with periodic boundary conditions, will blur the sharpness of any transition.

3 Numerical approach

The weighted average is found by a Monte-Carlo method that uses the Metropolis algorithm. The algorithm says that if you are in a configuration X_n, and consider moving to a configuration X_t, then you must evaluate the ratio $r = w(t)/w(n)$, the ratio of the weights of the two configurations, so

$$r = w(t)/w(n) = \exp\left[(E_n - E_t)/kT\right] \tag{2}$$

If $r > 1$ then take the step from X_n to X_t, while if $r < 1$ take the step only with a probability r. In the steady state this process produces a random walk through the configurations, visiting each one in proportion to its weight. Initially it produces a walk towards the steady state. In principal the physical properties can be found by averaging over a suitable length in the steady state.

From any given configuration X_n the transition to X_t is considered, such that X_t is just one spin-flip away from X_n. With this choice the ratio r can be calculated from a knowledge of the state of the spin to be flipped and its four nearest neighbours. The procedure is then to sweep the whole lattice, deciding whether to flip each spin in turn.

4 Computation

Write a program to do this calculation using periodic boundary conditions, where for an $N \times N$ array the neighbours of the site at (k, N), with $k < N$, are at $(k \pm 1, N)$, $(k, N-1)$ and $(k, 1)$. Try an array of about 30×30 lattice sites. Arrange to look at a display of the configuration from time to time. This can conveniently be done by writing out the array as $+$ and $-$ signs, which is much quicker than calling a graph routine.

Pick a site inside the lattice. Try to flip its spin according to the method in the previous paragraph. Do the same for the full $N \times N$ lattice; this is called a sweep. Implement this sweep process using the template given in appendix A.1.

Calculate the magnetic moment, $M = \sum_i S_i$, after each sweep, and also the cumulative magnetic moment (i.e. the sum of all magnetic moments to date). Store these results so as to be able to plot each of these quantities against sweep number and study the approach to equilibrium. Consider starting the cumulation after several sweeps so as to avoid the effect of the initial state.

Do a few runs to see how many sweeps you need to get a reasonably steady average magnetic moment — as many as 500 may be needed near the phase transition. For $J/kT > 0.44$ you may find a regime in which M changes sign from time to time without changing magnitude much, so you may want to plot $|M|$ instead.

Run your program for a range of values of J/kT that span the critical value, and plot M as a function of J/kT. Do this for $B/kT = 0$, 0.001 and 0.01.

▶ **You may opt at any point to discuss results or plots with a demonstrator before proceeding.**

5 Final considerations

Consider the following in your write-up:

Discuss how the Metropolis Hastings method works and comment on its disadvantages. Suggest another method that could be used to tackle this problem. For a detailed discussion of the Metropolis method and its variants, consult [5]. You could also discuss the effect of an applied magnetic field on the behaviour of M versus J/kT.

6 Preparing for assessment

Part A Computing Practicals are an exercise in computer programming as well as in scientific report writing. Your report must be written in the style described in AD34 — *the art of scientific report writing*[a]. To write your report, you have a few options; Microsoft Word or LaTeX typesetting are most common. Whatever your choice, the system you use must be able to produce text-readable PDF format (not PDF created from a scanned image on a printer).

When you have finished assembling your code (and have tested it completely) and your report is complete, you must upload your work (both the code and the report) electronically to WebLearn (described in the Part A Guide in Canvas). Be sure to complete these uploads well in advance of meeting with a demonstrator for marking. Deadlines and Computing Demonstrator schedules are also found in Canvas.

At marking time, be prepared with a copy of your report and your logbook (where you wrote extra notes during your program development). Be prepared to describe your code and demonstrate its execution.

[a] `www-teaching.physics.ox.ac.uk/practical_course/Admin/AD34.pdf`

A Functions to be implemented

A.1 Sweep

```
function [S_next] = sweep(S_init, JkT, BkT)
% Author: ??? , Date: ??/??/????
% Perform a full sweep of the configuration given values of J and B
% Input:
% * S_init: (N x N) matrix of the initial configuration
% * JkT, BkT: the values of J / kT and B / kT
%
% Output:
% * S_next: (N x N) matrix of the next configuration after one full sweep
%
% Example use:
% >> JkT = 0.4;
% >> BkT = 0.01;
% >> S = rand(30);
% >> for i = [1:500]
%        S = sweep(S, JkT, BkT);
%    end
% >> imagesc(S); % shows the current configuration
end
```

Bibliography

[1] L. Onsager, Crystal statistics 1. A two dimensional model with an order-disorder transition, *Phys. Rev.* 65, 117-149, 1943. doi: 10.1103/physrev.65.117

[2] H. M. Rosenberg (editor), *The Solid State*, 3rd edition, OUP, 1988.

[3] J. M. Yeomans, *Statistical Mechanics of Phase Transitions*, Clarendon Press, 1992.

[4] S. E. Koonin and D. C. Meredith, *Computational Physics*, Westview Press, 1998.

[5] D. J. C. Mackay, *Information Theory, Inference and Learning Algorithms*, CUP, 2003