

Jon Emil Gudmundsson*
University of Uppsala

Monte Carlo method and the Ising model

Course: Statistical methods in Physics
Teacher: Gunnar Ingelman

*jeg1@hi.is

Abstract

A Monte Carlo algorithm for a two dimensional Ising model is proposed and implemented using Matlab. The results of computer simulations agree with other sources that claim that the critical value of interaction strength is close to 0.44.

Table of contents

1	Introduction	1
2	The Model	1
3	Running the model	4
3.1	A bigger lattice	6
4	Comparing the model with theory	8
	References	10
A	montecarlo.m	11
B	ising.m	13

1 Introduction

The two dimensional Ising Model is very suitable for practising simple Monte Carlo methods. It describes a lattice with a discrete number of particles that have some binary property such as spin. We can study the evolution of the system over time depending on a particular variable called the interaction strength. At a certain value of interaction strength we notice a phase transition, albeit a weak one. The interaction strength at this transition is compared to a theoretical value.

2 The Model

We can define a Hamiltonian for a system that is dependent on the arrangement of spins on a lattice and from that deduce properties such as Magnetisation and susceptibility [1]. Suppose that the Hamiltonian is

$$H = -J \sum_{\langle ij \rangle} S_i S_j - B \sum_i S_i \quad (2.1)$$

where $\langle ij \rangle$ means that we sum over the nearest-neighbour pair of spins. This means that the spin at site ij interacts with at $i(j \pm 1)$ and $(i \pm 1)j$. We will be assuming periodic boundary conditions in our model which means that every spin will interact with four other spins regardless of their position on the finite lattice. We refer to figure 1 for better understanding of the proposed system. Here J is

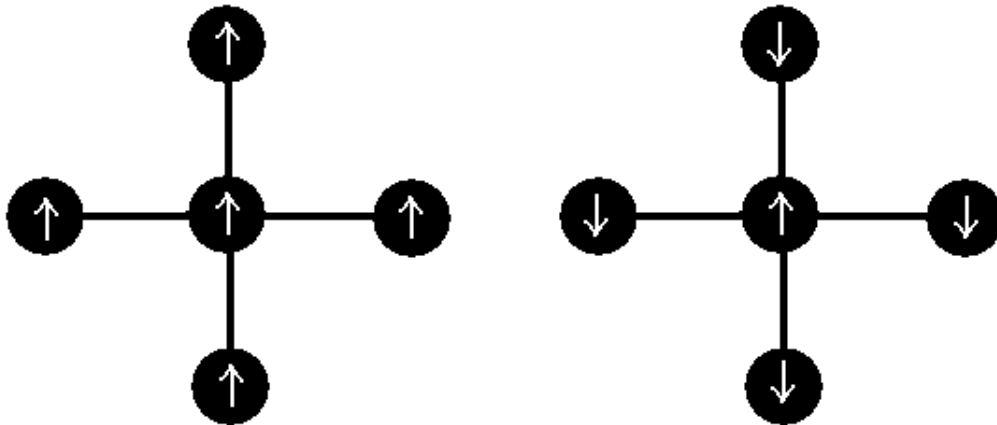


Figure 1: The energy of particle on the left is low since all the neighboring particles have the same alignment of spin. In contrast, the energy of the particle on the right is at its highest since all the neighboring particles have a different spin alignment.

the dimensionless interaction strength and B represents the energy involved in

the Magnetisation of the lattice and is also dimensionless. From the Hamiltonian we can calculate the partition function which is

$$Z = \sum_i e^{-H_i} \quad (2.2)$$

where we sum over all the particles in the lattice. Then the probability of finding the system in a certain state, denoted \mathbf{S} , is

$$p(\mathbf{S}) = \frac{e^{-H(\mathbf{S})}}{Z} \quad (2.3)$$

We can now calculate quantities such as Magnetisation and susceptibility using well known equations from statistical physics. We have

$$M = \frac{\partial \log Z}{\partial B} = \sum_{\mathbf{S}} p(\mathbf{S}) \sum_i S_i \quad (2.4)$$

for Magnetisation and

$$\chi = \frac{\partial M}{\partial B} = \sum_{\mathbf{S}} p(\mathbf{S}) \left(\sum_i S_i \right)^2 - M^2 \quad (2.5)$$

for susceptibility. The energy of the system is

$$E = \sum_{\mathbf{S}} p(\mathbf{S}) H(\mathbf{S}). \quad (2.6)$$

and the specific heat is

$$C = \sum_{\mathbf{S}} p(\mathbf{S}) H^2(\mathbf{S}) - E^2. \quad (2.7)$$

Considering only a relatively small 20 by 20 particle lattice it would be a computational feat to calculate these quantities considering that there are 2^{400} different configurations. However, we can estimate these quantities using a Monte Carlo method. We begin with a random configuration of spins on a 20 by 20 lattice. We then find how likely it is for each particle to change its spin using (2.3). The ratio of probabilities is

$$r = \frac{p(\mathbf{S}_f)}{p(\mathbf{S})} = e^{H(\mathbf{S}) - H(\mathbf{S}_f)} \quad (2.8)$$

where \mathbf{S}_f represent the configuration after the spin has been flipped. The spins of particles in which $r > 1$ or $r < 1$ but greater than a uniformly distributed random number between 0 and 1 have the potential of being flipped. To dampen the process we only flip the spin of a randomly selected subgroup which is considerably smaller than the original group. This makes for a more likely development of the system. More information about the dampening process and its control variable, *randTol*, can be found in chapter 3. Iterating until a stable value for quantities

such as M and E is obtained will give as an estimation of the true value. We must point out that in the limit of an infinitely large lattice, it is possible to solve the Ising model exactly. If we assume that $B = 0$ the expressions are simplest and we see that the energy is [2]

$$E = -N^2 J \coth(2J) \left[1 + \frac{2}{\pi} \xi K_1(\kappa) \right], \quad (2.9)$$

the Magnetisation is

$$M = \pm N^2 \frac{(1 + z^2)^{1/4} (1 - 6z^2 + z^4)^{1/8}}{(1 - z^2)^{1/2}} \quad (2.10)$$

when $J > J_c$ but vanishes when $J < J_c$ and the specific heat is

$$C = N^2 \frac{2}{\pi} (J \coth(2J))^2 \left(2K_1(\kappa) - 2E_1(\kappa) - (1 - \xi) \left[\frac{\pi}{2} + \xi K_1(\kappa) \right] \right). \quad (2.11)$$

Here

$$\kappa = 2 \frac{\sinh(2J)}{\cosh^2(2J)} \leq 1, \quad (2.12)$$

$$\xi = 2 \tanh^2(2J - 1) \quad (2.13)$$

and $z = e^{-2J}$. The critical value of J for which $\kappa = 1$ is $J_c = 0.4406868$. The complete elliptical integrals of the first and second kind are

$$K_1(\kappa) \equiv \int_0^{\pi/2} \frac{d\phi}{(1 - \kappa^2 \sin^2 \phi)^{1/2}}, \quad (2.14)$$

$$E_1(\kappa) \equiv \int_0^{\pi/2} d\phi (1 - \kappa^2 \sin^2 \phi)^{1/2}. \quad (2.15)$$

These thermodynamic parameters can be calculated using numerical methods and compared to the results of the Monte Carlo model.

3 Running the model

We wrote a code in Matlab that implements the model, a part of the code is borrowed from particularly well written Matlab code by Tobin Fricke [3]. The code can be seen in the appendix. We choose to begin by creating a square lattice with 400 particles and assign random spin orientation. We also choose a random value between 0 and 1 for the interaction strength and then watch how the system evolves over 1000 steps. Note that a positive value for interaction strength means that the energy is lower if the spin is in the same direction as that of its neighbors. This corresponds to ferromagnetism while a negative value of J means that the spins will tend to be anti-aligned corresponding to anti-ferromagnetism. The speed of evolution is controlled by the variable *randTol* and in this case we decided that *randTol* = 0.1. A value of 0.1 means that only 10% percent of the originally selected group will have its spin flipped. In essence this parameter tries to mimic the evolution of real systems. Even though a certain particle will have a smaller energy with its spin flipped it doesn't mean that all the particles in the lattice that follow that criterion will have their spins flipped immediately. We watched the evolution of 12,000 systems and took note of some of the important parameters. We then plot the total energy of the system as a function of interaction, see figure 2. Although it isn't very distinctive we can see some transition at an interaction

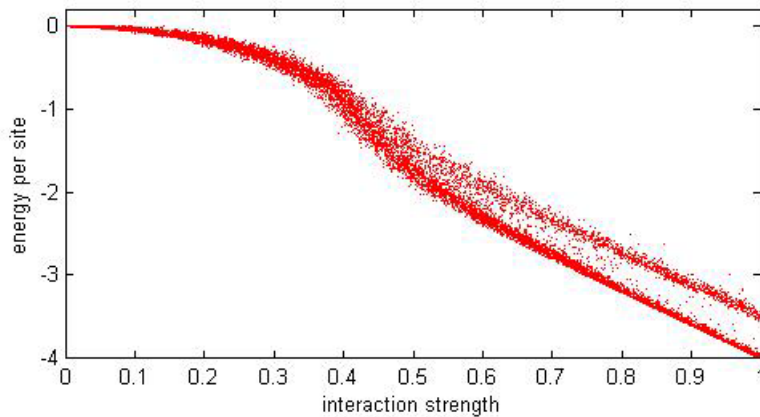


Figure 2: Energy as a function of interaction strength for the 20^2 particle lattice.

strength between 0.4 and 0.5. This can be seen more clearly when we look at the change in Magnetisation as a function of interaction strength, see figure 3. When the strength of interaction is low the spin of particles tends to be randomly oriented which results in a zero net Magnetisation. This corresponds to a high temperature (the crossing of Curie's temperature) since $J = -E/kT$ which means that J is inversely proportional to T . However, as we increase the interaction strength the spins tend to align. When looking at the relation between Magnetisation and

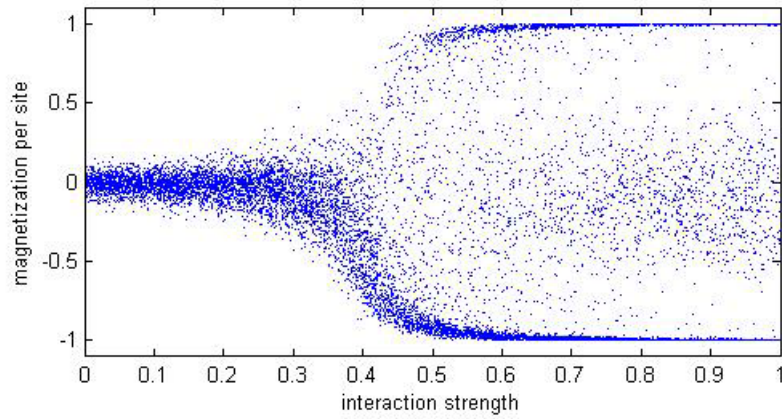


Figure 3: Magnetisation as a function of interaction strength for the 20^2 particle lattice.

energy, see figure 4 we see that there are two regions of low energy corresponding to spin up orientation and spin down. There are also low energy cases that do not show a high net Magnetisation. This happens when we have relatively large clusters of either spin up and spin down. This is analogous to the domains of ferromagnets. It is good to notice that a net Magnetisation or the formation of

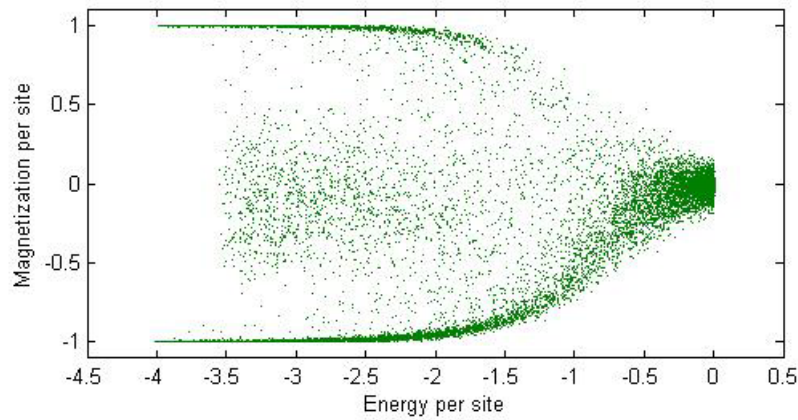


Figure 4: Magnetisation as function of energy for the 20^2 particle lattice.

domains leads to decreased energy. This is in agreement with observations of the behaviour of real materials.

3.1 A bigger lattice

We would now like to see if the phase transition at an interaction strength of around $J = 0.44$ is more apparent for lattices with higher number of particles. We therefore initialize a square lattice with 100^2 particles and watch how the system evolves over 10,000 steps. The energy versus interaction strength relation can be seen in figure (5). Note that in the case of the smaller lattice we only needed

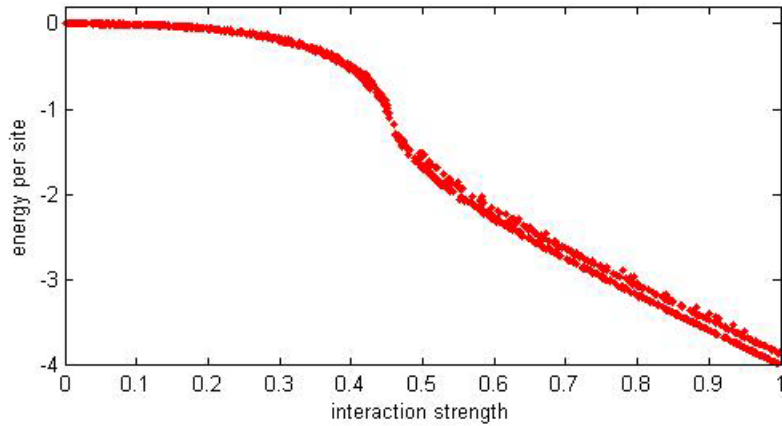


Figure 5: Energy as function of interaction strength for the 100^2 particle lattice.

to look at the evolution of the system over 1,000 steps to reach an equilibrium. We see that there are two reasons for an increased computational time in a larger lattice. One, since it is bigger the flipping of spins for more particles needs to be decided. Two, it takes greater time for a larger system to reach equilibrium which means that you have to let the system evolve over a larger number of steps. In this

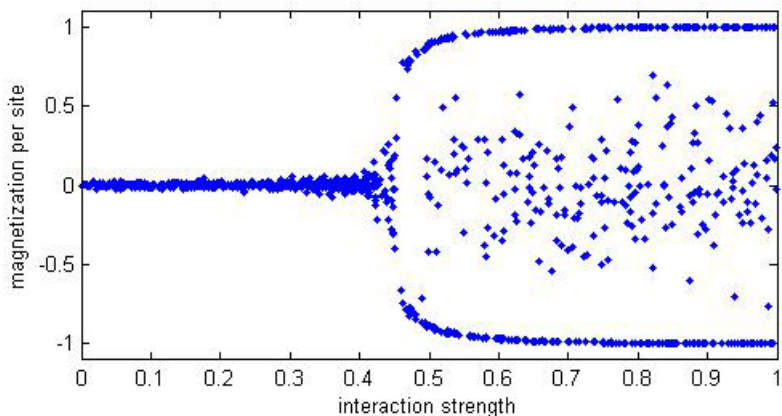


Figure 6: Magnetisation as function of interaction strength for the 100^2 particle lattice.

case, since the calculations are more demanding, we only look at the evolution of

1,000 systems. Unlike before the value of the parameter *randTol* was 0.5 which meant that the speed of evolution was much faster. The computer time needed to finish these computations was approximately 43 thousand seconds on a Pentium 4, 3GHz processor. As can be seen in figure 5 the phase transition is much more apparent. The splitting of the line which becomes more apparant as interaction strength increases is probably due to different energy for the lattice with domains, i.e. zero net magnetisation, and the lattice with a total magnetisation of -1 or 1. We also notice, see figure 6 and 7, that more systems seem to populate the

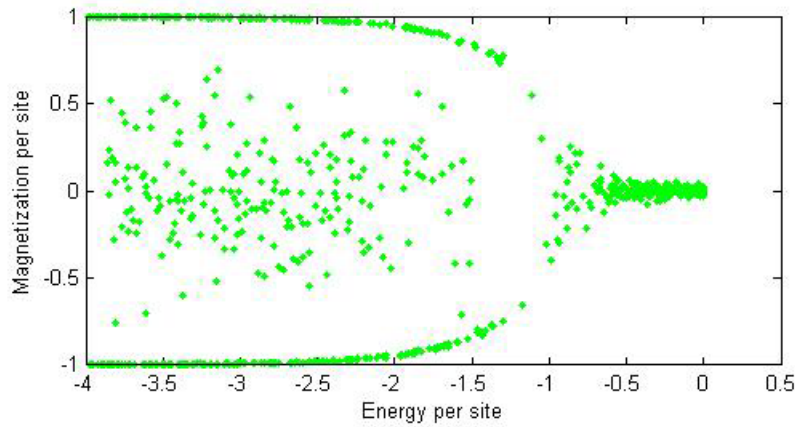


Figure 7: Magnetisation as function of Energy for the 100^2 particle lattice.

$M = -1$ and $M = 1$ branches than for the case of a 400 particle lattice. It turns out that this is in agreement with theory. We will end by comparing these results with theoretical model proposed in chapter 2.

4 Comparing the model with theory

It is easy to calculate the theoretical values for energy and Magnetisation according to equations (2.9) and (2.10). We can then compare them to the values obtained using our model. The theoretical model assumes an infinite lattice which affects the meaning of all our comparisons. It is however a reassuring thing that our Monte Carlo model gives better results for the bigger lattice. This probably implies that the greater lattice you model the more agreement you will find with the theoretical value. We can see that the theoretical model for the energy is quite different from what we obtain using our model, see figure 8. But like

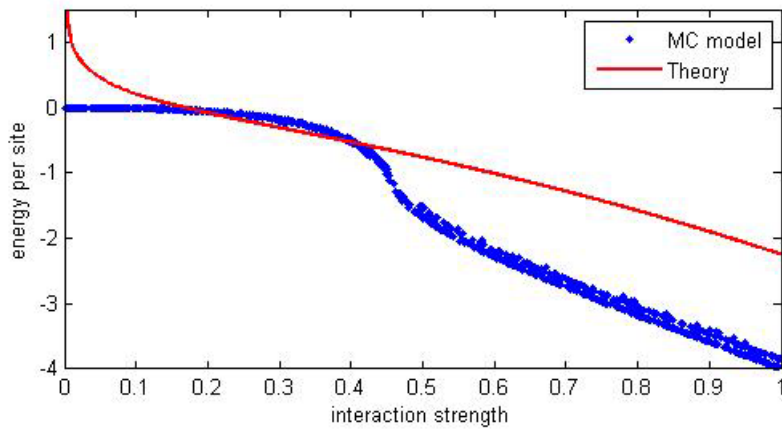


Figure 8: Energy as a function of interaction strength for the 100^2 particle lattice.

we said before, the theoretical model was in the limit of an infinite lattice and therefore we cannot be too surprised that theory and our model do not agree. We

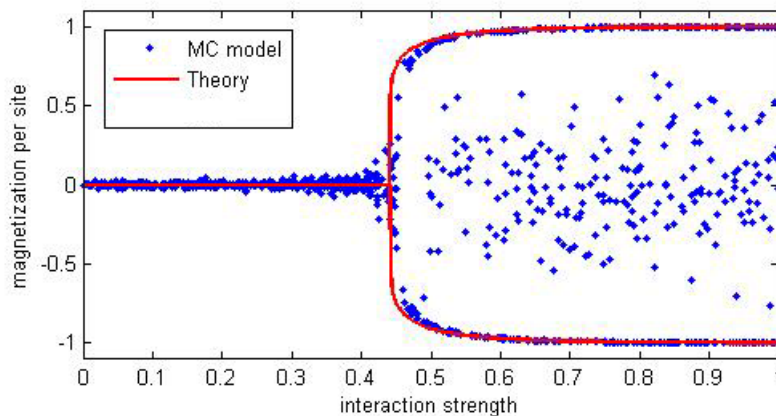


Figure 9: Magnetisations as a function of interaction strength for the 100^2 particle lattice.

would also like to point out that according to theory the energy per site goes to

infinity when the interaction strength goes to zero. This is in agreement with the

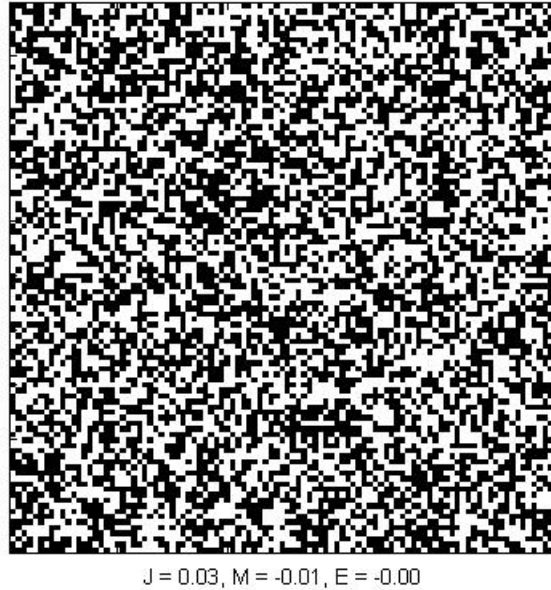


Figure 10: A random distribution of spins at low interaction strength.

fact that the interaction strength is reversly proportional to temperature and as temperature goes to infinity the energy of each particle should too. In figure 9 we see a much nicer agreement between theory and the Monte Carlo model. However, like we mentioned before, there are high interaction strength cases where the net Magnetisation is close to zero and these correspond to the formation of domains in ferromagnetic materials. We must then ask, why does theory not predict formation of domains in real material? This is a deep question. Could it be as a result of the fact that real life materials arent composed of infinite lattices but instead are finite and in many cases have impurities that cause a more caotic development of the lattice? Of course our model is very primitive and comparison with real materials isn't very meaningful. We will end by showing figures from the optional visulisation part of the Monte Carlo program. There we see how small interaction strength results in a random distribution of spins which in turn gives a net zero magnetisation and relatively high energy, see figure 10. In contrast, high interaction strength will either result in the formation of domains or an average magnetisation with an absolute value of one, see figure 11 for case of domain formation.

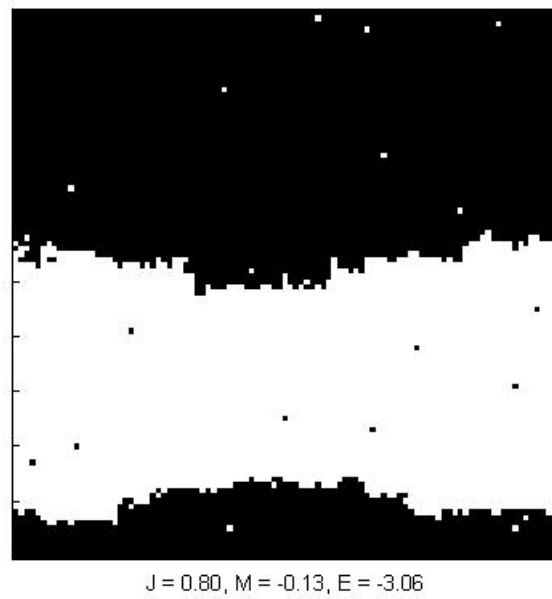


Figure 11: The formation of domains at relatively high interaction strength.

References

- [1] S. E. Koonin and D. C. Meredith, 1990. *Computational Physics*. Addison-Wesley Publishing Company, California.
- [2] K. Huang, 1963. *Statistical Mechanics*. John Wiley and Sons, New York.
- [3] Tobin Fricke. *Monte Carlo investigation of the Ising model*. Taken on the 9th of March 2008.
<http://web1.pas.rochester.edu/~tobin/notebook/2006/12/27/ising-paper.pdf>

Appendix

A montecarlo.m

Contents

- Initial Configuration
- Setting up the theoretical model
- The Monte Carlo Loop
- Figure Generation

% File montecarlo.m

Initial Configuration

```
n_grid = 20; % Size of grid
Ms = [];
Js = [];
Ns = [];
Es = [];
```

Setting up the theoretical model

```
J = linspace(0,1,1000);
kappa = 2.*sinh(2*J)./(cosh(2*J)).^2;
kappam = 2*(tanh(2*J)).^2-1;
z = exp(-2*J);
% Calculating Energy & Magnetisation as a function of interaction strength
E = -J.*(coth(2*J)).*(1+(2/pi)*kappam.*K1(kappam));
M1 = ((1+z.^2).^(1/4).*(1-6*z.^2+z.^4).^(1/8))./(1-z.^2).^(1/2);
M2 = -((1+z.^2).^(1/4).*(1-6*z.^2+z.^4).^(1/8))./(1-z.^2).^(1/2);
M1(1,1:441) = 0; %When J < J_c, M = 0
M1(1,1:441) = 0; %When J < J_c, M = 0
```

The Monte Carlo Loop

```
for i = 1:12000
```

```
%while (1),
    % Choose a random value between 0 and 1 for the interaction strength
    J = rand()+1e-10;

    % Perform a simulation
    [M, N, E] = ising2(n_grid, J);

    % Records the results
    Ms = [Ms M/(n_grid^2)];
    Es = [Es E/(n_grid^2)];
    Ns = [Ns N];
    Js = [Js J];
    i = i+1;
end

Error using ==> evalin
Undefined command/function 'ising2'.
```

Figure Generation

Energy per site, versus interaction strength

```
figure(1)
plot(Js, Es, 'ro');
hold on
plot(J,E);
ylabel('energy per site');
xlabel('interaction strength');
pbaspect([2 1 1]);
print(gcf, '-depsc2', 'ising-energy');

% Magnetisation per site, versus interaction strength
figure(2)
plot(Js, Ms, 'bo');
hold on
plot(J,M1)
hold on
plot(J,M2)
ylabel('Magnetisation per site');
xlabel('interaction strength');
```

```
ylim([-1.1 1.1]);
pbaspect([2 1 1]);
print(gcf, '-depsc2', 'ising-Magnetisation');

% Magnetisation per site, versus Energy per site
figure(3)
plot(Es, Ms, 'o', 'Color', [0 0.5 0]);
xlabel('Energy per site');
ylabel('Magnetisation per site');
pbaspect([2 1 1]);
print(gcf, '-depsc2', 'ising-mvse');
```

B ising.m

```
function [M, num, E] = ising(N,J)

B = 0;
M = []; % The total magnetic field of the system
E = []; % The total energy of the system

randTol = 0.1; % The tolerance, dampens the spin flip process

% First we generate a random initial configuration
spin = (-1).^(round(rand(N)));

% Then we let the system evolve for a fixed number of steps
for i=1:1000,
    % Calculating the total spin of neighbouring cells
    neighbours = circshift(spin, [ 0 1]) + ...
        circshift(spin, [ 0 -1]) + ...
        circshift(spin, [ 1 0]) + ...
        circshift(spin, [-1 0]);

    % Calculate the change in energy of flipping a spin
    DeltaE = 2 * (J*(spin .* neighbours) + B*spin);

    % Calculate the transition probabilities
    p_trans = exp(-DeltaE);
```



```
% Decide which transitions will occur
transitions = (rand(N) < p_trans ).*(rand(N) < randTol) * -2 + 1;

% Perform the transitions
spin = spin .* transitions;

% Sum up our variables of interest
M = sum(sum(spin));
E = -sum(sum(DeltaE))/2; % Divide by two because of double counting

% Display the current state of the system (optional)
image((spin+1)*128);
xlabel(sprintf('J = %0.2f, M = %0.2f, E = %0.2f', J, M/N^2, E/N^2));
set(gca, 'YTickLabel', [], 'XTickLabel', []);
axis square; colormap bone; drawnow;

end

% Count the number of clusters of 'spin up' states
[L, num] = bwlabel(spin == 1, 4);
```