Patrick Donelan (#94275)

Computational Physics
1. Project 4: Ising Model

In this project we extend the *quantum spin chain* model of Project 3 into two dimensions. To recap, a quantum spin chain is a one dimensional lattice of quantum spin sites, where each site can be either *spin up* or *spin down*. By extending the model to two dimensions we create a mesh of spin sites. This is known as the *Ising Model*. We constrain our model so that sites can only interact orthogonally with their nearest neighbours (see Figure 1).

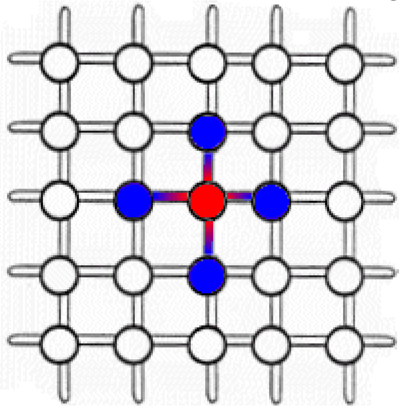


Figure 1 – 2D Mesh (based on image from www.gigaflop.demon.co.uk/comp/chapt3.htm)

In Project 3 the one dimensional lattice looped back on itself to form a chain. In two dimensions, we apply the same boundary conditions to obtain a toroidal surface (see Figure 2).

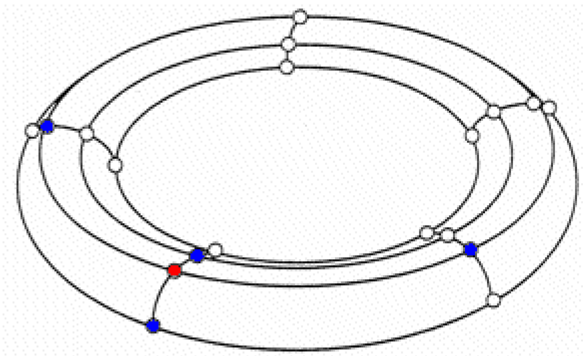


Figure 2 – Ising Model ((based on image from from www.gigaflop.demon.co.uk/comp/chapt3.htm)

The Ising Model can represent many physical systems (for example, binary alloys). In this project we choose to model a magnet. In our model each site is a microscopic magnetized regions. We can imagine these magnetized regions as being (idealised) atoms in a metal; the atoms can be either aligned (spin up) or anti-aligned (spin down).

If we have N_s spin sites in our 2D lattice, then the energy of a particular configuration is given by:

$$E\left(\vec{\sigma}^{z}\right) = -J\sum_{\langle\alpha\beta\rangle}\sigma_{\alpha}^{z}\sigma_{\beta}^{z} - B_{z}\sum_{\alpha}\sigma_{\alpha}^{z} \tag{1.1}$$

where:

$\vec{\sigma}^{z}$ represents the configuration of N_s spin sites,

σ_{α}^{z} represents the spin of site α ($\pm\frac{1}{2}$),

J is the interaction (coupling) strength,

B_z is the strength of the magnetic field in the z-direction and

$\langle\alpha\beta\rangle$ indicates that the double summation is over nearest neighbours (as indicated in Figure 1) such that each link is counted only once.

We want to model not just any sort of magnet, but a *ferromagnet*. A ferromagnet is a piece of ferromagnetic material, in which the microscopic magnetized regions have been aligned by an external magnetic field. To simulate a ferromagnet with the Ising model, we set J = 1 (so that the Energy is lowered if neighbouring spins are aligned). We do not wish to consider the effects of an external magnetic field, so we set B = 0.

Physically, the atoms in a metal will tend to align with their neighbours if this produces a lower energy configuration. Temperature plays an important role as well; at high temperatures the atoms are randomly bumped and jiggled about, increasing the likelihood of random flips. We simulate both of these phenomena in our model (using the Metropolis algorithm which will be explained shortly). An interesting feature of this system is that the it exhibits a distinct order-disorder transition at a certain (critical) temperature.

At high temperature, disorder rules. A typical system configuration is shown in Figure 3.

The top section of Figure 3 shows the various buttons used to control the program and a scrollbar which is used to control the temperature (given in terms of β which is inversely proportional to temperature). Figure 3 shows a typical configuration for β = 0.01. The graphical section shows a flattened-out view of the lattice (which as far as the algorithm is concerned wraps back on itself). Blue squares correspond to sites in the spin down state, red squares correspond to sites in the spin up state.

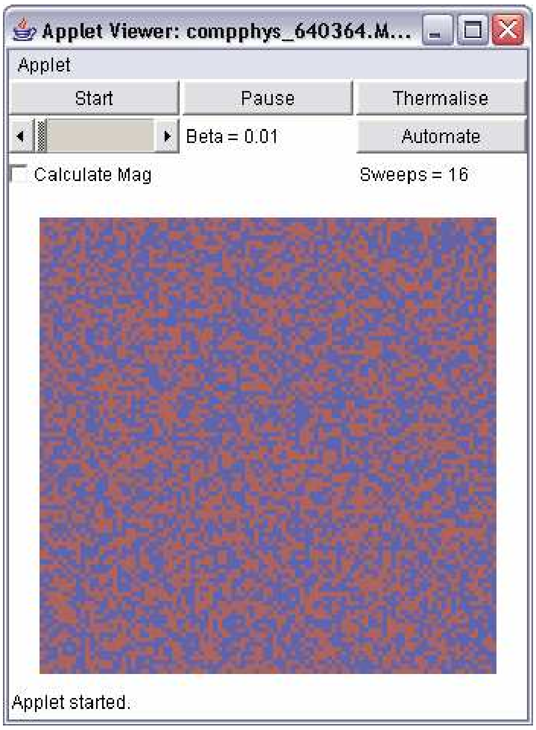


Figure 3

We can see from Figure 3 that the distribution of spin-up and spin-down sites is essentially random.

Figure 4 shows the result of increasing β to 1. At this lower temperature, the spin sites have begun aligning themselves with their neighbours into magnetic domains. It is as if a battle is being waged between the red squares and the blue squares, each side trying to gain as much territory as possible.

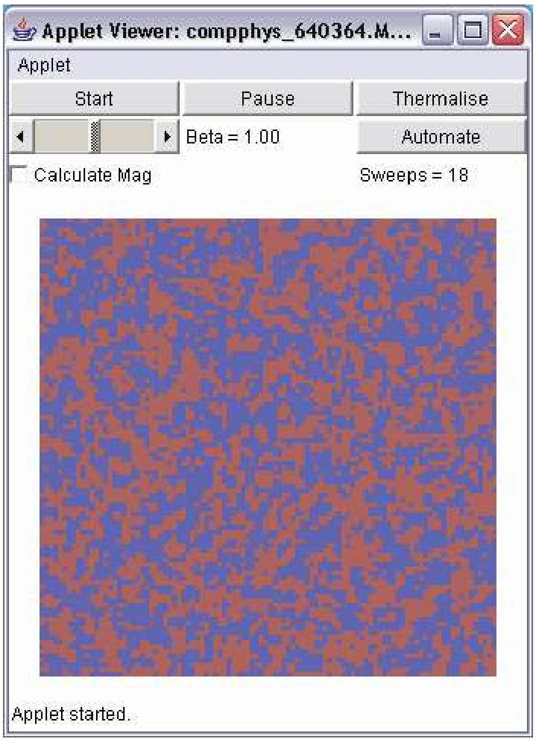


Figure 4

By letting the algorithm run for longer, we see that one spin-state begins to dominate, in this case spin-down (blue). See Figure 5.

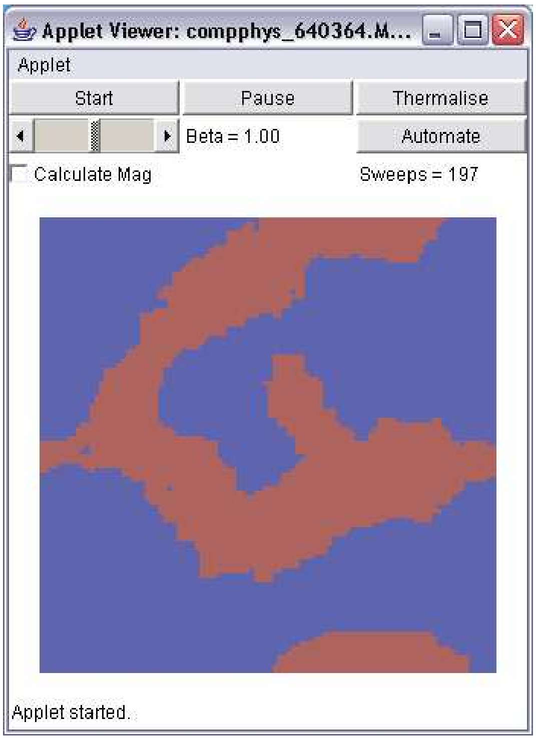


Figure 5

Eventually blue 'wins', and the ferromagnet is completely in the spin-down state. This is the most common end-result of the system where the temperature is low. It is, however, possible for a continuous band of spin-up or spin-down to be created in the toroid surface, which cannot be eliminated by the opposite state. Figure 6 shows an example of this. This is also a stable end-state, although there may be minor fluctuations along the border if it is not completely horizontal or vertical.

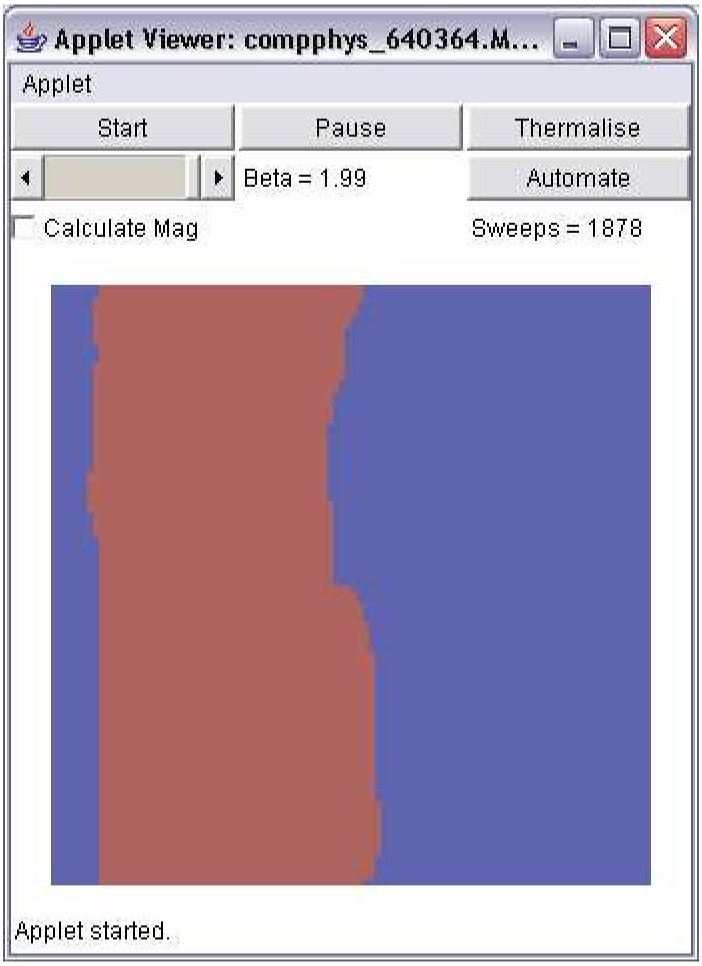


Figure 6

The critical temperature is the border between these two extremes (order and disorder). It can be shown analytically that for an infinite lattice, the critical temperature lies at $\beta = \frac{1}{2}\ln\left(1+\sqrt{2}\right) \simeq 0.4407$. We will investigate our model shortly to see how it compares to this theoretical result.

Magnetisation

In our model, we are interested in measuring the magnetisation as a function of temperature. The magnetisation of a given configuration is the just the sum over all spins:

$$M\left(\vec{\sigma}^{z}\right) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sigma_{i,j} \qquad \textbf{(1.2)}$$

From equation (1.2) we can see that the random distribution of spin-up and spin-down states in Figure 3 will correspond to a low magnetisation because the individual sites will effectively cancel each other out. On the other hand, the one-sided victory that is fast-approaching in Figure 5 will correspond to a very high magnetisation magnitude because all sites are in the same state. Because neither spin state can completely dominate in Figure 6, it corresponds to a relatively lower magnetisation magnitude, somewhere between zero and the value of a one-sided win.

However what we want to calculate is the magnetisation expectation value, that is, the ensemble average. The obvious way to do this would be to generate every possible configuration and compute the sum shown in equation (1.2). However for even modest sized lattices this is an impossible task as there are far too many configurations to be considered. So how do we compute the magnetisation? We turn to a probabilistic technique known as the Monte-Carlo procedure.

The Monte-Carlo Procedure
The basic idea behind the Monte-Carlo procedure is to find an approximate solution to an integral or summation by sampling only some of the data points at random and applying a suitably chosen weighting function. If the weighting function has similar properties to the function being evaluated then the Monte-Carlo method can accurately compute the integral or summation, and if many dimensions are involved (typically 4 or more) then it will be exponentially faster than standard non-probabilistic evaluation techniques (such as trapezoid, Simpson's, etc..).

How do we apply this technique to our Ising model to calculate the magnetisation? Firstly we need a way of generating configurations to evaluate (our random sample points). We want a weighting function that will generate configurations according to the distribution of micro-states of the canonical ensemble. Physically, the probability of a given state is given by:

$$P\left(\vec{\sigma}^z\right) = \frac{e^{-\beta E\left(\vec{\sigma}^z\right)}}{Z} \tag{1.3}$$

Where Z is the partition function of the Ising model, given by:

$$Z = \sum_{\vec{\sigma}^z} e^{-\beta E\left(\vec{\sigma}^z\right)} \tag{1.4}$$

So if we can generate configurations according to this probability distribution, the ensemble averages will be approximated by a summation over a restricted set of states which are more important according to the weight function, and the Monte-Carlo procedure will give us a good approximation of the magnetisation. But how exactly do we implement this? Luckily for us there exists a standard algorithm for this purpose, known as the Metropolis algorithm.

The Metropolis Algorithm
The Metropolis Algorithm works as follows:

- Visit a lattice site at random and try to flip the spin according to the following rules:
  - If the energy is lowered by the flip then accept it, OR
  - If $e^{-\beta E\left(\vec{\sigma}^z\right)} > \eta$ where η is a random number uniformly distributed in [0,1] then accept it,
  - Otherwise reject it.
- Perform N_s such random visits, one for each site in the lattice. Consider this to be a single *sweep*.

If we perform many such sweeps, then the configurations produced will be approximately weighted by our probability distribution (unfortunately a proof of this is outside the scope of this course), and the ensemble average of the magnetisation will be given by the Monte-Carlo average:

$$M = \langle M \rangle \approx \frac{1}{N}\sum_{k=1}^{N} M_k \tag{1.5}$$

where M_k is the magnetisation of the kth configuration.

Using this idea, we generate many such configurations (allowing for thermalisation and de-correlation – see next section), and measure the magnetisation of each using equation (1.2). Our approximation of the magnetisation is then the average of these measured values, given by:

$$M = \langle M \rangle \pm \sigma_M \tag{1.6}$$

where the statistical error is the standard deviation:

$$\sigma_M = \sqrt{\frac{1}{N}\left\langle \langle M^2 \rangle - \langle M \rangle^2 \right\rangle} \tag{1.7}$$

In this way we use the Monte-Carlo method to calculate the ensemble average, like an experimenter taking repeated measurements of the same quantity every few minutes to compute an average value. The same procedure can be used to calculate other physical quantities, although for this project we only consider magnetisation. Obviously we want to take as many measurements as possible (large sample size) to reduce the statistical error.

Hot and Cold Starts

When using the Metropolis algorithm, we need to decide what state we are going to start from. The program that I wrote can be started from two different states:

- All states aligned in either spin-up or spin-down. This ordered state is known as a *cold start* because as we saw in Figure 5 it is the state that the system most commonly evolves to when the temperature is low. With our understanding of the metropolis algorithm, we can see that at a low temperature there will be very few random flips; states will only we flipped if they lower the energy of the system and unless there is a continuous band in the toroid surface (Figure 6) this will force the system into complete alignment.
- Random alignment. This disordered state is known as a *hot start* because as we saw in Figure 3 it is the state of the when the temperature is high. With our understanding of the metropolis algorithm, we can see that at a high temperature there will be many random flips. If the temperature is high enough these random flips will drown out flips that lower the energy of the system and distribution of spin-up and spin-down states will be random.

Thermalisation

As mentioned previously, we want to use the program to calculate magnetisation as a function of temperature. When the value of β is changed, we need to give the system time to respond to the change, that is, we need to perform several sweeps so that the effects of the new temperature can propagate across the lattice. For example, Figure 7 shows the system after a sudden change of β from 0.1 to 1.5. It is clear that the low temperature is forcing the system into a total spin-up state, but not enough sweeps have been performed for this to happen fully. If we were to measure the magnetisation right now, we would obtain a magnitude less than the maximum magnitude, even though at this temperature the system will eventually converge to a totally-aligned state. This measurement would then lower the overall average (later measurements would see a totally-aligned system) and increase the statistical error, although we could limit these effects by increasing our sample size.

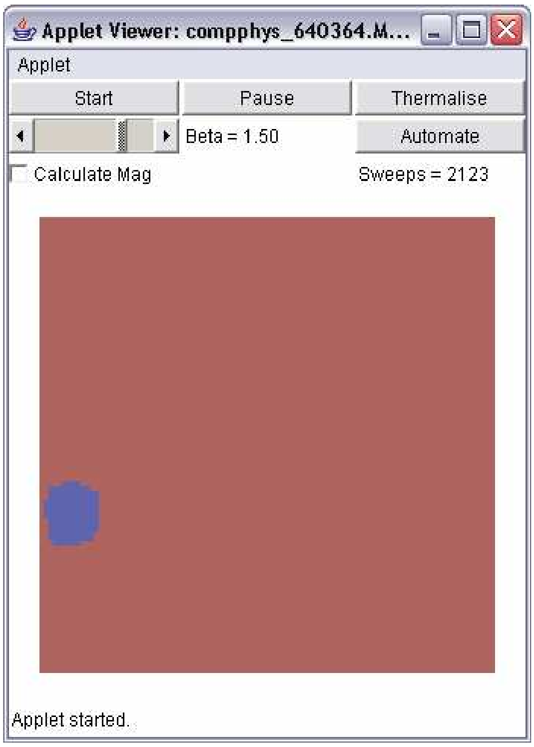


Figure 7

The value of *thermaliseSweeps* in my program is customisable. I used a value of 100 when performing measurements. Combined with a large sample size this generally gave good results as I could watch the system evolving and check visually that by the time thermalisation was over the system was in or close to its final state. However sometimes by fluke the system would evolve through a state that required a very long time to converge on its final state, such as the diagonal stripe shown in Figure 8. In these cases the first few measurements would be inaccurate, but with a large sample size they would be outweighed by the later measurements in the sample. I used a sample size of 200 for all of my systematic investigation.

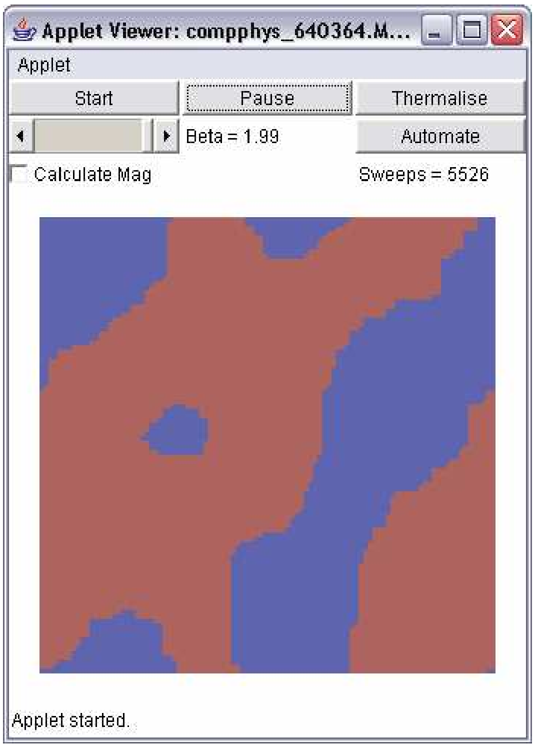


Figure 8

Ideally the program would keep thermalising until it converged on a totally-aligned state or it detected that the number of random fluctuations was high enough to keep the system in an evolving state (such as around the critical temperature). In hindsight I wish I had implemented my program like this, as it would have kept the statistical uncertainty extremely low everywhere except near the critical temperature!

De-correlation

Since we will be performing statistical analysis on our magnetisation measurements, we need them to be statistically independent (or as close to as possible). To dilute this correlation we perform de-correlation sweeps between measurements, just like the experimenter waiting for a fixed amount of time to pass between real-world measurements. For temperatures where the system is converging to a totally-aligned state, these extra sweeps will give the system extra time to converge. Close to or above the critical temperature these sweeps will allow enough random fluctuations to take place that successive measurements are approximately statistically independent. The only limitation of this technique is that if the system has evolved to a continuous-band state, no number of sweeps will allow it converge to a totally-aligned state, and successive measurements will all show a low magnetisation magnitude.

I made the decision that a continuous-band state was a by-product of my model and not truly representative of the physical system. The major limitation of my model is that it only takes into account nearest-neighbours in an orthogonal sense; diagonal neighbours are not taken into account, and hence the continuous bands that occur are either horizontal or vertical. I speculate that a model incorporating diagonal neighbours would allow these bands to be broken down diagonally by the dominant state, allowing the system to evolve to a totally-aligned state. For these reasons, I took steps to avoid continuous-band states. For example, if when measuring the magnetisation for a particular temperature the system evolved to a continuous-band state from a hot start, I would re-seed the hot start so that the system started from a different random configuration and hence evolved along a different path in configuration-space.

When systematically measuring magnetisation as a function of temperature (starting from a high temperature), I initially reset the system after each temperature change. This gave the system much more chance of evolving into a continuous-band state, which disrupted my results. I found that it was better to keep the system’s state between temperature changes, as this way as long as the first evolution path resulted in a totally-aligned state, all subsequent temperature decreases would not change this end result.

I am now of the opinion that, rather than starting from a high temperature as suggested in the notes, a better approach would have been to start from a low temperature and a cold start. This way the system would start in a totally-aligned state and only make major movements out of this state as the random fluctuations became strong enough (at and above the critical temperature). I tried this out and found that this method eliminated continuous-band states altogether. Unfortunately I had already completed the several hours worth of systematic investigation of magnetisation so I wasn't able to reap the time savings that this method provides!

The value of *autoDecorrelationSweeps* in my program is customisable. I used a value of 50 for all of my systematic investigation.

Systematic Investigation

Using my program, I performed a systematic investigation of magnetisation as a function of temperature for lattices of different sizes. For each lattice size I generated a graph of magnetisation vs. β, using the standard deviation of the magnetisation sample for each β value as the error. Figure 9 shows a flowchart the program used for this investigation.

Figure 9

Figure 10 shows the magnetisation curve for the smallest lattice size considered, 8 x 8.

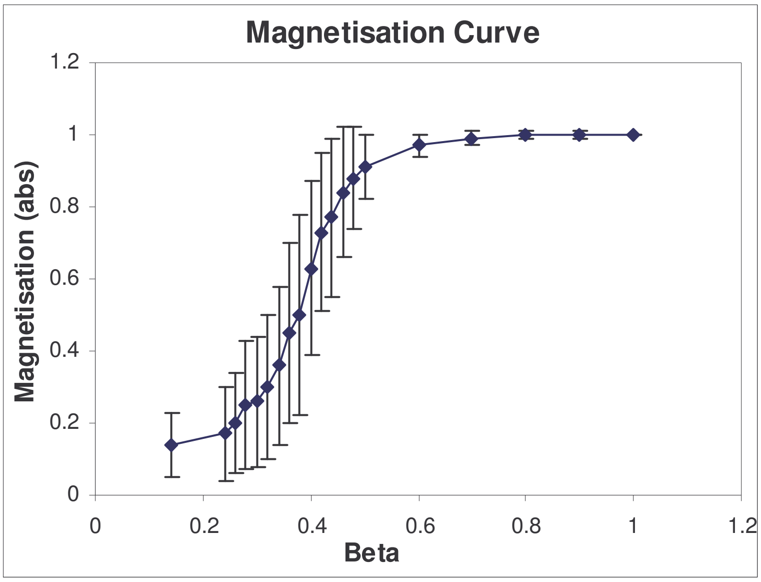
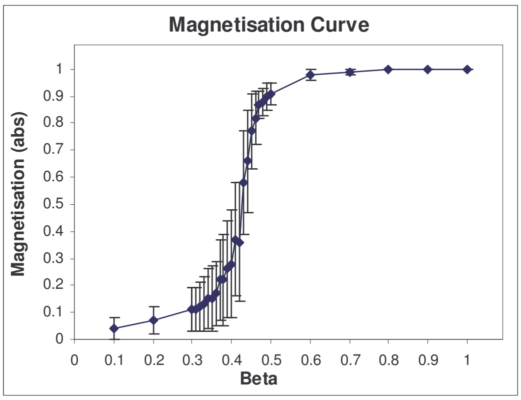


Figure 10

There are several features of Figure 10 worth pointing out. Firstly, we see that at high ß values (low temperature) the error is very small. At these points (ß > 0.6), the system has converged on a total-state solution. The small error is due to the occasional temperature-induced random flips. Contrast this with the low-temperature end of the scale, where the error is very large, of the same order as the magnetisation itself. The data points at this end of the scale are close to zero because the distribution of spin-up and spin-down states are randomly balanced, and the large error is caused by the frequent temperature-induced random flips.

We can also see a transition from low magnetisation to high magnetisation, accompanied by large error. This area, based around ß = 0.4 is the order-disorder transition that we are interested in. With such a small lattice it is hard to pinpoint the critical temperature, so I next tried running the program on larger lattices.

Figure 11, Figure 12 and Figure 13 show the magnetisation curves for increasingly large lattice sizes.



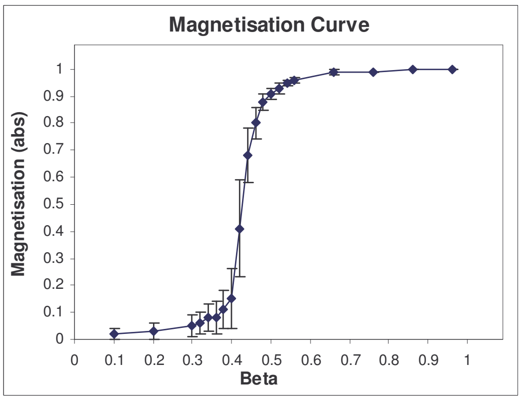


Figure 11 - 20x20 (left), 40x40 (right)

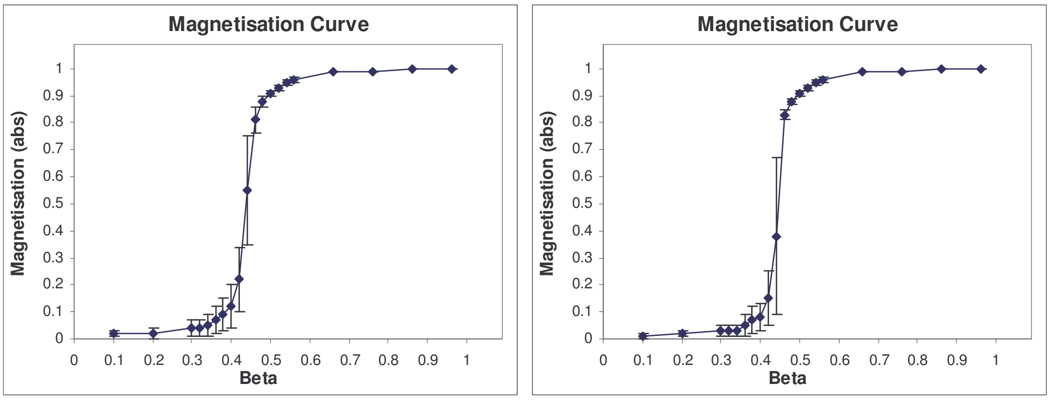


Figure 12 - 60x60 (left), 80x80 (right)

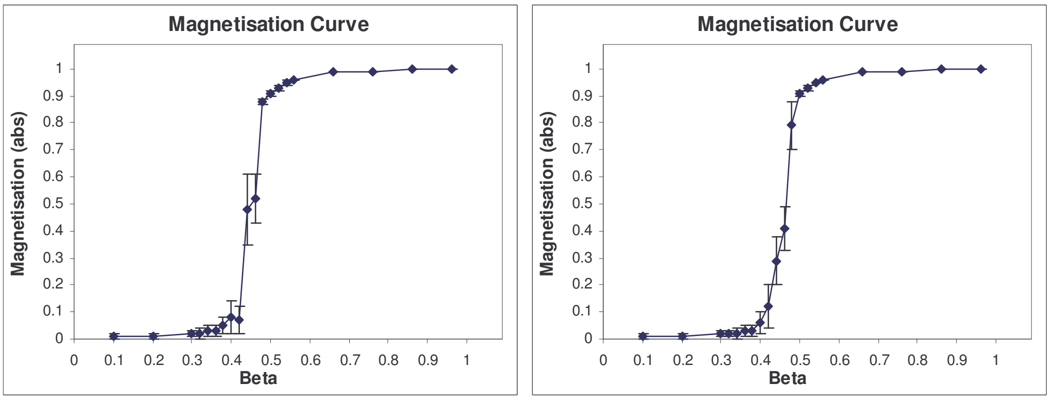


Figure 13 – 100x100 (left), 120x120 (right)

Figure 11, Figure 12 and Figure 13 show that as the lattice size grows, the transition from low to high magnetisation becomes increasingly sharper.

Another interesting thing to note is that the uncertainty, especially on high-temperature points decreases as the lattice size increases. This is because with more sites the random distribution of spins cancel out more evenly to zero (as compared to smaller lattice sizes such as 8x8).

Critical Temperature

Table 1 shows the critical temperature as a function of lattice size. This was obtained by selecting from each of the lattice samples (the data for Figure 10, Figure 11, Figure 12 and Figure 13) the value of ß that corresponded to a magnetisation of 0.5 (half-way between the low and high magnetisation extremes) within uncertainties. The error bars is then the range of ß values that correspond to a magnetisation of 0.5 within uncertainties (or the distance between adjacent ß points if only one ß value corresponds to a magnetisation of 0.5).

Table 1

Lattice Size	Beta	Error
8x8	0.4	0.1
20x20	0.43	0.05
40x40	0.44	0.02
80x80	0.44	0.02
100x100	0.44	0.02

Figure 14 shows this data graphically.

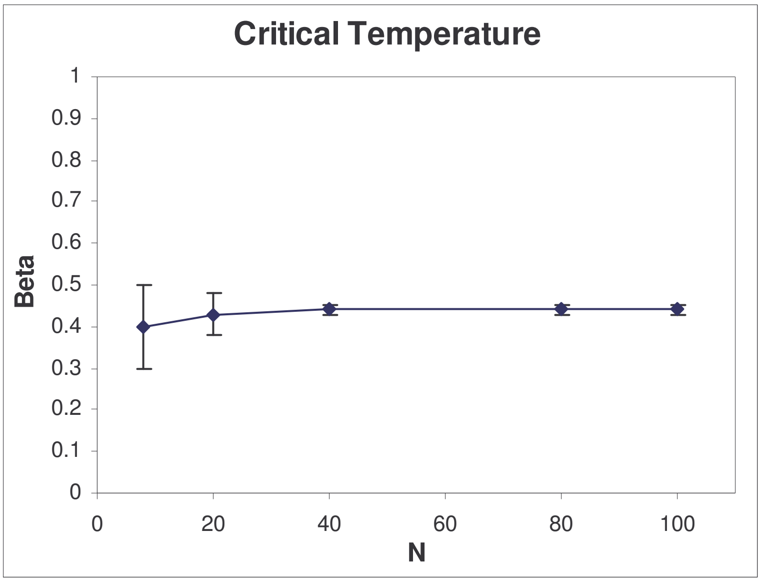


Figure 14

We can see from Figure 14 that a lattice size of 40x40 is large enough to give the critical point with tight error bounds. This computed critical temperature (ß = 0.44 $\pm$ 0.02) matches the analytic value (0.4407) very well within uncertainties, where the only limitation on our accuracy is the spacing between ß values in the algorithm. Decreasing this spacing has the side-effect of slowing the program down, although now that we have a fairly good idea where the critical value is we could run the Metropolis algorithm only over a smaller ß range with smaller steps, for example 0.44 to 0.45 in steps of 0.001. Knowing that the critical temperature would lie somewhere close to 0.4, I had already written my program so that in the ß range [0.25, 0.55] it reduced the step size from 0.1 to 0.02 (and then back to 0.1 once it left this range) to give a greater resolution close to the critical temperature but not too much of a performance hit overall.

Lattice Size

We have established that a lattice of at least 40x40 is necessary to obtain a good approximation of the critical temperature, and that the accuracy improves as we continue increasing the lattice size. This is not surprising considering the analytic value we are comparing our computed values to is for an infinite lattice, but what are the practical effects of increasing the lattice size?

The most immediate (and predictable) effect of increasing lattice size is that the algorithm slows down because more random visits are required to complete a single sweep, and there are more sites to sum over when calculating the magnetisation.

Even if we are prepared to run the program overnight to get our results, there is another practical consideration: the effects of a ß change takes more sweeps to propagate across the lattice. As a result, the number of thermalisation and de-correlation sweeps required begins to increase. This increases further the time needed to run the algorithm.

Figure 15 shows the results of running the algorithm with a lattice size of 140x140 with the same sweep settings used in the previous systematic analysis. Several erroneous points are visible (as a result of ß changes not having enough time to propagate across the lattice), especially close to the critical temperature.

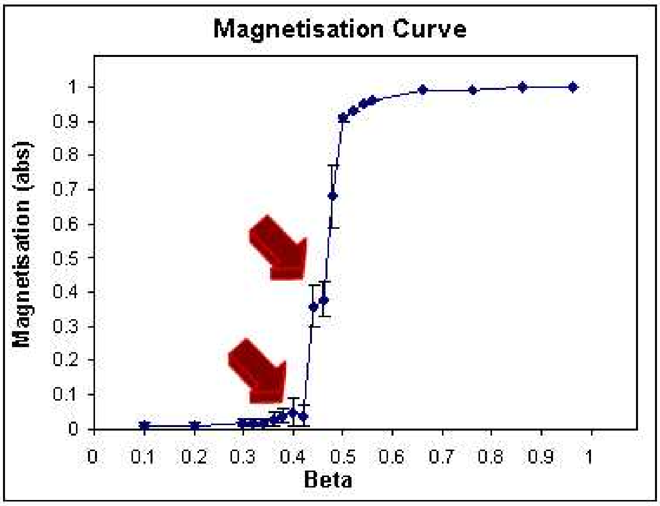


Figure 15

Statistical Errors

Several factors have been discussed that can be used to control statistical errors:

- Increase thermalisation sweeps (especially as lattice size grows)
- Increase de-correlation sweeps (especially as lattice size grows)
- Increase sample size (especially as lattice size grows)
- Increase temperature slowly from a cold start rather than decreasing temperature from a hot start to avoid continuous-band states
- Implement diagonal nearest-neighbour interactions
- Implement dynamic thermalisation detection

The pros and cons of these changes have been discussed throughout the report.