PHYS3020 Computational Project Report

Ryan White
s4499039

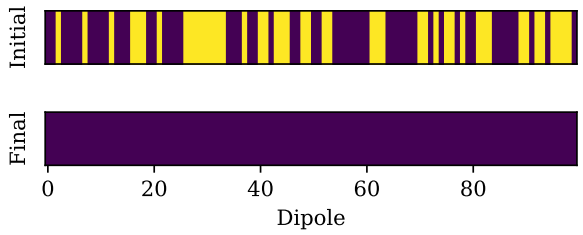
Semester 2, 2022

The programming language Python was used for this project, with the packages `numpy` and `matplotlib` used to perform analysis and generate plots.

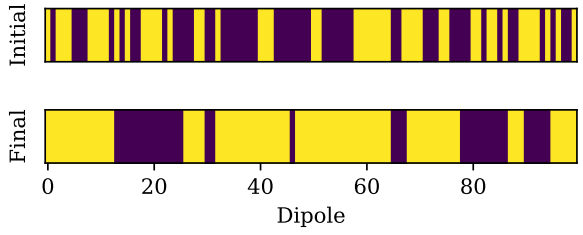
1 Question 1

1.1 1D Ising Model

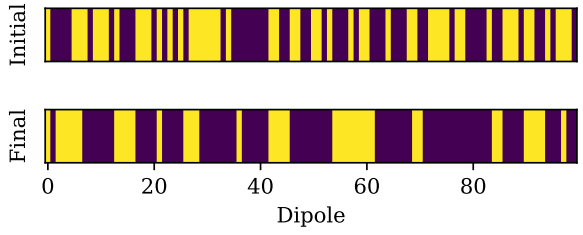
Beginning from an initial (approximately) random state with periodic boundaries, the system was subject to 10^5 time steps, so that each one of the 100 dipoles could have, in principle, 10^3 chances to flip their alignment. The results are shown, for 4 different temperatures, in Figure 1.



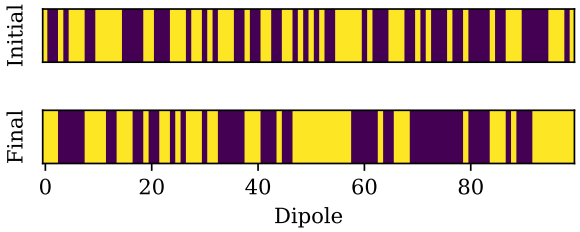
(a) Initial and final states with $T = 0.5\epsilon/k$.



(b) Initial and final states with $T = 1\epsilon/k$.



(c) Initial and final states with $T = 2\epsilon/k$.



(d) Initial and final states with $T = 5\epsilon/k$.

Figure. 1 The initial and final states of a 1D Ising Model with $N = 100$ dipoles, joined on the horizontal axis. Yellow bars represent a spin-up state, while purple bars represent a spin-down state.

As the temperature gets lower, it appears that the size of spin-parallel chunks increases. We'd expect that as $T \to 0$, all of the spins would be aligned in parallel, while as $T \to \infty$, each dipole would be aligned antiparallel to its neighbours.

The Python code that was used to generate both the initial state (given by the function `gen_init_Ising()`) and the metropolis algorithm (in 1D given by `metropolis()`) is shown in the following code block.

```
def gen_init_Ising(N):
    initState = np.random.uniform(0, 1, N)
    s = [1 if initState[i] >= 0.5 else -1 for i
    in range(N)]
    return s

def metropolis(states, iters, temperature):
    beta = 1 / temperature
    RVs = np.random.uniform(0, len(states) - 1,
    iters)
    RVs = [int(x) for x in RVs]
    for i in range(iters):
        if RVs[i] == len(states) - 1:
            Ldipole = states[RVs[i] - 1]
            Rdipole = states[0]
        elif RVs[i] == 0:
            Ldipole = states[len(states) - 1]
            Rdipole = states[1]
        else:
            Ldipole = states[RVs[i] - 1]
            Rdipole = states[RVs[i] + 1]
        energy = lambda D : -1 * ((Ldipole * D)
    + (Rdipole + D))
        dipole = states[RVs[i]]
        initEnergy = energy(dipole)
```

```
        flipEnergy = energy(-1 * dipole)
        deltaEnergy = flipEnergy - initEnergy
        if deltaEnergy <= 0:
            dipole *= -1
        else:
            flipProb = np.exp(-beta *
    deltaEnergy)
            evaluation = np.random.uniform(0, 1)
            if evaluation <= flipProb:
                dipole *= -1
        states[RVs[i]] = dipole
    return states
```

1.2 Ising Thermodynamic Parameters

We're given that the partition function of the system of N dipoles is

$$Z = (2\cosh(\beta\epsilon))^N \tag{1}$$

The average internal energy of the system is then

$$\begin{aligned} U = \langle E \rangle &= -\frac{\partial}{\partial\beta}\ln Z \\ &= -\frac{\partial}{\partial\beta} N \ln\left(2\cosh(\beta\epsilon)\right) \\ &= -\frac{\partial}{\partial\beta} N \ln\left(e^{\beta\epsilon} + e^{-\beta\epsilon}\right) \\ &= -N \cdot \frac{1}{e^{\beta\epsilon} + e^{-\beta\epsilon}} \cdot \left(\epsilon e^{\beta\epsilon} + -\epsilon e^{-\beta\epsilon}\right) \\ &= -N\epsilon \left(\frac{e^{\beta\epsilon} - e^{-\beta\epsilon}}{e^{\beta\epsilon} + e^{-\beta\epsilon}}\right) \\ &= -N\epsilon\tanh(\beta\epsilon) \end{aligned}$$

And since the internal energy per dipole is $u = U/N$, we get that

$$u = -\epsilon\tanh(\beta\epsilon) \tag{2}$$

Next, the free energy if given by $F = -kT\ln Z$, and so

$$\begin{aligned} F &= -kT\ln Z \\ &= -kT \cdot N\ln(2\cosh(\beta\epsilon)) \\ &= -kTN\ln(e^{\beta\epsilon} + e^{-\beta\epsilon}) \\ &= -kTN\ln\left(\frac{1 + e^{-2\beta\epsilon}}{e^{-\beta\epsilon}}\right) \\ &= -kTN\left[\ln(1 + e^{-2\beta\epsilon}) - \ln(e^{-\beta\epsilon})\right] \\ &= -kTN\left[\ln(1 + e^{-2\beta\epsilon}) + \beta\epsilon\right] \\ &= N\left(-\epsilon - kT\ln(1 + e^{-2\beta\epsilon})\right) \end{aligned}$$

And since the free energy per dipole is given by $f = F/N$,

$$f = -\epsilon - kT\ln(1 + e^{-2\beta\epsilon}) \tag{3}$$

The entropy can be obtained by $S = -\partial F/\partial T$, and so the entropy per dipole can be obtained by

$$\begin{aligned} \mathcal{S} = S/N &= -\frac{\partial f}{\partial T} \\ &= -\frac{\partial}{\partial T}\left(-\epsilon - kT\ln(1 + e^{-2\beta\epsilon})\right) \\ &= k\ln(1 + e^{-2\epsilon/kT}) + \left(\frac{kT}{1 + e^{-2\epsilon/kT}}\right) \\ &\quad \cdot -\frac{2\epsilon}{kT^2}e^{-2\epsilon/kT} \cdot -1 \\ &= \frac{2\epsilon}{T}\frac{e^{-\beta\epsilon}}{e^{\beta\epsilon} + e^{-\beta\epsilon}} + k\ln(1 + e^{-2\beta\epsilon}) \\ &= \frac{2\epsilon}{T}\frac{e^{-\beta\epsilon} + e^{\beta\epsilon} - e^{\beta\epsilon}}{e^{\beta\epsilon} + e^{-\beta\epsilon}} + k\ln(1 + e^{-2\beta\epsilon}) \\ &= \frac{2\epsilon}{T}\left(\frac{e^{-\beta\epsilon}}{e^{\beta\epsilon} + e^{-\beta\epsilon}} - \tanh(\beta\epsilon)\right) + k\ln(1 + e^{-2\beta\epsilon}) \end{aligned}$$

But $e^{\beta\epsilon}/(e^{\beta\epsilon} + e^{-\beta\epsilon}) = \frac{1}{2}(\tanh(\beta\epsilon) + 1)$, so

$$\begin{aligned} \mathcal{S} &= \frac{2\epsilon}{T}\left(\frac{1}{2}\left[\tanh(\beta\epsilon) + 1\right] - \tanh(\beta\epsilon)\right) + k\ln(1 + e^{-2\beta\epsilon}) \\ &= \frac{\epsilon}{T}\left(1 - \tanh(\beta\epsilon)\right) + k\ln(1 + e^{-2\beta\epsilon}) \end{aligned} \tag{4}$$

Finally, the specific heat capacity is given by

$$\begin{aligned} c = \frac{C_V}{N} &= \frac{1}{N}\frac{\partial U}{\partial T} = \frac{\partial u}{\partial T} \\ &= \frac{\partial}{\partial T}(-\epsilon\tanh(\beta\epsilon)) \\ &= -\epsilon\left(\frac{1}{\cosh^2(\epsilon/kT)} \cdot -\frac{\epsilon}{kT^2}\right) \\ &= \frac{\epsilon^2\beta}{T\cosh^2(\beta\epsilon)} \end{aligned} \tag{5}$$

1.3 Plots of Thermodynamic Properties

For a range of different values of temperature, the internal energy, free energy, entropy, heat capacity and reduced magnetisation (all with respect to each dipole) was calculated at an approximate equilibrium value.
Beginning with internal energy per dipole, the experimental value for each system state was calculated by

$$u = \frac{1}{N}\left(-\epsilon\sum_{i,j} s_i s_j\right) \tag{6}$$

These values, contrasted by the theoretical value shown in equation (2), are shown in Figure 2

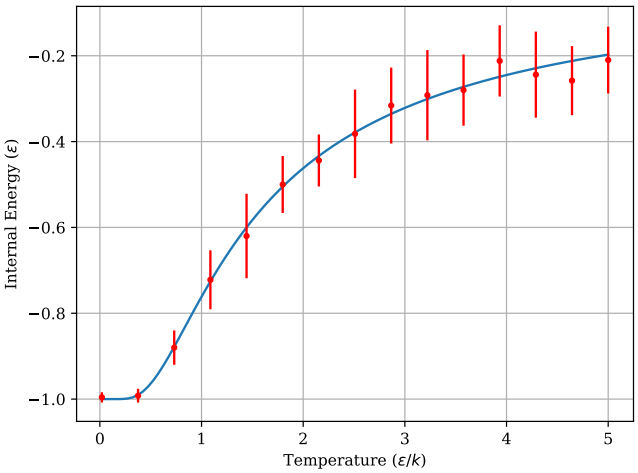


Figure. 2 Average internal energy per dipole for 15 temperatures, compared with theoretical value. The theoretical relationship is shown by the blue curve, while the data points are given in red. The data points are the average across 20 trials, with the errorbars representing the standard deviation of the values across those trials.

Since all possible combinations for a system of 100 dipoles is a large number, I invoked Sterling's approximation to calculate the entropy of any state. The total number of dipoles N was used with the number of observed spin-up dipoles $N_\uparrow$ to calculate entropy by

$$\mathcal{S} \approx \frac{1}{N}\left(N\log N - N_\uparrow \log N_\uparrow - (N-N_\uparrow)\log(N-N_\uparrow)\right) \tag{7}$$

With this as the basis of the experimental approximations for entropy, the calculated values are contrasted against the theoretical relationnship (equation (4)) in Figure 3

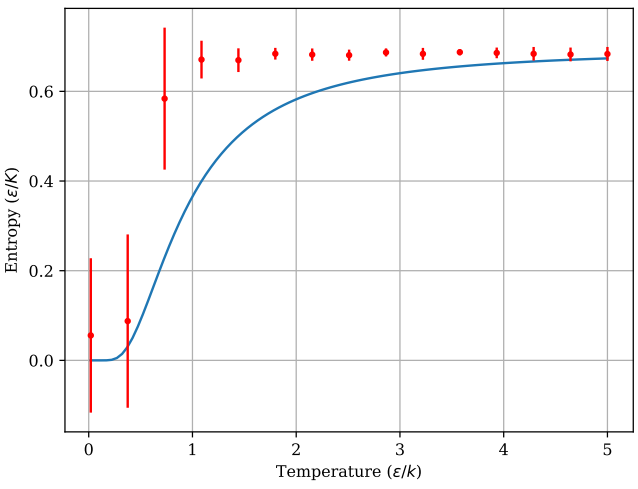


Figure. 3 Average energy per dipole across 15 temperatures. As before, the blue curve is the theoretical relationship, and the red points the data averaged across 20 trials.

With the internal energy and entropy known per dipole, the free energy could be computed by

$$f = u + T\mathcal{S} \tag{8}$$

The computed values together with the theoretical model (equation 3) are shown in Figure 4.

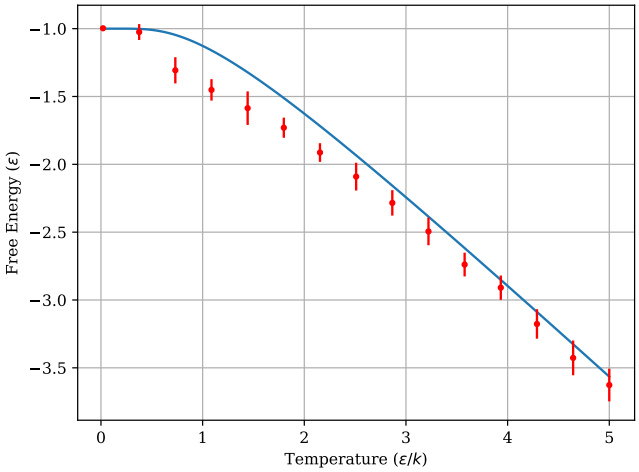


Figure. 4 Average free energy per dipole across 15 temperature values and 20 trials.

The specific heat capacity of the model can be computed by

$$c = \frac{\langle U^2\rangle - \langle U\rangle^2}{kT^2N} \tag{9}$$

Once again, the computed values and the theoretical model (equation (5))are contrasted in Figure 5.

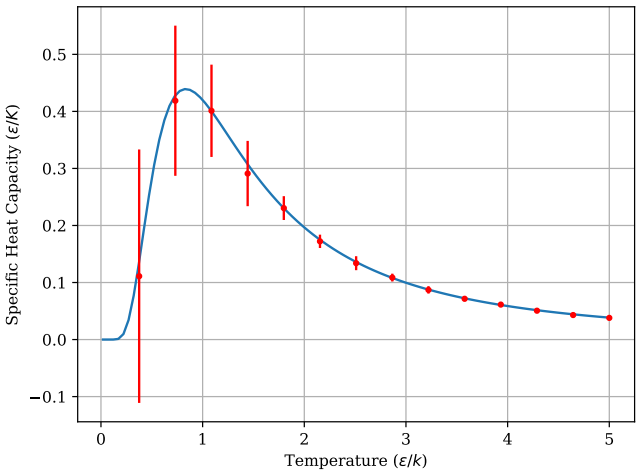


Figure. 5 Specific heat capacity for a system at 15 temperatures, averaged across 20 trials.

Finally, the reduced magnetisation per dipole is given by the average spin value across the lattice. That is,

$$m = \overline{s} \tag{10}$$

No theoretical relation exists for this in terms of temperature, only in terms of the limit of temperature. The experimental observations of the net magnetisation per dipole is shown in Figure 6.

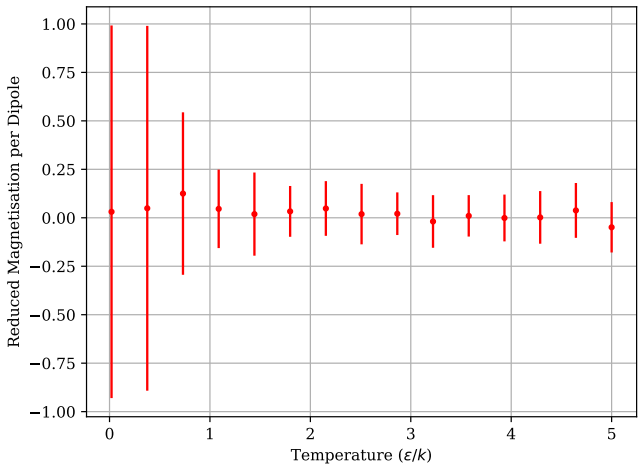


Figure. 6 Reduced magnetisation per dipole for 15 temperatures, averaged across 20 independent trials.

1.4 Plot Discussion

It's clear from looking at Figure 1 that increasingly large chunks of like-spin states emerge at lower temperatures for the 1D Ising model with periodic boundaries. We see that at sufficiently low temperature, the entire lattice transitions to the same spin state, while at high temperatures, the 'chunks' of spin states tend to size one.

Using the equation for internal energy per dipole to calculate u for some system state, the data aligns perfectly to the theoretical model (within uncertainty) as shown in Figure 2. This shows that, as the temperature increases, the internal energy of the system goes to 0 due to the dipoles being aligned antiparallel with respect to their neighbours.

Calculating the entropy of the system via the Sterling approximation for a system of size $N = 100$ was likely too much of an approximation given the disagreement in the model and data in Figure 3. We see that for low values of temperature (mostly parallel dipole alignment), the entropy is within agreement with the model, but quickly shoots up faster than the model predicts. We believe this is due to the entropy calculation method not taking into account the spin alignment of neighbouring dipoles, and only counting the total number of spin-up dipoles with respect to the total number of dipoles. As such, large chunks of parallel-aligned dipoles would contribute just as much to the entropy as would a totally antiparallel-aligned system which would, in principle, be much more likely to occur. In the temperature limit to infinity, though, the data aligns well with the model to little uncertainty.

Given that the free energy per dipole is calculated according to the internal energy and the entropy, we would expect some deviation from the model since the entropy deviates from the model. That is exactly what we see in Figure 4, where the data is slightly more negative than the model on account of the entropy being too large with respect to its own model. As the free energy has the entropy as a negative term, this yields the lower value which is seen. Apart from this, the overall trend of the data aligns well with the model.

The data for heat capacity was incredibly sensitive to the equilibrium of the system, and we had to leave the system for many more spin-flip opportunities than the other thermodynamic property examinations to get data that aligned to the model. We observed deviation from the model only at low temperatures (which is evident by the significantly larger error bars in low temperature data points), and excellent, consistent agreement with the model at larger temperatures as seen in Figure 5. Curiously, we observe that the system is least willing to change its temperature when at $T \sim 1\epsilon/k$.

Finally, we observe the average magnetisation per dipole in Figure 6, where no exact solution exists. Clearly, lower temperatures correlate with larger standard deviations for the average magnetisation of a 1D Ising model. We postulate that this is due to low temperature systems aligning almost totally parallel in *either* spin-up or spin-down states (hence the large scatter), and high temperature systems being consistently aligned antiparallel (and so have a rough average of 0 with minimal scatter). This is supported by the instantaneous system states shown in Figure 1.

1.5 Net Magnetisation at Different Temperatures

Figure 7 shows the average magnetisation of a system for 3 different temperatures, and two different system sizes across 100 trials. We see immediately that for a lower number of dipoles, the system becomes more magnetised on average (by inferring the absolute value of the magnetisation, $|m|$) for all temperatures.
Even though the average magnetisation for each temperature (across all trials) is about 0, the standard deviation is vastly bigger for lower temperatures and number of dipoles. As such, we'd expect that as $N \to \infty$ that both $\overline{m} = 0$ and s.d.$(m) \to 0$, essentially saying that an infinite 1D Ising model should have a time-average magnetisation of 0.

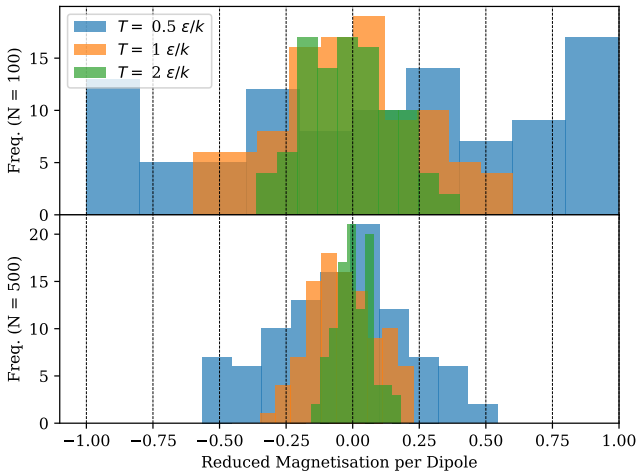


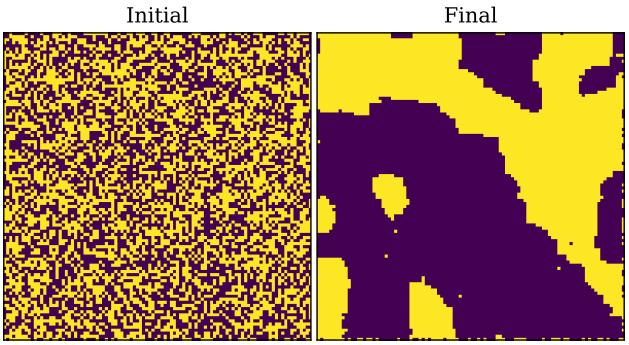
Figure. 7 Final state reduced magnetisation per dipole for different temperatures and number of dipoles, across 100 trials. The top histogram represents trials with 100 dipoles, and the bottom 500 dipoles.

Because of this, we'd expect that an infinite 1D Ising model will have no total phase transition as there will always be some infinite number of dipoles aligned anti-parallel (and so the net magnetisation will always be 0). For sufficiently low temperature *finite* systems, we already see some total phase transitions in the upper half of Figure 7 (with $T = 0.5\epsilon/k$ and $N = 100$).

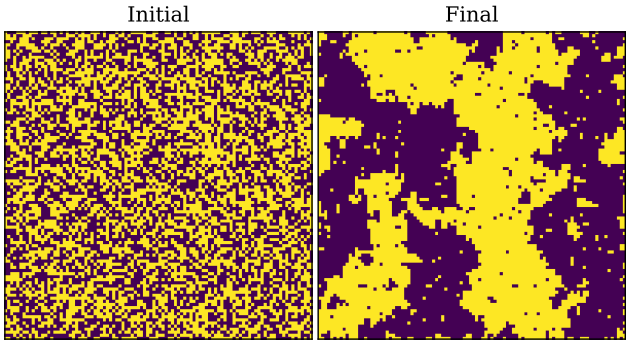
2 Question 2

2.1 2D Ising Model

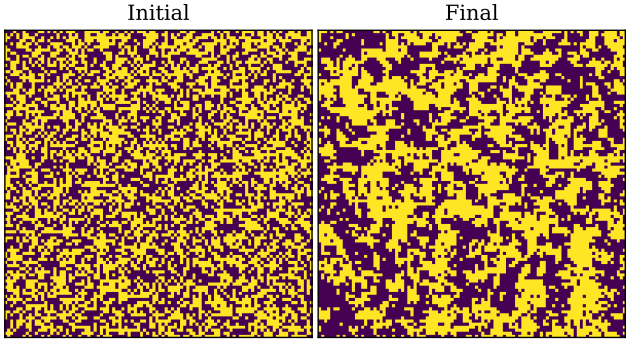
As in Question 1, a 2D lattice was initialised in a random state and left for 10^6 states to settle into an equilibrium position. The results for this across 3 different temperatures are shown in Figure 8.



(a) Initial and final states with $T = 1\epsilon/k$.



(b) Initial and final states with $T = 2\epsilon/k$.



(c) Initial and final states with $T = 3\epsilon/k$.

Figure. 8 The initial and final states of a 2D Ising Model with $N = 100^2$ dipoles in a square lattice. Yellow bars represent a spin-up state, while purple bars represent a spin-down state.

As time progressed (i.e. as the state number increased) for the lower temperatures, regions of parallel-aligned spins arose and grew. Some of this behaviour was seen in the $T = 3\epsilon/k$ state, but not to the same extent as the other two temperatures.

The code that generates these lattices is shown in the following code block.

```
def gen_init_Ising2d(N):
    s = np.zeros((N, N))
    for i in range(N):
        initState = np.random.uniform(0, 1, N)
        s[i, :] = [1 if initState[i] >= 0.5 else
     -1 for i in range(N)]
    return s

def metropolis2d(states, iters, temperature):
    '''
    Parameters
    ----------
    states : np.array
        and NxN array composed only of -1 and 1,
     corresponding to the spin of each dipole
    iters : int
        The number of iterations (different
    states) to simulate
    temperature : float
        The temperature (in units of epsilon / k
    ) of the reservoir
```

```
    '''
    beta = 1 / temperature
    RVhoriz = np.random.uniform(0, states.shape[0] - 1, iters) # generate some random positions along horizontal axis
    RVhoriz = [int(x) for x in RVhoriz] # make them correspond to an index
    RVvert = np.random.uniform(0, states.shape[1] - 1, iters) # as above but along vertical axis
    RVvert = [int(x) for x in RVvert]
    for i in range(iters):
        if RVhoriz[i] == states.shape[0] - 1: # if the index is on the right border
            left = RVhoriz[i] - 1
            right = 0
        elif RVhoriz[i] == 0: # if the index is on the left border
            left = states.shape[0] - 1
            right = 1
        else:
            left = RVhoriz[i] - 1
            right = RVhoriz[i] + 1
        if RVvert[i] == states.shape[1] - 1: # if the index is on the bottom border
            up = RVvert[i] - 1
            down = 0
        elif RVvert[i] == 0: # if the index is on the top border
            up = states.shape[1] - 1
            down = RVvert[i] + 1
        else:
            up = RVvert[i] - 1
            down = RVvert[i] + 1
        dipole = states[RVhoriz[i], RVvert[i]] # this is the value of the randomly chosen dipole
        Ldipole = states[left, RVvert[i]]   # value of the left dipole
        Rdipole = states[right, RVvert[i]] # right dipole
        Udipole = states[RVhoriz[i], up] # upper dipole
        Ddipole = states[RVhoriz[i], down] # lower dipole
        # below is a lambda function to calculate the internal energy due to the dipole (according to eq (1) in notes)
        energy = lambda D : -1 * ((Ldipole * D) + (Rdipole * D) + (Udipole * D) + (Ddipole * D))
        initEnergy = energy(dipole)
        flipEnergy = energy(-1 * dipole)
        deltaEnergy = flipEnergy - initEnergy
        if deltaEnergy <= 0:
            states[RVhoriz[i], RVvert[i]] *= -1
        else:
            flipProb = np.exp(-beta * deltaEnergy)
            evaluation = np.random.uniform(0, 1)
            if evaluation <= flipProb:
                states[RVhoriz[i], RVvert[i]] *= -1
    return states
```

2.2 Crystal Size with Temperature

As was observed in the previous section, the size of the 'crystals' in the square lattice became larger as T decreased, and became larger faster at lower temperatures. It was observed that these parallel-aligned dipole crystals became significant for temperatures lower than $T = 3\epsilon/k$, and were a significant proportion of the total size of the lattice. This suggests that some critical temperature T_c exists between 2 and 3 ϵ/k, that is, $2\epsilon/k \leq T_c \leq 3\epsilon/k$, where the entire system will become aligned parallel at some spin value.

2.3 Thermodynamic Properties of the 2D Ising Model

In each of the figures below, the $N = 20^2$ population lattice was given 5×10^4 flip chances to approximate equilibrium, the $N = 50^2$ population 5×10^5 chances, and the $N = 100^2$ population 5×10^6 chances.

To begin with, the internal energy per dipole for the 2D Ising model was computed in a very similar way to equation (6), where the only difference was that neighbours both horizontally and vertically were taken into account (rather than just horizontally as in the 1D model).

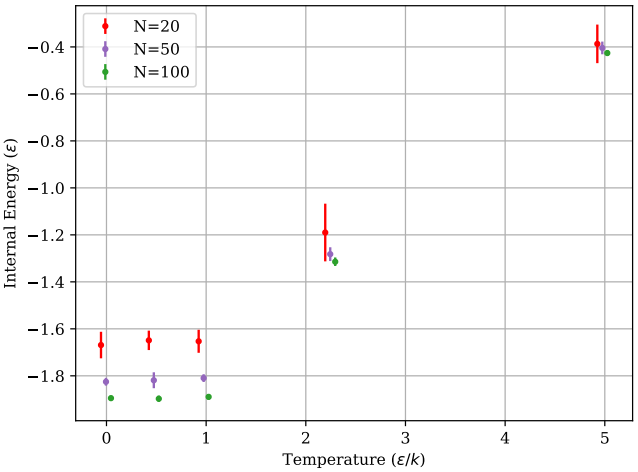


Figure. 9 The average internal energy per dipole for a square lattice with various side lengths, against multiple temperatures and 10 trials. Since all 3 population lattices were tested at the same temperature, the data points were shifted about the temperature axis slightly so that each data point could be discerned from the others. The tested temperatures in this plot and all of the other plots in this subsection were $T \in \{0.02, 0.5, 1, 2.27, 5\}\,\epsilon/k$.

The entropy per dipole of the lattice was computed in the same way as in equation (7), yielding the data in Figure 10.

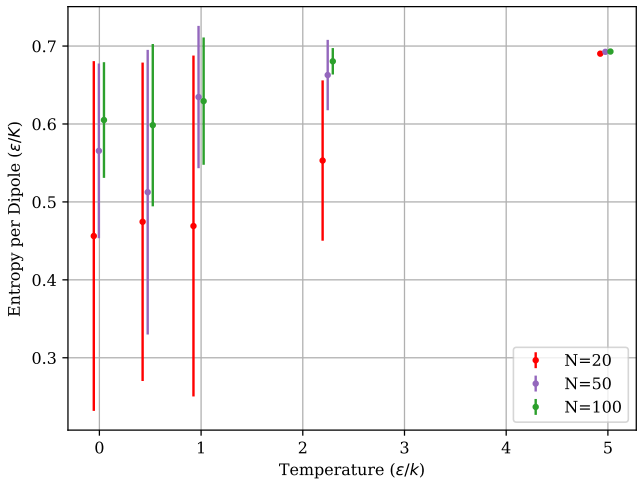


Figure. 10 Entropy per dipole of the 2D Ising lattice, with varying temperature and dipole population across 10 trials. The temperatures tested are the same as in Figure 9, with the same data-point temperature jitter.

The free energy per dipole was calculated in the same was as in the 1D case.

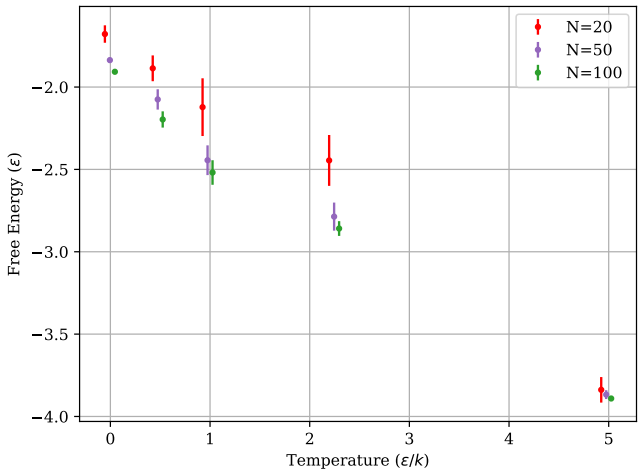


Figure. 11 Free energy per dipole of the 2D Ising lattice, with varying temperature and dipole population across 10 trials. The temperatures tested are the same as in Figure 9, with the same data-point temperature jitter.

The specific heat capacity was computed in the same was as in the 1D Ising model case.

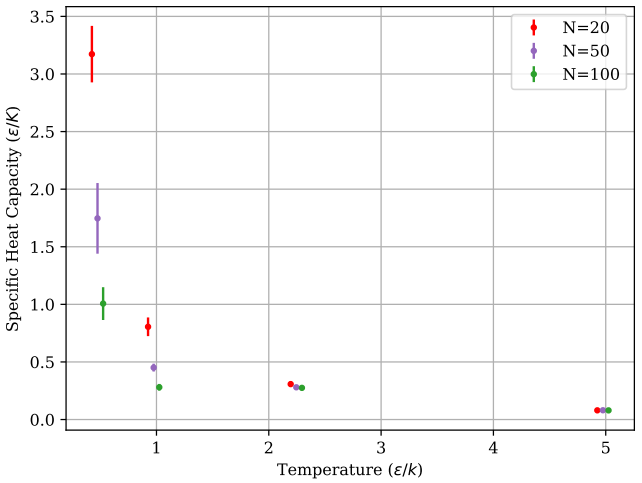


Figure. 12 Specific heat capacity of the 2D Ising lattice, with varying temperature and dipole population across 10 trials. The temperatures tested are the same as in Figure 9 (bar the first temperature, $T = 0.02\epsilon/k$, as the variance was unreasonably large), with the same data-point temperature jitter.

We note here that the specific heat capacity appears to be significantly lower for larger population lattices in the low temperature limit. At just over $T = 2\epsilon/k$ (i.e. at the critical temperature T_c), the heat capacity approximates the same value no matter the lattice population. As such, we'd expect that the specific heat capacity at T_c of an infinite 2D lattice would be this value too.
After this value of $\sim T_c$, we note that the specific heat capacity decreases consistently across population sizes.

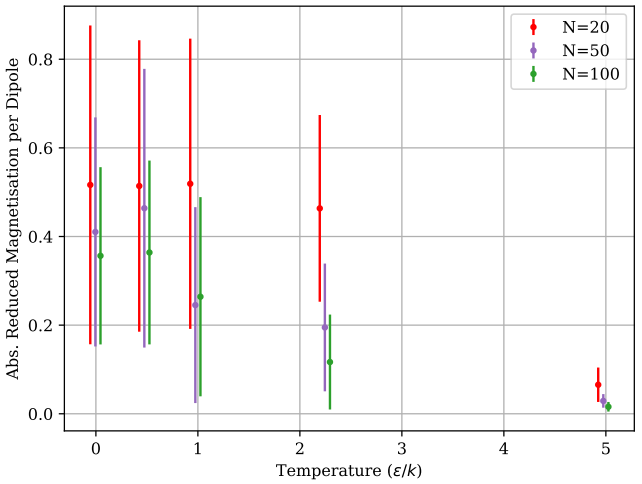


Figure. 13 Absolute value reduced magnetisation per dipole of the 2D Ising lattice, with varying temperature and dipole population across 10 trials. The temperatures tested are the same as in Figure 9, with the same data-point temperature jitter.

As with the entropy, the values of the absolute reduced

magnetisation per dipole are not exactly what we expected for the high population lattices. In the entropy case, we expected the entropy per dipole to be lower across the low temperature and high population data points. In the magnetisation case, we expected the absolute reduced magnetisation per dipole to be much larger for high population lattices. We attribute these discrepancies to a low number of spin flip chances. With more time permitting, we would be able to run 10 to 100 times as many iterations which we expect would correct these observations. In both circumstances, the high temperature data points align with expectations, however.

2.4 Mean Magnetisation for Different Sized Lattices

The mean positive and negative magnetisation per dipole for a range of temperatures and 2D lattice sizes is shown in Figure 14.

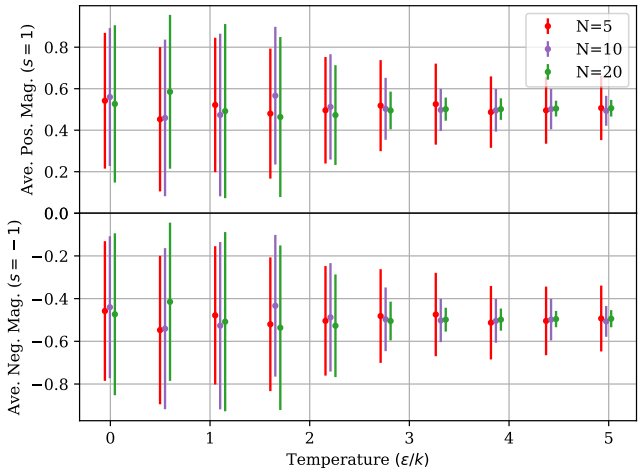


Figure. 14 Mean magnetisation per positive and negative dipole over a range of temperatures and population sizes. As in the previous subsection, the different population data points were scattered slightly about their true temperature value in order to discern the values and variance in the data points. The data points in the figure were averaged across 100 trials. The number of spin-flip chances given per system was dependent on population size, with $N_{\text{flip}} = N^2 \times 10^3$.

The variance of the data points (especially in the higher population lattices) begins to get smaller after $T = 2\epsilon/k$, which suggests that values are more consistently averaged at no net magnetisation for temperatures above about $T = 2.5\epsilon/k$. Furthermore, this suggests (along with previous tests on the model in the previous subsections) that the critical temperature lies between 2 and 2.5 ϵ/k; that is, $2\epsilon/k \leq T_c \leq 2.5\epsilon/k$.
Figure 14 shows that the average magnetisation is consistent against lattice populations, and so we claim that T_c is not dependent on population for the 2D Ising model.

2.5 Heating and Cooling of a Lattice

DNF
I apologise for the state of this draft, PHYS3020 has so many assignments and I'm struggling to keep up. This is the best I could do given the time frame and my other courses.