Investigating the Kosterlitz-Thouless Phase Transition in the 2D XY Model

Esha Sajjanhar, Keerthana Sudarshan
Ashoka University

Final Project Report for Statistical Mechanics (PHY-3610) Course
Written under the guidance of Professor Bikram Phookun and Philip Cherian
Submission Date: 14 December, 2023

ABSTRACT

In this project, we study the properties of the 2D XY model which is a lattice model of spins that can make any arbitrary angle with the X-axis. We simulate the system using Monte-Carlo sweeps and the Metropolis algorithm, and study the properties of this system both above and below the critical temperature at which this phase transition occurs. We also devise an algorithm to detect vortexes and antivortexes in the system, and thus observe changes to number of vortexes with temperature. We find that our results are largely in agreement with existing literature. The specific heat and susceptibility are found to have a peak above the critical temperature. The energy, magnetization and vorticity show significant change at the critical temperature $T_{KT} \approx 0.88$, and the correlation length diverges at this temperature as well. Finally, the critical exponent of susceptibility ν is found to be near 0.7.

1 Introduction

Lattice systems are a vital tool of theoretical and computational physics. Such models are very well suited to computational simulations due to their fundamentally discrete nature. One of the best-known lattice models in statistical physics is the 2D Ising model. The Ising model is a system of interacting spins arranged on a lattice, resembling a ferromagnetic system. Each spin has two possible states- 'up' (+1) or 'down' (-1), and can interact with its nearest neighbours. The Hamiltonian of the system is given by,

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i \cdot s_j - h \sum_j s_j \tag{1}$$

where J gives the strength of the spin-spin interaction while h gives the strength of the external magnetic field. The energy of the system is minimised when neighbouring spins point in the same direction. This simple model provides an example of a phase transition.

Another interesting lattice system is the 2D XY model. The 2D XY model is a lattice system very similar to the 2-state Ising model, with the change that the spins on the lattice are not restricted to two states. Each spin can make an arbitrary angle $0 \leq \theta \leq 2\pi$ with the x-axis. This system was studied by Michael Kosterlitz and David Thouless who found that it undergoes a topological phase transition which was later called the *Kosterlitz-Thouless phase transition.* This transition between ordered and disordered states of the system occurs due to topological defects known as 'vortexes' in the lattice. Kosterlitz and Thouless won the 2016 Nobel Prize for their theoretical description of the KT transition in this model. Certain superfluids and liquid crystals can be modelled by the planar XY model and exhibit this phase transition, so studying vortexes in these simulated models can help us understand the properties of the corresponding physical continuous models.

2 Model

2.1 2D XY Model

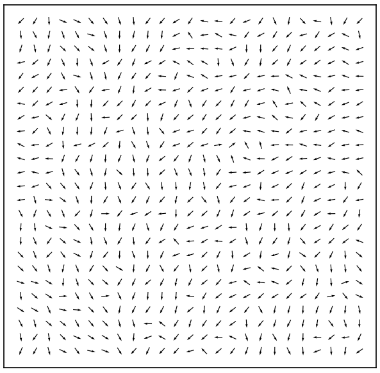


Figure 1: 2D XY model on a 25 x 25 lattice.

As discussed earlier, spins in the 2D XY model can make an arbitrary angle $0 \leq \theta \leq 2\pi$ with the x-axis. We can write each spin s_i as a 2-D vector $\boldsymbol{s_i}$ given by $(\cos\theta_i, \sin\theta_i)$. The most general form of the Hamiltonian for such a system is given by,

$$\mathcal{H} = -\sum_{i,j=nn(i)} J_{ij} s_i \cdot s_j - \sum_j h_j s_j, \tag{2}$$

where J_{ij} gives the strength of the interactions between spins i, j and h_j is the external field at the lattice site j. In our project, we have no external field, and set the interaction term $J_{ij} = 1 \ \forall \ i, j$. Thus, the Hamiltonian takes the form,

$$\mathcal{H} = -\sum_{i,j=nn(i)} \cos(\theta_i - \theta_j). \tag{3}$$

In order to solve this analytically, we approximate the discrete lattice to a continuous system, and take $\cos(\theta_i - \theta_j) \approx 1 - \frac{1}{2}(\theta_i - \theta_j)^2$. This gives a Hamiltonian of the form:

$$\mathcal{H} = 2N + \frac{1}{2}\int d\mathbf{r}(\nabla\theta)^2 \tag{4}$$

$2N$ is the ground state energy of the system when all the spins are completely aligned with each other. The solutions corresponding to the local minimum of the Hamiltonian satisfy the equation $\nabla^2\theta(\mathbf{r}) = 0$. In the surface integral for $\mathcal{H}$, r goes from a to L, where a is the minimum separation of spins in the lattice, and L is the size of the lattice. In this project, we consider the 2D XY model on a square lattice such that each spin has four nearest neighbours. In order to emulate an infinite lattice, we use periodic boundary conditions.

We model the evolution of such a 2D XY model at a range of temperatures, and look at various characteristics of the system including energy, magnetization, specific heat and susceptibility. The energy is calculated very simply using the Hamiltonian formula. The magnetization is given by the sum of all the spins on the lattice. In our system, each spin is labelled by its angle θ from the x-axis, so the magnetization is calculated as:

$$\langle M \rangle = \left| \left(\sum_i \cos\theta_i, \sum_i \cos\theta_i \right) \right| \tag{5}$$

The specific heat at constant volume (in this case, constant spin density) is property of fluctuations in the average energy, and thus is a function of the variance of energy. It is calculated as:

$$C_v = \frac{\sigma_E^2}{T^2} = \frac{\langle E^2 \rangle - \langle E \rangle^2}{T^2} \tag{6}$$

The magnetic susceptibility of the system is similarly calculated from the variance of the magnetization (though it is divided by T rather than T^2). However, we expect that at high temperatures, there is large degree of disorder in the system- that is, all the spins are randomly aligned, and thus the overall magnetization $\langle M \rangle$ of the system goes to zero. Therefore, the susceptibility is often taken to be:

$$\chi = \frac{\langle M^2 \rangle}{T}$$

However, in practice we notice a considerable difference between $\langle M^2 \rangle / T$ and σ_M^2 / T, so we will examine each of these separately.

2.2 Phase transitions and Vortexes

Many spin models such as the Ising model, the XY model and the Heisenberg model show phase transitions at a finite temperature. This is a change in the characteristics of the system such as a shift from an ordered state (the spins are highly aligned and in the local minima of the Hamiltonian) to a disordered state (spins are highly disordered and the system is in an energetically unfavourable state).

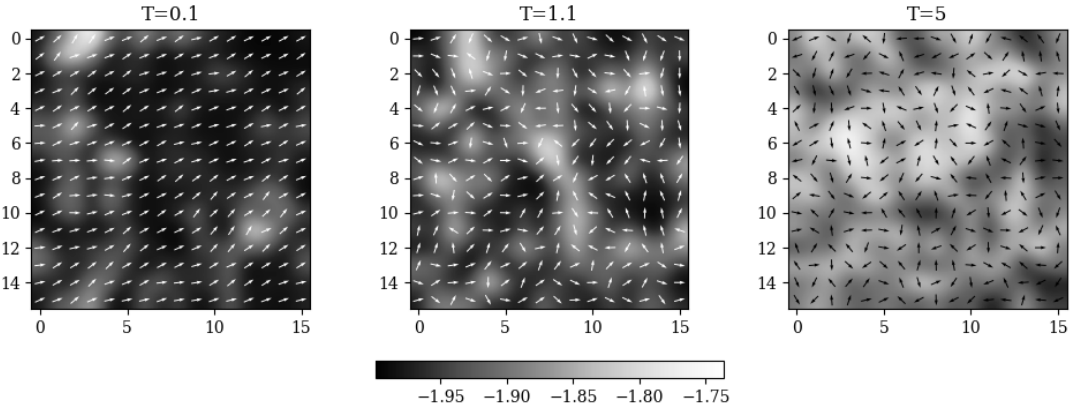


Figure 2: A 16x16 planar XY model, simulated at 3 different temperatures, showing ordered spins, formation of domains, and disordered state. Below the spin lattice, we plot an interpolated energy map of the system, showing the energy in different parts of the lattice. In the plots above, lighter areas correspond to higher energy and darker areas correspond to lower energy.

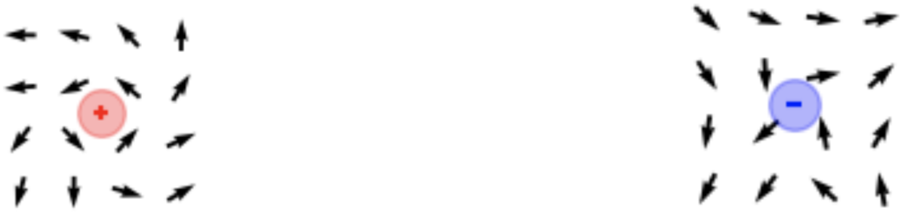
The 2D XY model, compared to higher or lower dimensions, shows a peculiar phase transition. In one dimension, there is no phase transition at all. In three and higher dimensions, the XY model exhibits a phase transition from ferromagnetic characteristics arising from spontaneous magnetization at low temperatures, to paramagnetic characteristics at higher temperatures. In 2D, the XY model exhibits the Berezinski-Kosterlitz-Thouless transition, which is a topological phase transition occurring due to the presence of *vortex-antivortex pairs*.

Returning to the Hamiltonian, we saw that the condition for the minima was $\nabla^2 \theta(\mathbf{r}) = 0$. This is satisfied by one of two states: either $\theta(r)$ should be a constant which is the ground state of having all spins aligned, or it should satisfy the boundary condition:

$$\oint \nabla \theta(\mathbf{r}) \cdot d\mathbf{l} = 2\pi n \tag{7}$$

where n is any integer. $n = 0$ corresponds to spins not having any net rotation around the closed loop. If $n \neq 0$, there is a net rotation of 2π, which is the condition for the presence of a vortex within the loop.

A vortex is topological defect in the spin system, that is formed when the spins take angles such that when traversing a closed loop in a counterclockwise direction, they rotate by a multiple of 2π (Fig. 3a). Vortexes behave like a virtual particles- they can move around on the lattice and collide with other vortexes. They are a fixed feature of the system, do not disappear trivially. They can however annihilate when they collide with an antivortex, which is another virtual particle where the spins rotate by -2π in a closed loop (Fig. 3b). For both vortexes and antivortexes, the integer n in the closed loop integral is known as the charge of the vortex.



(a) Vortex generated in a planar XY model. The spins rotate by $+2\pi$ in a counterclockwise direction.

(b) Antivortex generated in a planar XY model. The spins rotate by -2π in a counterclockwise direction.

In general, vortexes are not formed at low temperatures as they are a distortion in the field of spins, and thus add a large amount of energy to the system which is not a favourable state. The energy of a vortex is estimated as follows:

Since the vortex is spherical symmetric, the integral 7 depends only on the magnitude r rather than the vector $\mathbf{r}$. Thus, we can write:

$$\oint \nabla\theta(r) \cdot d\mathbf{l} = 2\pi r|\nabla\theta| = 2\pi n \implies |\nabla\theta| = \frac{n}{r} \tag{8}$$

Substituting this into the equation of the Hamiltonian:

$$E_v = E_0 + \frac{J}{2}\int d\mathbf{r}(\nabla\theta(r))^2 = E_0 + \frac{J}{2}\int_0^{2\pi}\int_a^L r dr d\theta \frac{n^2}{r^2} = \pi n^2 J \ln L/a \tag{9}$$

For practical purposes we can simply set $n = 1$, as vortexes of greater than unit charge are very rare in a system. Thus, we have:

$$E_v = E_0 + \pi J \ln L/a \tag{10}$$

If we look at a vortex-antivortex pair and take a large enough closed loop such that it encircling both, the total rotation of angles will cancel $((+2\pi) + (-2\pi) = 0)$ and the closed integral is zero. The energy of a such a pair is given to be:

$$E_{\text{pair}} = 2E_c + E_1 \ln R/a \tag{11}$$

E_c is the energy of each vortex core and R is the distance between the vortex and antivortex. The energy of a bound pair (R is not very large) is lower than that of independent vortexes. Thus, the existence of bound vortex-antivortex pairs is favourable at much lower temperatures compared to free vortexes.

The free energy can be obtained from the thermodynamic relation $F = E - TS$, where S is the entropy of the system. Looking at an $L \times L$ portion of the lattice, the number of places where the vortex center might be located is L^2/a^2 (a is the lattice separation). Thus, the estimated entropy is $S = 2k_B \ln (L/a)$, and the free energy of a vortex is:

$$F = E_0 + (\pi J - 2k_B T) \ln (L/a) \tag{12}$$

Thus, we will see a change in behaviour at some non zero temperature. For $T < \pi J/2k_B$, second term is positive and so the free energy increases as L increases. Since the system seeks to be in a state of minimum free energy, in an infinite system it is not favourable to have a large number of vortexes. On the other hand, for $T > \pi J/2k_B$, the second term has a negative contribution. Thus, at higher temperatures, adding vortexes to the system reduces the free energy.

These two ideas lead to the argument for the KT phase transition. At low temperatures, the vortexes and antivortexes exist strictly in bound pairs that do not move too far from each other, and there are very few of these pairs at all. As the temperature increases, more pairs may be allowed to form, but they still exist in the same bound state. Above some critical temperature T_{KT}, The vortexes suddenly unbind and move about freely, and a huge number of new vortexes are formed. This sudden unbinding of vortex-antivortex pairs and the sudden increase in the number of free vortexes is the main characteristic of the KT transition.

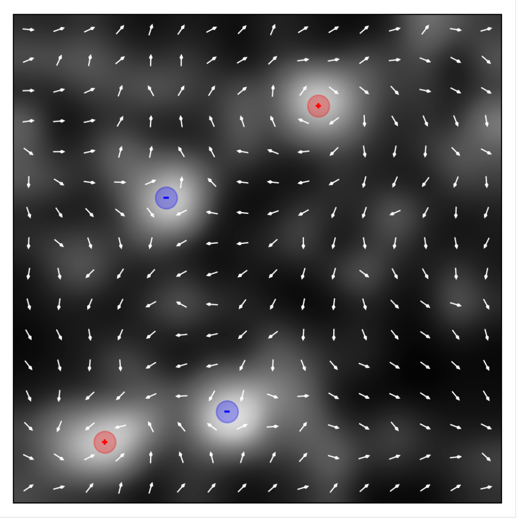


Figure 4: Simulation of a 16x16 lattice at $T = 0.4$. Vortexes and anti-vortexes are marked out with red and blue respectively. An energy map is shown below, with higher energies in lighter colors.

The temperature at which this phase transition occurs was estimated theoretically by Mattis (1984) using the transfer matrix method, and found:

$$\frac{k_B}{J} T_{KT} = 0.8816 \tag{13}$$

Another indicator of a phase transition in spin systems is a peak in the specific heat vs. temperature graph. However, another peculiarity of the 2D XY model is that this peak does not seem occur at the critical temperature T_{KT}. Monte-Carlo simulations have shown consistently that a smooth peak in the specific heat consistently appears around 15-20% above the critical temperature, near $T = 1.1$. The same is observed in the magnetic susceptibility. Thus, these measures cannot be used to estimate the critical temperature.

Another characteristic of the phase transition is a divergence of the correlation length ξ. The correlation function $C(r)$ is a measure of how much spins separated by a distance influence each other. The correlation length is the characteristic length over which the correlation function decays. We evaluate the spin-spin correlation function as,

$$C(r) = \langle s_i s_j \rangle - m^2 = \cos(\theta_i - \theta_j) - m^2, \tag{14}$$

where m is the average magnetization per spin in the system. We expect that the correlation function will follow a power-law decay for temperatures below the critical temperature and show exponential decay above the critical temperature.

We expect that the system will be infinitely responsive at the critical temperature, and every spin will affect every other spin in the lattice- thereby giving a correlation length of the order of the lattice size. For an infinite lattice, the correlation length would go to infinity. We can plot the correlation length against temperature to find the divergence at the critical temperature. Thus, the system has a disordered form for high temperatures and an ordered form for low temperatures.

3 Methods

3.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a Markov-chain Monte Carlo method which is useful for importance sampling. It is of immense use in Statistical Physics because it allows us to obtain the values of average quantities using a limited number of samples. To simulate physical systems, we can draw samples from a weighted distribution which accurately represents the likelihood of different microstates. Here, we attach the Boltzmann weight to each microstate and use this to determine whether the state is accepted by the system.

To evolve the system through a single Monte-Carlo sweep, we execute a trial flip on a random spon in the lattice. The trial flip is immediately accepted if it lowers the energy of the system or leaves it unchanged. If not, it is accepted with a probability given by $e^{-dE/T}$ where dE is the change in energy due to the trial and T is the temperature of the system. This is repeated N times for N spins in the lattice and constitutes a single Monte-Carlo sweep. We execute around 10,000 such sweeps to allow the system to reach equilibrium. All subsequent analysis and calculations are carried out after these sweeps have been completed.

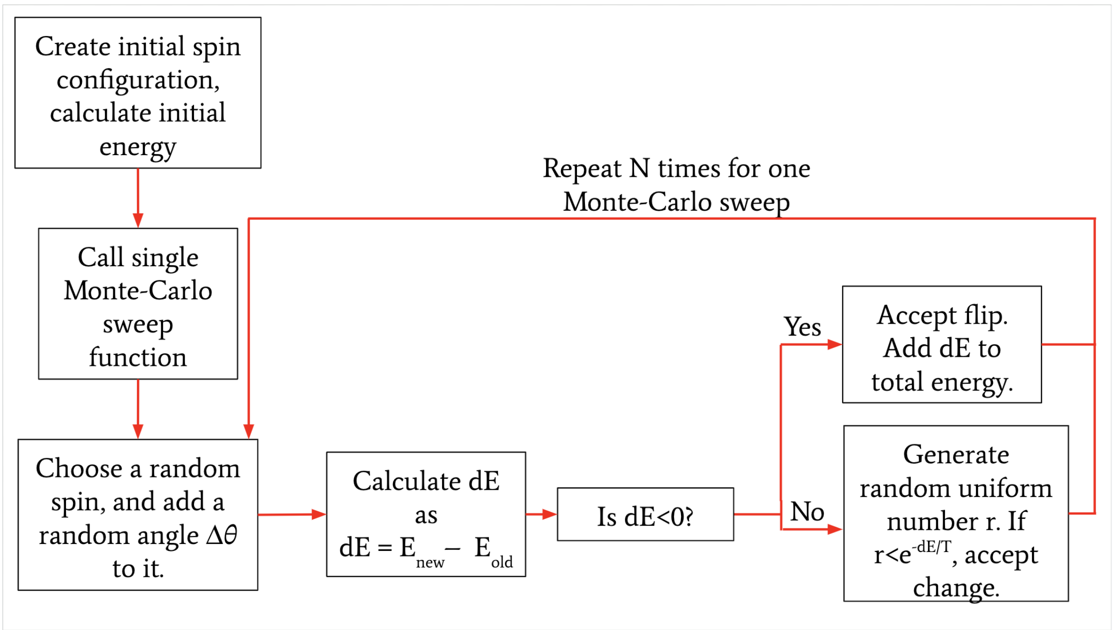

Create initial spin configuration, calculate initial energy
Call single Monte-Carlo sweep function
Repeat N times for one Monte-Carlo sweep
Choose a random spin, and add a random angle $\Delta\theta$ to it.
Calculate dE as $dE = E_{new} - E_{old}$
Is dE<0?
Yes
No
Accept flip. Add dE to total energy.
Generate random uniform number r. If $r<e^{-dE/T}$, accept change.

Figure 5: Flowchart illustrating the Metropolis-Hastings algorithm as used in our simulation.

3.2 Detecting Vortexes

Theoretically, in a continuous system, the presence of a vortex or anti-vortex is checked by looking at the spins in a closed loop, and checking if the satisfy the equation:

$$\oint \nabla\theta(\mathbf{r}) \cdot d\mathbf{l} = 2\pi n \tag{15}$$

where n is any integer value. If n is positive, the closed loop encircles a vortex. If n is negative, it encircles an anti-vortex, and if n is zero, there is no distinct topological defect. However, in a discrete system, we need to interpret the closed integral in a different way. The following algorithm was used to detect vortexes in a planar XY model.

(For this algorithm, we require all spins to have an angle between 0 and 2π with respect to the x-axis. Each spin is labelled only with its angle, since all are of unit length.)

1. First, we loop over all 2x2 plaquettes in the system.
2. For each plaquette, we calculate differences in the angles of the spins going in a anti-clockwise order. That is, if the spins are labelled θ_1, θ_2, θ_3, and θ_4 (going in anti-clockwise order), then we calculate $\Delta_1 = \theta_2 - \theta_1$, $\Delta_2 = \theta_3 - \theta_2$, $\Delta_3 = \theta_4 - \theta_3$, and $\Delta_4 = \theta_1 - \theta_4$.
3. Then, we calculate the product of these spins, $P = \Delta_1 \cdot \Delta_2 \cdot \Delta_3 \cdot \Delta_4$, and check its sign. We also calculate their sum as $S = |\Delta_1 \% \pi| + |\Delta_2 \% \pi| + |\Delta_3 \% \pi| + |\Delta_4 \% \pi|$

   - If $P > 0$ or $S \neq 2\pi$: There is no vortex.
   - If $P < 0$ and $S = 2\pi$: There is a vortex or an anti-vortex in the 2x2 plaquette.

   The logic behind this calculation is this: in order to see a vortex, we want to see the spins rotate *continuously* by an angle of $2\pi n$ as we traverse the loop, so obviously we would want to check that the total of the difference is 2π. But we also want to ensure that the spins do not turn back at any point on the loop. The spin angle should continuously increase or decrease without interruption. Thus, when there is a vortex/anti-vortex, the angle differences should be all positive or all negative with a single exception- when the spins cross the x-axis. Thus, the angle differences around a vortex will have a single negative quantity, and the angle differences around an anti-vortex will have a single positive quantity, and in both cases the product is negative.
4. If a vortex or anti-vortex is detected, the coordinates of this vortex (the center of the plaquette) are added to a list, and can be plotted on a image of the spins to show the vortex.

3.3 Temperature Quench

As mentioned, adding a vortex to the system adds a large amount of energy, as it distorts the spins from their preferred state of alignment. Thus, the probability of obtaining a vortex-anti-vortex pair in a low temperature system in very low, and we would have to simulate the system for a long time in order to detect any. In order to see vortexes in shorter time scales, we do what is known as a temperature quench (as described in Tobochnik and Gould, pg. 300). To do this, we initially equilibrate the system at a temperature far above the critical temperature, say around $T = 2.5$. Then, we suddenly drop the temperature to below the critical temperature (by simply changing T in the Metropolis-Hastings algorithm) and equilibrate at this new T. At the higher temperature, there would have been much more randomness in the orientation of the spins, and many unstable vortexes would form. Lowering the temperature after this will cause many of the unstable vortexes to disappear, but stable equilibrium vortexes that are true topological defects will remain.

4 Results

4.1 Energy and Specific heat

Below, we have plotted the average energy per spin and the specific heat per spin against temperature, for a range both above and below the critical temperature. The simulations were done for a square lattice of 2500 spins. At each temperature, the system was equilibrated for 10,000 Monte-Carlo sweeps, and then run for a further 30,000 steps while calculating the energy and magnetization. The averages and variances were calculated from these data of these 30,000 steps.

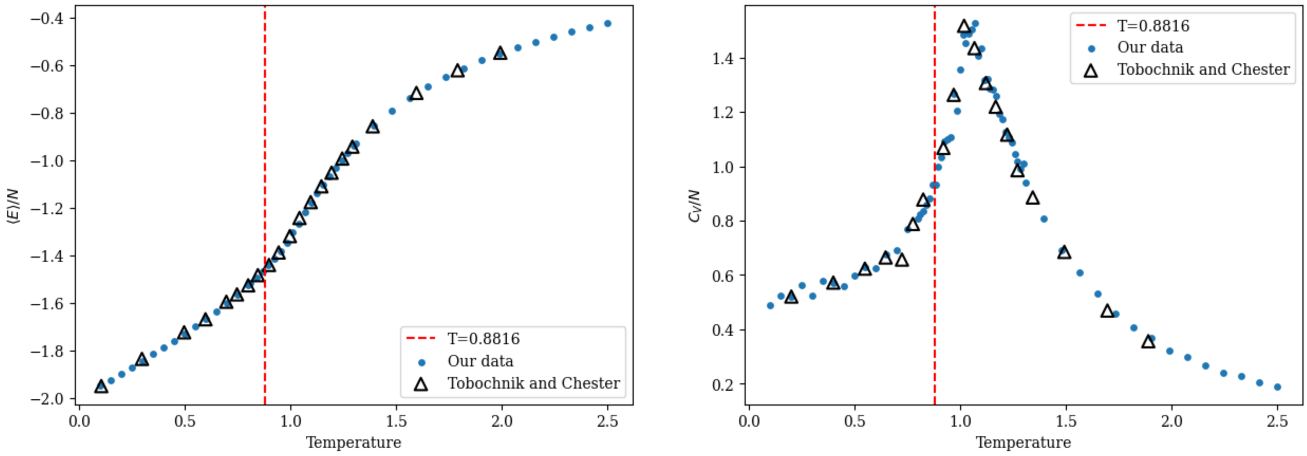


Figure 6: Average energy per spin and the specific heat per spin for 2500 spin system, plotted against temperature. Data obtained by Tobochnik and Chester are plotted alongside. The critical temperature T_{KT} is also marked for reference.

We find that the behaviour of $E(T)$ changes at T_c, where we observe an inflection point. This curve is in close agreement with the results obtained by Tobochnik and Chester (their data plotted alongside results from our own simulations).

Looking at the variation on specific heat, we can see that the peak is observed slightly above the critical temperature, as expected, and once again the results are in close agreement with was obtained by Tobochnik and Chester. In the graph below, the peak is found to be near $T \sim 1.08$

4.2 Magnetization and Susceptibility

The magnetization and susceptibility (σ_M^2/T) were calculated in a similar manner to the energy and specific heat, from the same system (2500 spins) with the same number of Monte-Carlo sweeps.

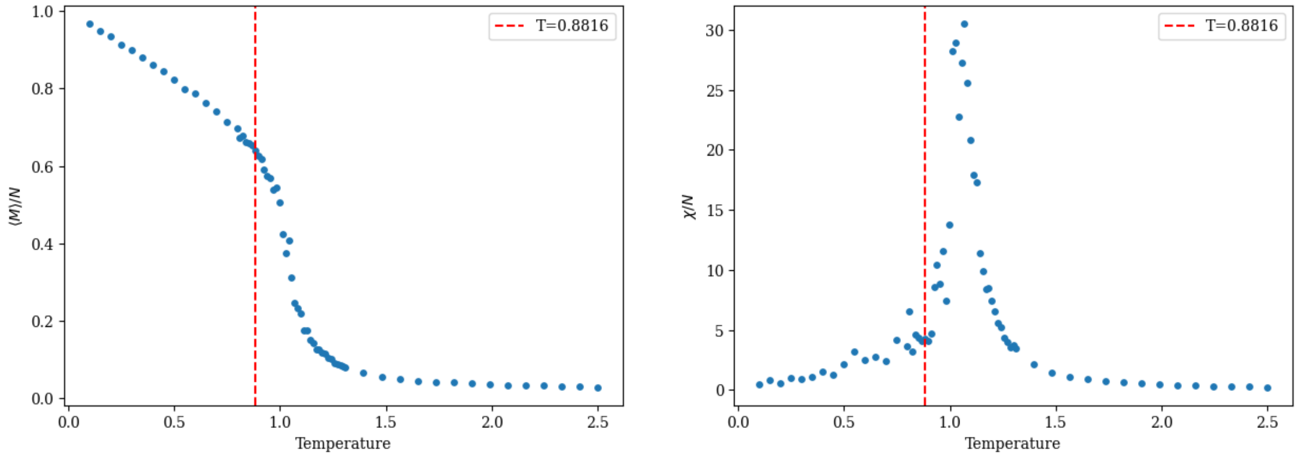


Figure 7: Average magnetization per spin and the susceptibility per spin for 2500 spin system, plotted against temperature. The critical temperature T_{KT} is also marked for reference.

We find that the behaviour of $M(T)$ also changes at T_{KT}. For very low temperatures, the magnetization seems to decrease more or less linearly with increasing temperature. At T_{KT}, the behaviour changes from linear into an exponential fall. At very high temperatures, the magnetization goes to zero, as expected from a system with a high degree of disorder in the spins. The susceptibility shows a very sharp peak (compared to the more rounded peak of the specific heat) above the critical temperature. Both the specific heat and susceptibility consistently peak at the same temperature, which typically falls between $T = 1.05$ and $T = 1.15$.

4.3 Critical Exponents

In the planar XY model, we expect to see an exponential decay in the susceptibility just after the critical temperature, which is described by the equation:

$$\chi \sim e^{b/t^{\nu}} \tag{16}$$

t is the reduced temperature of the system, and is given by $t = (T - T_C)/T_C$ where T_C is the critical temperature. b is a constant specific to the system, and ν is the critical exponent of the system, dictating the exponential fall. We can see this in the higher temperature region of the susceptibility graph above. However, in order to look at the region directly after the critical temperature, we cannot use the susceptibility directly as we would encounter the discontinuity above the critical temperature. Therefore, we must instead look at the square of the magnetization alone: $\langle M^2 \rangle / N$. This is also acceptable since the magnetization goes to zero above the critical temperature, so the susceptibility and $\langle M^2 \rangle$ will match. In the graph below, the values of $\langle M^2 \rangle$ were calculated for the same system (2500 spins), and for the same number of Monte-Carlo sweeps as the previous results.

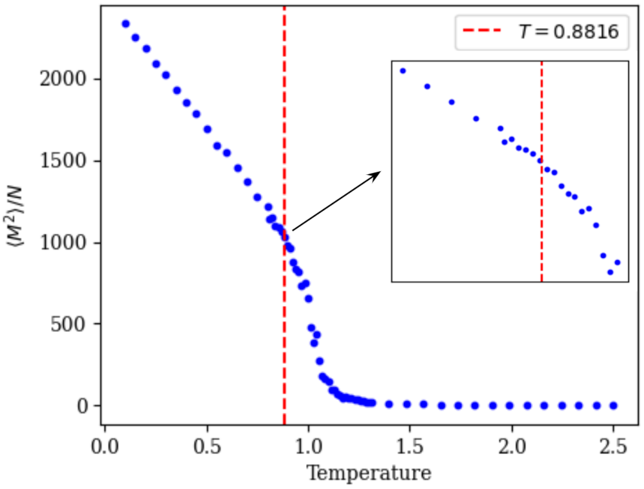
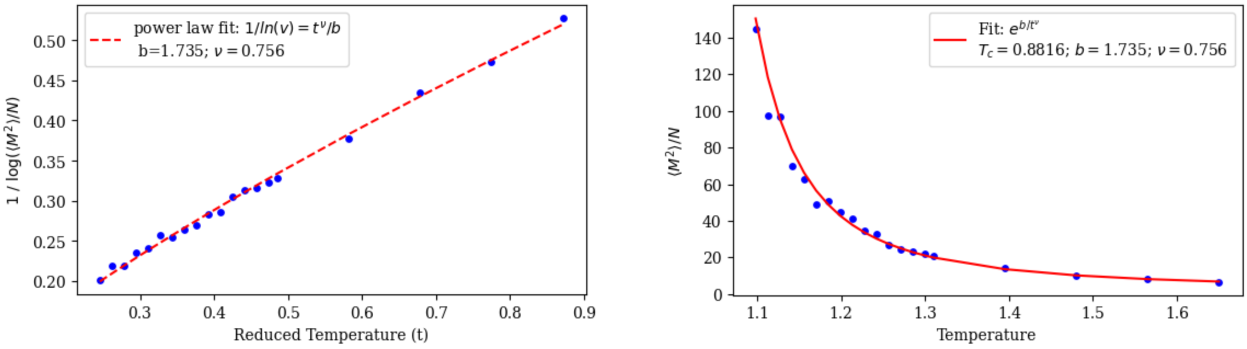


Figure 8: Average square of magnetization per spin plotted against temperature, with critical temperature T_{KT} marked. Inset graph shows a close-up of the behaviour change around T_{KT}

The square of the magnetization (Fig. 8) shows a very similar behaviour to the magnetization- initially linear with temperature, then a sudden change to exponential decay at the critical temperature, rapidly falling to zero for higher temperatures. Now, in order to find the critical exponents we want to look at the behaviour just after the critical temperature, from around $T = 1$. We first want to convert the equation into a form we can more easily fit.

$$\langle M^2 \rangle = e^{b/t^{\nu}} \implies \frac{1}{\ln\left(\langle M^2 \rangle\right)} = \frac{1}{b} t^{\nu}$$

By plotting $1/\ln(\langle M^2 \rangle)$ vs. the reduced temperature, we can do a fit for a power law equation using the `curve_fit` function from the `scipy` library (of course, in order to plot against t we would need to assume the critical temperature, which we take to be $T_C = 0.8816$). This graph is shown in figure below, along with the plot of $\langle M^2 \rangle / N$ versus T with the exponential fall calculated from the `curve_fit` parameters.



(a) Power-law fit of $1/\ln(\langle M^2 \rangle/N)$ v.s. t. The values of $\langle M^2 \rangle$ are taken from the region directly after the critical temperature.

(b) Plot of $\langle M^2 \rangle/N$ v.s. T. The fitted line is drawn using the parameters b and ν found from the curve-fit of the power law graph.

From the graphs above, we obtain the values:

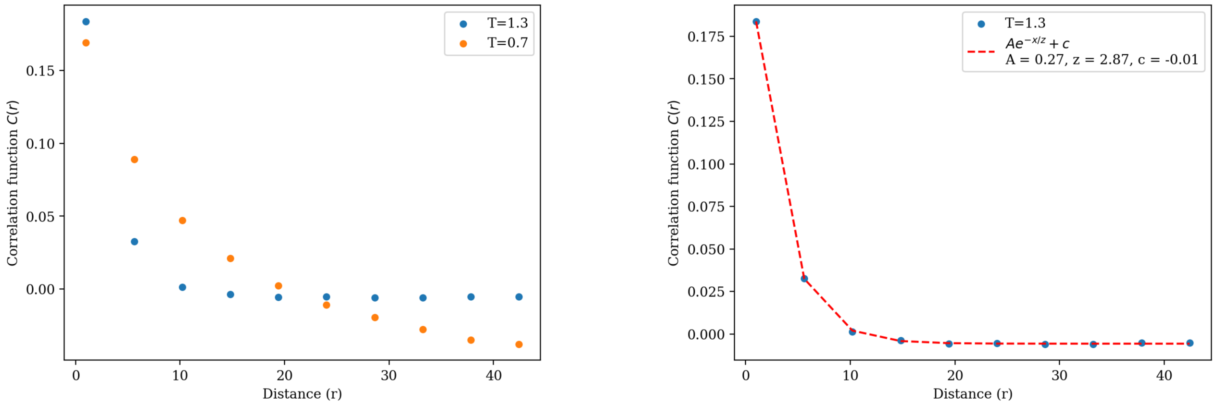
$$b \approx 1.735 \ ; \ \nu \approx 0.756 \tag{17}$$

This is similar to values obtained from other simulations. Tobochnik and Chester found that the best fits gave the critical value ν between 0.48 and 0.70, and the value of the constant b between 1.10 and 1.70. Thus, our values would seem to be fairly accurate.

An important note to make is that the KT theory predicts that the critical exponent for susceptibility should be $\nu \sim 0.5$. However, computer simulations have often found that $\nu \sim 0.7$ gives a better fit. Our value of $\nu = 0.756$ is largely in agreement with this.

4.4 Correlation length and critical temperature

Above the critical temperature, we expect $C(r)$ to decay exponentially as $e^{-\frac{r}{\xi}}$ where ξ is the correlation length. ξ gives the length scale over which any spin in the lattice can affect another spin. We can obtain ξ by fitting an exponential curve of the form $Ae^{r/\xi} + c$ to $C(r)$ for temperatures above T_c.[1] To obtain the curve of best fit, we use the `curve_fit` function from the `SciPy` library.



(a) Correlation function against distance. The curve falls exponentially for $T = 1.3$ and as a power law for $T = 0.7$.

(b) Fitting an exponential curve to the correlation function in order to get the correlation length ξ.

Figure 10: Correlation functions calculated for a 60x60 lattice. We consider one temperature above $T_c = 0.88$ and one below.

It is important to note that the correlation length, as defined here, can only be evaluated for $T > T_c$, since $C(r)$ is an exponential function of r only in this region. Thus, to find the divergence in ξ, we need to evaluate ξ at temperatures close to T_c. As shown below, we do find that the correlation length shoots up, but we do not see it increase to the order of the lattice size. Additionally, we find a peak at $T \sim 0.95$ instead of $T_c = 0.88$. However, this can be understood in terms of Tobochnik and Chester (1979) who found that the critical temperature lies between 0.85 and 0.95. The subsequent exponential decay of the function is a characterstic

[1] Tobochnik and Chester (1979) evaluated the correlation function using a slightly different convention. This has been discussed in the Appendix.

feature of the KT transition which can be understood in terms of the presence of unbound vortices. This has been discussed in [4].

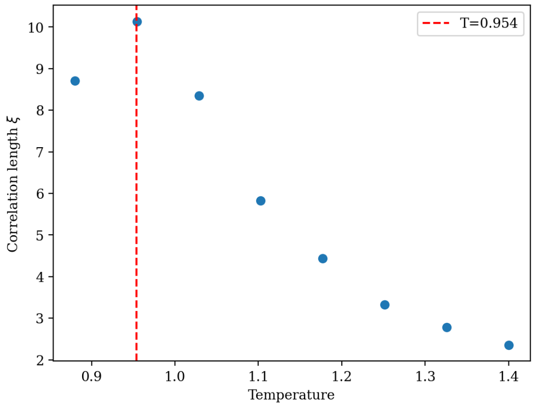


Figure 11: Correlation length as a function of temperature for a 60x60 lattice. The correlation length is highest near the critical temperature and then falls exponentially as a function of temperature.

4.5 Vorticity

Lastly, we look at the vorticity, or the number of vortexes per spin of the system at different temperatures. We calculated the vorticity as well as the free vorticity (the number of unpaired vortexes or antivortexes) for a system of 1024 spins (32 x 32 lattice). The system was equilibrated for 1,000 Monte-Carlo sweeps at each temperature, and averages of vorticity were calculated from the next 6,000 sweeps. In this simulation, the initial lattice configuration used for each subsequent temperature value was the final lattice configuration of the previous temperature.

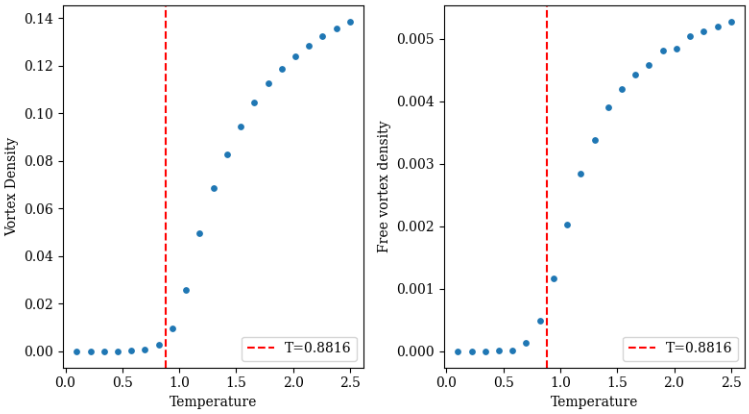


Figure 12: Graphs of vorticity and free vorticity at different temperatures, with the critical temperature marked. Data generated from 1024 spin system.

As expected, below the critical temperature there are very few vortex pairs, and even fewer free vortexes. At the critical temperature, we observe a sudden rise in both the vorticity and the free vorticity, which indicates the unbinding of vortex-antivortex pairs, as well as the generation of many new vortexes. This is characteristic of the KT phase transition expected at this temperature. At much higher temperatures, the increase in vorticity begins to decreases, which indicates the lattice reaching a saturation point in the number of vortexes.

Just below the critical temperature before the phase transition, the vorticity v is expected to change as:

$$v \sim e^{2\mu/T} \tag{18}$$

2μ is the energy required to create one vortex-antivortex pair. We can check this relation by plotting a graph of $\ln v$ versus $1/T$, shown below. We have also compared this data with the data obtained by Tobochnik and Chester.

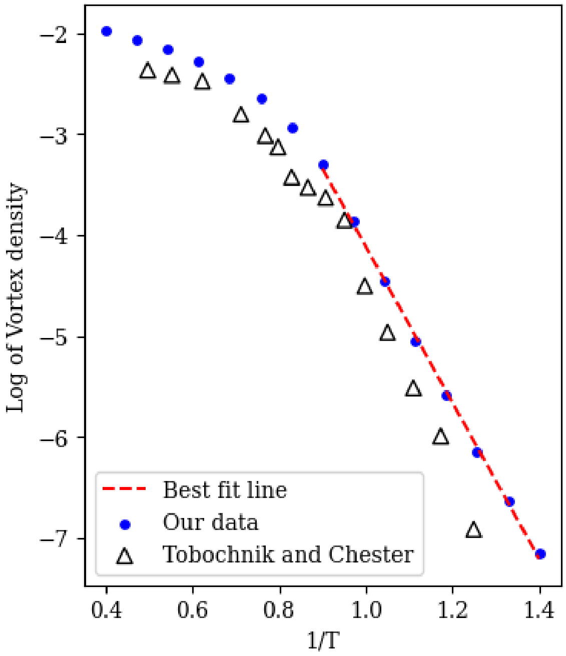


Figure 13: Graph of $\ln v$ versus $1/T$, with data from Tobochnik and Chester (1979), and best fit line of the linear portion below T_{KT}. Slope of the line = -7.74

As seen in the graph above our data is in close match with the existing literature data, and we obtain the energy of a vortex pair to be:

$$2\mu \approx 7.740 \tag{19}$$

Tobochnik and Chester obtained $2\mu \approx 9.4$, while the theoretically predicted value from Kosterlitz and Thouless's work is $2\mu = 10.2$. Therefore, our value is close to the known values, though the error margin is significant.

A feature of the graph to note is that at higher temperatures, although it no longer fits a straight line, the slope of the graph decrease in magnitude, which could indicate that the energy required required to form new vortexes is decreasing. This is what we would expect from the KT theory.

5 Conclusion

In our study of the 2D XY model, we found that the energy and magnetization of the system show a change in their behaviour near the critical temperature. We also use the magnetization as a function of temperature to find the critical exponents for the system. We obtained the values $b = 1.735$ and $\nu = 0.756$, which are in agreement with data from other simulations. The correlation length increases near the critical temperature and subsequently falls rapidly. We also find that there are many vortex-antivortex pairs at low temperatures, and free vortexes only exist at higher temperatures. The energy required to create a vortex-antivortex pair at low temperatures was found to be ~ 7.74. We observe the existence of closely bound pairs at low temperatures and the unbinding of these vortex pairs at the critical temperature $T_{KT} = 0.88$ These results together help us examine the KT phase transition.

6 Acknowledgements

We would like to thank Professor Bikram Phookun for giving us the opportunity to do this project, as well as the course TF Philip Cherian for his guidance and advice on various aspects of the project. We would also like to thank Professor Somendra Bhattacharjee for his advice.

7 References

1. *Statistical and Thermal Physics: With Computer Applications, Second Edition* by Tobochnik & Gould
2. *Monte Carlo study of the planar model*, Tobochnik & Chester, Physical Review B, 1979.
3. *First-order transition in a 2D classical XY-model using microcanonical Monte Carlo simulations*, Ota et. al, Pramana Journal of Physics, 1993.
4. Henrik Jeldtoft Jensen's notes on *The Kosterlitz-Thouless Transition.*
5. *Kosterlitz-Thouless Transition* by the Azimuth project
6. *Transfer matrix in plane-rotator model*, by Mattis, Daniel C., Physics Letters A, September 1984
7. Wikipedia: Classical XY Model

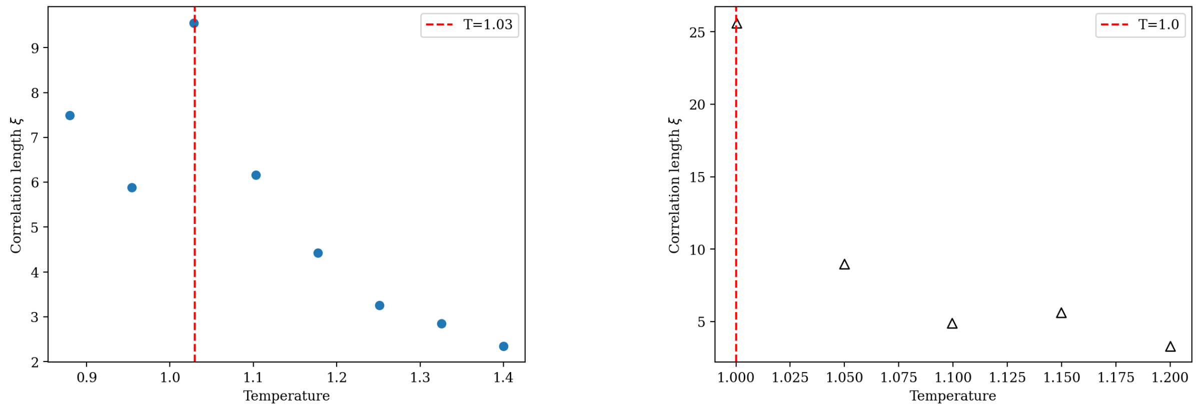
8 Appendix

8.1 Correlation function

Tobochnik and Chester (1979) evaluated the correlation function as,

$$C(r) = \cos(\theta_i - \theta_0), \tag{20}$$

where average magnetization per spin, m, is set to 0 and the second spin in the pair is always the first spin in the lattice. Using this method, we replicate the form of their results but do not find similar values.



(a) Correlation length against temperature plotted using an alternate algorithm.

(b) Correlation length against temperature as presented by Tobochnik and Chester (1979).

Figure 14: Correlation lengths calculated for a 60x60 lattice showing our data as well as Tobochnik and Chester's results.

While this method of evaluation should be equivalent to ours, we obtained somewhat inconsistent results using this scheme. We found that our results varied significantly between subsequent runs of the functions. This could be due to fluctuations in the value of the first spin in the lattice or due to issues with the exponential fitting method. We have not been able to identify the cause of this discrepancy.

8.2 Python Code

Here we provide the main functions used to simulate the planar XY, obtain the correlation function and detect vortexes in the lattice.

```python
import numpy as np
import matplotlib.pyplot as plt
from numba import njit
from scipy.optimize import curve_fit
```

```python
@njit
def mc_sweep(spin_lat, energy, L, J, T):
    '''
    Performs a single Monte-Carlo sweep over the lattice, consisting of N trial changes to random spins.
    '''

    N = L**2      # Defining the number of spins, assuming a square lattice.

    for x in range(N):    # Doing N trial changes, so on average every spin is flipped once in a sweep.
        i = int(L*np.random.uniform(0,1))    # Choosing a random spin
        j = int(L*np.random.uniform(0,1))
        old_spin = spin_lat[i][j]
        a = np.cos(old_spin - spin_lat[(i-1)%L][j])     # Calculating the initial energy for this site, by getting the contribution from each nearest neighbour
        b = np.cos(old_spin - spin_lat[(i+1)%L][j])     # Nearest neighbours are calculated with boundary conditions.
        c = np.cos(old_spin - spin_lat[i][(j-1)%L])
        d = np.cos(old_spin - spin_lat[i][(j+1)%L])
        E_old = -J*(a+b+c+d)

        test_spin = (old_spin + np.random.uniform(0,1)*2*np.pi)%(2*np.pi)     # adding a random angle between 0 and 2*pi
        # test_spin = np.random.uniform(0,1)*2*np.pi

        a = np.cos(test_spin - spin_lat[(i-1)%L][j])     # Calculating the new energy with the trial change.
        b = np.cos(test_spin - spin_lat[(i+1)%L][j])
        c = np.cos(test_spin - spin_lat[i][(j-1)%L])
        d = np.cos(test_spin - spin_lat[i][(j+1)%L])
        E_trial = -J*(a+b+c+d)
        dE = E_trial - E_old      # Calculating the change in energy


        if dE<=0:       # if energy is reduced, automatically accept.
            spin_lat[i][j] = test_spin      # Change energy and
```

```
                spin
            energy += dE
        else:       # if energy is increased, accept with Boltzmann weighted probability
            r = np.random.uniform(0,1)
            w = np.exp(-dE/T)
            if r<=w:
                spin_lat[i][j] = test_spin     # Change energy and spin
                energy += dE

    return spin_lat, energy
```

```
@njit
def mc_run(L=32, J=1, T=3, N_runs=100):
    '''
    Simulates an XY model of spins on a lattice using many Monte-Carlo sweeps, at a given temperature.
    '''
    E = np.zeros(N_runs)    # Initialize energy array
    M = np.zeros((N_runs,2), dtype=np.float64)    # Initialize magnetization array

    # Setup initial lattice
    spin_lat_0 = np.ones((L, L))
    for i in range(L):
        for j in range(L):
            spin_lat_0[i][j] *= np.random.uniform(0,1)*2*np.pi        # Choose random spins for each site.

    E_0 = 0
    M_0 = np.array([0,0], dtype=np.float64)
    for i in range(L):
        for j in range(L):
            a = np.cos(spin_lat_0[i][j] - spin_lat_0[(i-1)%L][j])        # Calculate contribution from each nearest neighbour
            b = np.cos(spin_lat_0[i][j] - spin_lat_0[(i+1)%L][j])
            c = np.cos(spin_lat_0[i][j] - spin_lat_0[i][(j-1)%L])
            d = np.cos(spin_lat_0[i][j] - spin_lat_0[i][(j+1)%L])
            E_0 += -J*(a+b+c+d)/2     # Calculate initial energy
            M_0[0] += np.cos(spin_lat_0[i][j])      # Calculate
```

```
               initial magnetization in each direction.
            M_0[1] += np.sin(spin_lat_0[i][j])


    spin_lat = np.copy(spin_lat_0)    # Create copy so that
       initial array is preserved.
    E[0] = E_0
    M[0] = M_0

    # Start monte-carlo sweeps
    for k in range(1,N_runs):
        spin_lat, E[k] = mc_sweep(spin_lat=spin_lat, energy=E[k
           -1], L=L, J=J, T=T)      # Perform Monte-Carlo sweep,
           add energy to the energy array
        M[k,0] = np.sum(np.cos(spin_lat))      # Calculate and
           append magnetization values.
        M[k,1] = np.sum(np.sin(spin_lat))

    return spin_lat_0, spin_lat, E, M    # return initial array,
       final array, energy and magnetization
```

```
@njit
def id_vortex(spins):
    '''
    Detects vortices in a lattice of spins.
    '''

    x,y = np.shape(spins)    # Getting the dimensions of the
       array.
    L = x
    vortices = np.array([[0,0],[0,0]], dtype=np.float64)      #
       Initialize coordinates array for vortices. Two set of '
       junk' coordinates are added to correctly set the
       dimnesions of the array.
    antivortices = np.array([[L-1,L-1],[L-1,L-1]], dtype=np.
       float64)      # Similarly initialize coordinates array for
        anti-vortices.

    for i in range(x):     # Iterate over each spin in the
       lattice
        for j in range(y):

            # Go over the plaquette of spins associated with it, and calculate each difference in angle (in anti-clockwise order)
            a = (spins[i][(j+1)%L] - spins[i][j])
            b = (spins[(i+1)%L][(j+1)%L] - spins[i][(j+1)%L])
            c = (spins[(i+1)%L][j] - spins[(i+1)%L][(j+1)%L])
            d = (spins[i][j] - spins[(i+1)%L][j])

            product = a*b*c*d      # Get product of differences
            sum_ = abs(a % np.pi)+abs(b % np.pi)+abs(c % np.pi)+abs(d % np.pi)      # Get sum of differences

            if (product<0 and sum_==2*np.pi):  # Vortex found
                arr = np.array([a,b,c,d])
                arr.sort()    # Sort the array of differences into ascending order, to check vortex vs. antivortex.

                if arr[1]>0:    # Vortices will have 1 negative difference, so the second element should be positive.
                    coord = np.array([[j+0.5,i+0.5]])
                    vortices = np.append(vortices, coord, axis=0)    # Add coordinates of vortex to array
                elif arr[1]<0:    # Anti-vortices will have 1 positive difference, so the second element should be negative.
                    coord = np.array([[j+0.5,i+0.5]])
                    antivortices = np.append(antivortices, coord, axis=0)      # Add coordinates of antivortex to array

    nv = len(vortices) - 2    # Number of vortices (subtracting two 'junk' entries)
    nav = len(antivortices) - 2    # Number of antivortices (subtracting two 'junk' entries)
    return vortices[2:], antivortices[2:], nv, nav     # return coordinate arrays (without junk entries) and numbers of each kind.
```

```
@njit
def pbc_distance(rij, L):
    '''
    Get the correct distance between two particles, accounting for periodic boundary conditions.
    '''

    #input rij is the separation between the particles calculated without boundary conditions

    #checking if the separation rij violates the boundary conditions
    #and evaluating the true distance whenever this is the case

    if abs(rij[0]) > 0.5*L:
        rij[0] = rij[0] - L*(rij[0]/abs(rij[0]))
    if abs(rij[1]) > 0.5*L:
        rij[1] = rij[1] - L*(rij[1]/abs(rij[1]))

    return rij

@njit
def get_correlation(spins, mag_per_spin, bins=10):
    '''
    Gives the spin-spin correlation function for a given spin lattice.
    '''

    #obtaining the dimensions of the lattice
    L = np.shape(spins)[0]

    #defining the distance bins using the input for number of bins
    r = np.linspace(1, L/np.sqrt(2), bins)

    #empty array to store the number of spins which fit into any one distance bin
    nr = np.zeros_like(r)

    #empty array to store sum of the correlation values for each bin before calculating the mean
    Cr = np.zeros_like(r)
    dx = L/np.sqrt(2)/bins
```

```python

    #looping over all pairs of spins in the lattice
    #the first spin is indexed by [i,j] and the second spin is indexed by [p,q]
    for i in range(L):
        for j in range(L):
            for p in range(L):
                for q in range(L):

                    #continuing the loop only for pairs of distinct spins
                    if i != p or j != q:
                        #defining the difference between the 'coordinates' of the spin
                        x = p-i
                        y = q-j
                        #storing these in an array and finding the distance using periodic boundary conditions
                        rij = pbc_distance(np.array([x,y]), L)
                        dist = np.sqrt(rij[0]**2 + rij[1]**2)
                        #computing the correlation for this pair of spins
                        corr =  np.cos(spins[i][j]-spins[p][q])-mag_per_spin**2

                        #assigning this correlation to a distance bin
                        rbin = int(dist/dx)
                        nr[rbin] += 1
                        Cr[rbin] += corr

    #setting zero elements to 1 in order to prevent `nan` values in the output
    nr[nr<1] = 1

    #returning the average value of c(r) for each bin and the distance bins
    return Cr/nr, r

@njit
```

```python
def corr_avg(spins, energy, T,J=1, sweeps=10_000, bins=10):
    '''
    Gives the spin-spin correlation function averaged over many runs for a given spin lattice.
    The correlation is calculated after every 100 Monte-Carlo sweeps and its value is averaged across all runs to find the correlation.
    '''
    #defining an empty array to store value of correlation
    cr = np.ones((sweeps//100, bins))
    L = np.shape(spins)[0]                              #defining the shape of the lattice
    N = L**2                                            #number of spins in the lattice

    #looping over all monte carlo sweeps
    for mc in range(sweeps):

      #allowing the system to 'heat up' over 10_000 Monte-Carlo sweeps
        spins, energy = mc_sweep(spins, energy, L, J, T=T)

        #evaluating the correlation function every 100 steos
        if mc%100==0:
          #calculating average magnetization per spin for the lattice
            m = np.sqrt(np.sum(np.cos(spins))**2 + np.sum(np.sin(spins))**2)/N
            #evaluating the correlation for the given pair of spins
            cr[mc//100], bins_array = get_correlation(spins, m, bins=bins)

    #defining an empty array to store the values of correlation
    correlation = np.ones(bins)

    #looping over all distance bins
    for i in range(bins):
      #storing the correlation for each bin averaged across all monte-carlo sweeps
        correlation[i] = np.mean(cr[:, i])

    #returning the averaged correlations and associated distance bins
    return correlation, bins_array
```