Computational Final Project

Instructor: Dr. Frank Golf

Hayden Swanson

The University of Nebraska-Lincoln
Department of Physics
May 2023

1 Introduction

This computational project explores and implements a relatively new Monte Carlo algorithm called the Wang Landau algorithm (2008) for the Ising Model. The Ising model is a simple and well-understood model for testing and comparing new Monte Carlo algorithms in Condensed Matter Physics. For roughly 50 years the main driver in simulational physics has been the Metropolis Hasting algorithm which was developed in Los Alamos during the Second World War. The algorithm has known limitations for specific problems so new and more efficient algorithms, like the Wang Landau algorithm, were created to simulate more complicated physics problems.

The layout of this paper will introduce the 2D Ising model and the thermodynamics used in the project. With this simple model in mind, I will explain the Metropolis-Hastings algorithm and how it lends itself to the Ising Model. The Metropolis-Hastings algorithm requires some prerequisite knowledge of Monte Carlo methods and Markov chains which I will give a general explanation as needed. Wang and Landau implement new ideas building off this knowledge which will be discussed in detail. Finally, I will show the results of the simulation using the Wang Landau algorithm.

Even though this paper is in the context of Condensed Matter Physics the ideas used here are prevalent in a large number of areas even outside of physics. The applications of Monte Carlo techniques range from ray tracing in computer graphics to search and rescue strategies for the Coast Guard, risk assessment in businesses, and much more. Specifically, the Wang Landau algorithm is useful for super large systems with complex energy landscapes (will make sense why later!). For example, it is used to model complex proteins [6] and has been used in industry to optimize the distribution of auto parts.

2 Ising Model

The Ising Model is one of the simplest models of a solid to show a phase transition. It was a problem given to Ising by his research advisor Wilhelm Lenz in 1925. Interestingly, Ising is one of the most commonly mentioned names in names in theoretical physics literature. The model can be described as a lattice of points and the lattice for simplicity will just be a square grid of evenly spaced points. Each point has an associated spin σ_i of up or down (+1 or -1) where i denotes the location of spin. The model assumes that only the nearest neighbors of spins interact. It is important to note that the finite lattice in our case has periodic boundary conditions, so the first spin on the top left edge of a lattice ”feels” the spin of the square lattice’s bottom left and top right. The lattice can also be thought of as a toroid to satisfy the boundary conditions. In our case, neighboring spins that point in the same direction have a lower energy than if they were anti-aligned so the entire lattice configuration X has an energy H given by,

$$H(X) = -\sum_{<i,j>}^{N} \sigma_i \sigma_j, \tag{1}$$

where $< i, j >$ is shorthand for unique nearest neighbor spin pairs. This sum calculates the energy of a lattice of random up or down spins. The negative sign means the spins want to be aligned with one another (lower energy) and is therefore defined as ferromagnetic. The temperature dependence enters the model due to the fact that the spins can be randomly flipped due to temperature. We can write the partition function for the system as,

$$Z = \sum_{i=0}^{N} e^{\frac{-E_i}{k_B T}} = \sum_{E} g(E) e^{\frac{-E}{k_B T}}. \tag{2}$$

Here k_B is the Boltzmann constant, T is the temperature of the system and E_i is the energy of the microstate E_i. In the case of the Ising model, a microstate is just one particular configuration of the lattice's spin states. But, the first sum can be rewritten as the sum on the right where $g(E)$ is the *density of states.* The density of states is simply how many configurations have the same energy. For example, the minimum energy where all the spins are the same as a density of states of 2, all spin up or all spin down. The sum is over the entire energy spectrum (all possible energy configurations of the spins).

The derivation of the partition function can be found on Wikipedia but it can be understood in the context of probability. The probability for a particular spin configuration is simply,

$$p_i = \frac{e^{\frac{-E_i}{k_B T}}}{Z}. \tag{3}$$

This obviously satisfies $\sum_{i=0} p_i = 1$ so it makes sense as a probability. An important observation of the Wang Landau algorithm is that we can calculate all the averages for any thermodynamic quantity at any temperature if the density of states is calculated. I will show now just a few. The Gibbs Free energy F,

$$F = -k_B T ln(Z). \tag{4}$$

The internal energy $U(T) = \langle E \rangle_T$,

$$\langle E \rangle_T = \frac{\sum_E E g(E) e^{\frac{-E}{k_B T}}}{Z}. \tag{5}$$

The entropy,

$$S = \frac{U}{T} + k_B ln(Z). \tag{6}$$

The specific heat,

$$C(T) = \frac{\langle E^2 \rangle_T - \langle E \rangle_T^2}{T^2}. \tag{7}$$

We now have all the background to start discussing a Monte Carlo simulation of the Ising model. The Ising model is very rich and a lot more can be done with it (e.g. one could add a magnetic field, different boundary conditions, etc...).

2.1 Analytical Solution*

The analytical solution developed by Onsager in 1943 for the 2D Ising Model (no external magnetic field) took about 15 pages of dense complicated mathematics of groups. Later people simplified the result but is still quite dense. The most compact solution that yields the partition function I could find is below and contains an integral,

$$Z = \lambda^N, \tag{8}$$

$$ln\lambda = ln(2cosh(2/T)) + \frac{1}{\pi}\int_0^{\pi/2} ln\left[\frac{1}{2}(1 + (1 - K^2 sin^2(w))^{\frac{1}{2}}\right] dw, \tag{9}$$

$$K = \frac{2sinh(2/T)}{(cosh(2/T))^2}. \tag{10}$$

The integral is not analytical but can be solved numerically. So, technically the solution is not an analytical one but the errors from the numerical methods can be made to be much smaller than the ones from the Wang and Landau algorithm. This is discussed more in detail in the results section.

3 The Wang and Landau Model

As said before, the Wang and Landau algorithm is a type of Monte Carlo algorithm and can be simply described as a ”random walk in energy space with a flat histogram” [4]. Even though it uses more complicated and sophisticated Monte Carlo methods it has the advantage of being very simple to implement so I will first compare it to the Metropolis-Hastings algorithm to build up the Monte Carlo methods and ideas that can be compared to and motivate parts of the Wand and Landau algorithm. Moreover, the Metropolis-Hastings algorithm is an excellent introduction to Monte Carlo methods.

3.1 Metropolis Hasting Algorithm

All modern Monte Carlo methods have roots in the Metropolis-Hastings algorithm [4]. It relies on random number sampling to perform powerful calculations. In general, it draws random samples from a probability distribution where direct sampling is difficult or impossible. In the case of the Ising model, it depends on Z which is typically very difficult to calculate. By using a Markov Chain (each sequence of sample draws depends on *only* on the previous sample drawn), it can sidestep this difficulty (seen by the algorithm steps below, Z gets divided out). Moreover, the Metropolis-Hastings algorithm draws samples from a probability density function that is unknown ($e^{\frac{-E_i}{k_B T}}/Z$). As long as a function proportional to it is known ($e^{\frac{-E_i}{k_B T}}$), one can make a histogram of the randomly drawn samples. And if, for example, the histogram has the shape of a desired function an integral can be calculated (so it is also useful for calculating integrals).

Precisely, the algorithm can be explained in the following steps for calculating an integral, and then I will show similar steps for the Ising Model. Define the integrand $f(x)$ as $f(x) = p(x)g(x)$ where $p(x)$ is a probability density function that we are trying to draw from using the Metropolis-Hastings Algorithm and $g(x)$ is some known function. The product of the two should yield $f(x)$. An important feature of the algorithm is that $p(x)$ does not need to be normalized. This is what makes the Metropolis-Hastings algorithm so useful because that normalization factor could be very difficult to calculate. For clarity in the acceptance step (3), I define the normalized $p(x)$ to be $P(x)$.

1. Choose an initial x-value to start the algorithm

- The only requirement is that it has to be within your integration bounds.

2. Make a finite step in a random x direction
    - Here the only requirement is that the random direction should be picked such that it is equally likely to decrease or increase x. Otherwise, the algorithm needs a slight modification that is not considered here.
    - This is a Markov process so this finite step should depend on the current x value. For example, $x_{proposed} = x_{current} + step * U[-1, 1]$. Where *step* is some small step size and U is a random number between -1 and 1 drawn from a uniform distribution.
3. Calculate the acceptance ratio γ
    - The following acceptance ratio is a critical component of the algorithm. It tells you how probable the proposed x value, from 3, is compared to the current x value according to $P(x)$.
    - $\gamma = \frac{P(x_{proposed})}{P(x_{current})} = \frac{p(x_{proposed})}{p(x_{current})}$
4. Choose a random number between 0 and 1
    - This should be drawn from a uniform distribution.
5. Accept or Reject $x_{proposed}$
    - If $r < min(\gamma, 1)$ then accept $x_{proposed}$. Otherwise, reject it.
    - the *min* just ensures the probability is not greater than 1
6. Go back to 2
    - Do this some large amount of times to get enough draws (accepted x values) from P

To calculate the integral we now perform a Monte Carlo integration. The result that I will not derive because it is not relevant to the project as a whole is,

$$Integral = \frac{1}{N}\sum_{i=0}^{N}\frac{f(X)}{P(X)} = \frac{1}{N}\sum_{i=0}^{N} g(X). \tag{11}$$

Here X refers to randomly sampled values according to $P(X)$ which was obtained from the Metropolis-Hastings algorithm.

The algorithm in the context of the Ising Model is the following [4]:

1. Choose an initial lattice configuration
2. Pick a spin site i to flip
3. Calculate the change in energy ΔE of the lattice by flipping spin at site i
4. Generate a random number r using U[-1,1]

5. If $r < min(e^{\frac{-\Delta E}{k_B T}}, 1)$, accept the proposed energy
6. Go to 2

Depending on the probability density function that is being sampled it might have lots of bumps with maximums. The algorithm tends to get trapped around these maxims which can be a disadvantage. This is because of how the algorithm accepts and rejects x. If $P(x_{proposed}) > P(x_{current})$ then it will always be accepted and if it's less than 1 then there is a chance of rejection. Therefore, it is sampling more near maximums of $P(x)$ and has been shown to get stuck in these maximums. Depending on the problem this can be problematic and the Wang Landau algorithm doesn't suffer from this issue (the general group of sampling techniques that gets around this issue are called Multicanonical Sampling Methods).

3.2 The Wang and Landau Algorithm

Examples of systems with multiple maxima (bumpy energy landscape) that are widely spaced are the spin glass model, Potts model, and systems near phase transitions that exhibit a discontinuity in the first derivative of the free energy with respect to a thermodynamic variable.

The goal of Wang and Landau is to calculate the density of states (independent of temperature!) and thus the partition function. Here is how the algorithm works:

1. Set the density of states g(E) for the entire spectrum of energies to be 1 and a modification factor $f = e$
    - When performing calculations the numbers will get way too big and cause overflow. For practical purposes, one should use the natural logarithm of g(E). Then you set all $ln(g(E)) = 0$ and $ln(f) = 1$
    - Note, it is very important the spectrum of the possible energies is calculated correctly.
2. Choose an initial lattice configuration
3. Pick a random site i to flip
4. Calculate the acceptance probability ratio γ
    - $\gamma = \frac{g(E_{current})}{g(E_{proposed})}$
    - When working in terms of the natural log: $\gamma = e^{ln(g(E_{current}))-ln(g(E_{proposed}))}$
5. Generate a random number r using U[0,1]
6. If $r < min(\gamma, 1)$ accept the spin-flip and the associated proposed energy, otherwise reject the spin-flip
7. Set $g(E_{accepted}) = g(E_{accepted}) * f$
    - If working in natural logarithms, $ln(g(E_{accepted})) = ln(g(E_{accepted})) + f$
8. If the histogram is not flat go back to 3

- For each accepted energy on the spectrum it should be counted. This will build a flat histogram. If the minimum value of the histogram is greater than 0.8$*$maximum value of the histogram consider it flat

9. If the histogram is flat, decrease f by taking its square root and reset all the histogram values to 0 (but keep the density of states!) and go back to 3
   - If working in natural logarithms, divide $ln(f)$ by 2
10. Repeat steps 3 through 9 until $f = e^{10^{-8}}$
    - If working in natural logarithms, $ln(f) = 10^{-8}$
    - Now you have the density of states to use in calculations!

It is not intuitive how the algorithm works from the above recipe. So now, I will give some motivation for why the algorithm works. However, my complete understanding is not solid. So, I will briefly explain the main idea given in the original paper. I have added an appendix for a more in-depth treatment with sources. I have not included it here because of some concerns I have and is thus not completely rigorous.

In the original paper he writes, "Our algorithm is based on the observation that if we perform a random walk in energy space with a probability proportional to the reciprocal of the density of states $1/g(E)$, then a flat histogram is generated for the energy distribution. This is accomplished by modifying the estimated density of states in a systematic way to produce a “flat” histogram over the allowed range of energy and simultaneously making the density of states converge to the true value.” A potential rigorous reason the probability is constant is given in the appendix. Each time a state is accepted the accepted state (proposed or current) is multiplied by the f factor which is e to start out and is decreased systematically afterward. The nature of how the random walk moves in the energy space is shown by Landau in an animation [3] at the 23:02 mark. It is important to note that the probability $1/g(E)$ changes in each step.

4 Results

In this section, I will show the results for the Wang and Landau algorithm that I wrote. Specifically, I will showcase graphs for the Gibb's (he calls it the Gibb's free energy but others say it is the Helmholtz Free energy, anyways, I use equation (4)) Free energy and Entropy for different lattice sizes which are more difficult to calculate with the Metropolis-Hastings algorithm and most other Monte Carlo methods but for Wang and Landau it is trivial. I will also show the relative errors from the exact solution for the Free Energy. To easily showcase a phase transition of the 2D Ising Model I will present the Specific Heat as well.

4.1 Optimization and Density of States

The original algorithm described above is quite slow so I implemented two ideas that greatly increases the speed of the algorithm. My first idea was to somehow speed up the calculation of the entire energy of the lattice. The original method had two for loops to go over all rows and columns of the lattice. This meant the computer had to potentially compute the energy over 30 million times for a 10 by 10 square lattice of spins. However, to flip one lattice point you don't need to calculate the energy of the lattice just the change in energy by flipping one spin. This requires no for loop just some basic arithmetic calculations so the efficiency of the algorithm was improved.

The second idea came from reading the original paper on the algorithm [5]. Since the density of states is symmetric you only need to calculate the density of states for half of the energy spectrum for the 2D Ising Model. The only caveat is that the Wang and Landau algorithm suffers "large" errors near boundaries. So, by speeding up the calculation you lose some precision in the calculation. This error can be mitigated quite effectively if you extend your energy spectrum to be a little more than half the distribution so the edge effect is thrown away when you reflect the density of states. In Figure 1, the left plot shows the slow unoptimized Wang and Landau algorithm next to the optimized one without considering the edge effects. After considering the edge effects, the results of a faster Wang Landau algorithm are shown in Figure 2 without that bump shown above on the right. The "burn in" of Energies used was 10% the minimum energy value for the lattice size (probably way more than what is actually needed).

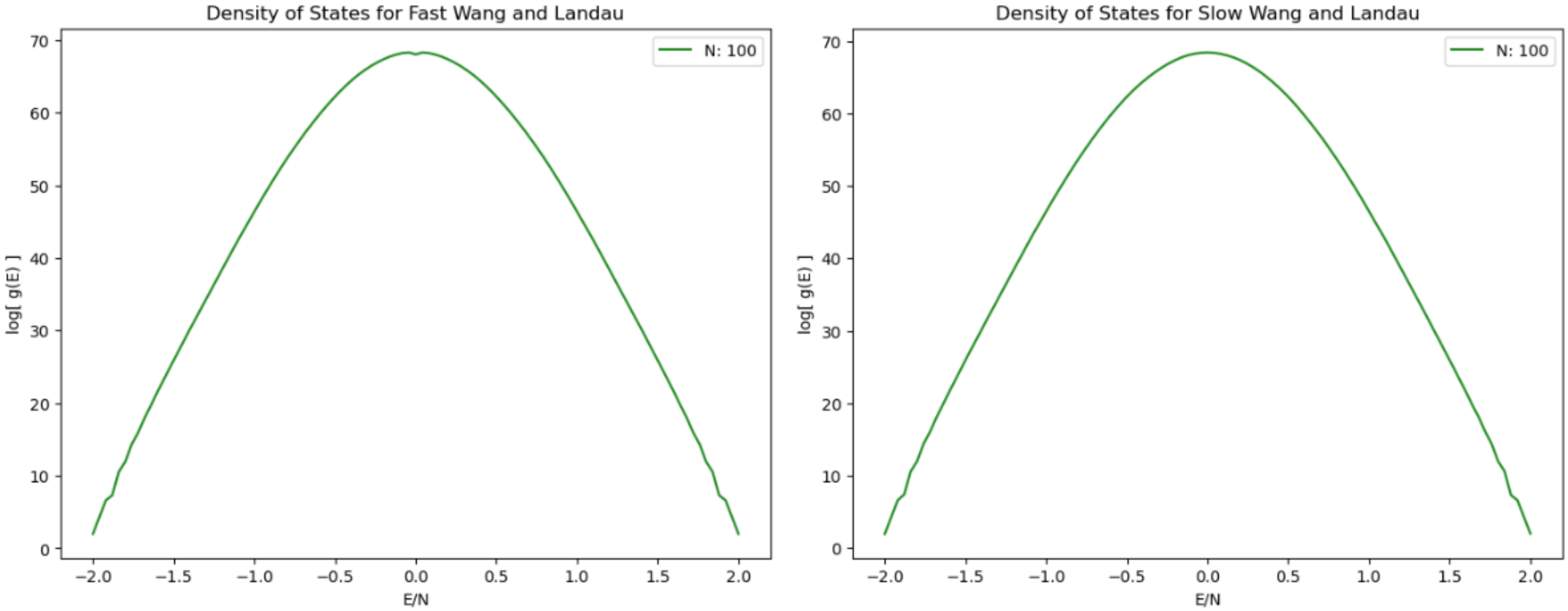


Figure 1. Left, shows the optimized Wang and Landau algorithm with boundary errors causing the dip in the density of states at E/N = 0. Right, shows the original good but slower algorithm for the density of states.

It is worth noting that there is still a dramatic optimization improvement that could be done that was not considered here. It is possible to divide up the spectrum of energies (with an overlap to avoid edge effects!) and calculate the density of states for all those concurrently.

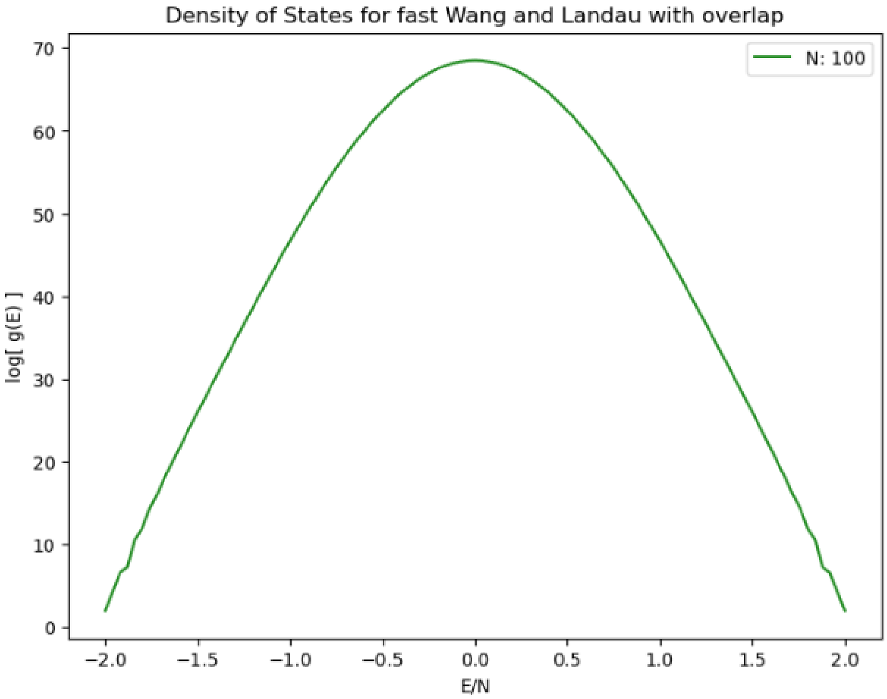


Figure 2. The fast Wang and Landau algorithm with no bad edge effects at E/N = 0. Shows qualitatively the fast optimization is implemented without major loss in precision.

4.2 Gibb's Free Energy and Comparing to the Analytical Solution

Using the density of states we can easily calculate F and S from (4) and (6) respectively. As mentioned before these are difficult to calculate with typical algorithms but more importantly, the analytical solution for these Thermodynamic variables is easier to calculate than the others. The exact solution, from [8], was calculated to compare the Free Energies and calculate the relative error,

$$\epsilon = |(F_{sim} - F_{exact})/F_{exact}|. \tag{12}$$

Where F_{sim} and F_{exact} is the Free energy calculated by the Wang and Landau algorithm and the Free Energy calculated analytically. It is important to point out that the exact solution had a difficult integral that needed to be evaluated. To do so I evaluated it using Monte Carlo methods and propagated the uncertainty,

$$\sigma(Integral) = \frac{\sigma(f(X))}{\sqrt{n_{toy}}}, \tag{13}$$

where $f(X)$ is the integrand evaluated by all the random numbers drawn from a uniform distribution n_{toy} times and $\sigma(f(X))$ is the standard deviation of all those points. The error was propagated appropriately using the Python uncertainties package [1]. Figure 3 shows the Free Energy and the relative errors on the right for a 16 by 16 lattice.

The maximum uncertainty of the relative error, with $n_{toys} = 10^7$ for the integral, was propagated through to be $2.23 \cdot 10^{-5}$ which is much smaller than the relative error between the algorithm and the ”analytical” solution. Thus, it is justified to just treat the ”analytical” as the analytical solution. To conclude, the Wang and Landau algorithm is accurate to within about 0.0035 of the true value of the Gibbs Free Energy of the 2D Ising Model.

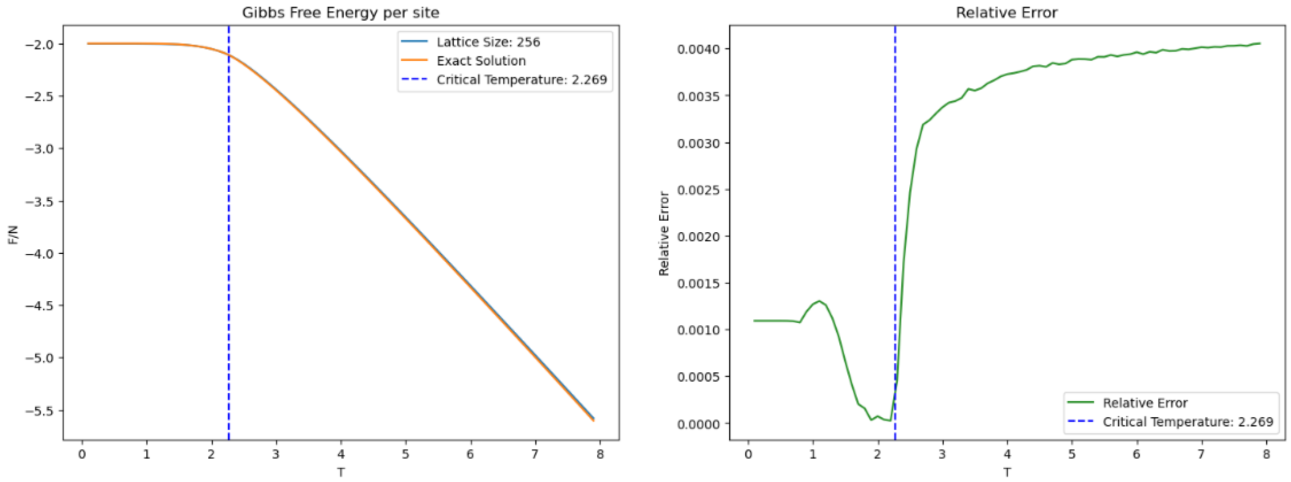
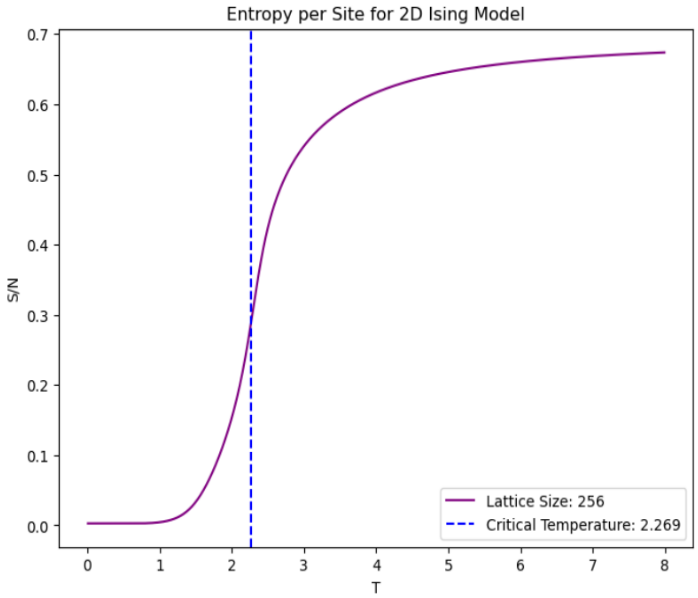


Figure 3. The left plot shows the Gibbs Free energy per site for a 16 by 16 lattice with the exact solution. The algorithm took 130 million sweeps to generate the density of states. The right plot shows the relative error between the “analytical” and the Wang and Landau algorithm.

4.3 Specific Heat Capacity, Entropy, and Phase Transitions

Now that it is shown that the Wang and Landau algorithm is producing the correct results we can explore some physical properties of the system. A phase transition is fairly intuitive since we observe, for example, liquid water boiling and becoming a gas or freezing and becoming a solid.



A general rigorous definition in terms of thermodynamics isn’t the most straightforward but a huge effort of computational statistical mechanics is to characterize and find the so-called ”critical points” where phase transitions occur. Understanding phase transitions has been an effort of science for well over 100 years and the Ising Model contributed greatly to our understanding of them.

The critical temperature T_C for the 2D Ising model has been found to be $2/ln(1+\sqrt{2}) \approx 2.269$ [8]. Using this can study what the critical temperature is referring to in terms of quantities like specific heat and entropy.

For the 2D Ising Model, the entropy shows quite clearly the phase transition. At colder temperatures, it starts off in a low entropy (high order) regime, and then around the critical temperature the entropy quickly changes. Then it tails off to a state of high entropy (low order).

One of the first rigorous definitions of a phase transition was offered by Ehrenfest. Roughly speaking it was about looking for nonanalycities or discontinuities in the different orders of the derivative of the Free energy [7]. It can be shown the that specific heat C_V is,

$$C_V = -T \left(\frac{\partial^2 F}{\partial T^2} \right)_V . \tag{14}$$

Now if we look at C_V and see any discontinuities then we can say the system underwent a phase transition. Figure 4 shows the specific heat for a number of lattice sizes (really small 4x4, 8x8, 16x16). It might be tempting to call the peaks discontinuities and call it a day. By eye, it doesn't seem to be a discontinuity, especially for the smaller lattices it is just a bump. In fact, the 1D Ising model has a similar peak for the specific heat (for the smaller 4 by 4 square lattice) but it is well known that it shows no phase transition. I believe the behavior of Figure 4 is explained by looking at the lattice sizes. As the lattice size gets large the peak approaches the critical temperature. In fact, Ehrenfest definitions might fail for small finite systems and better definitions for phase transitions have been developed [7] (not sure how boundary conditions play a role here).

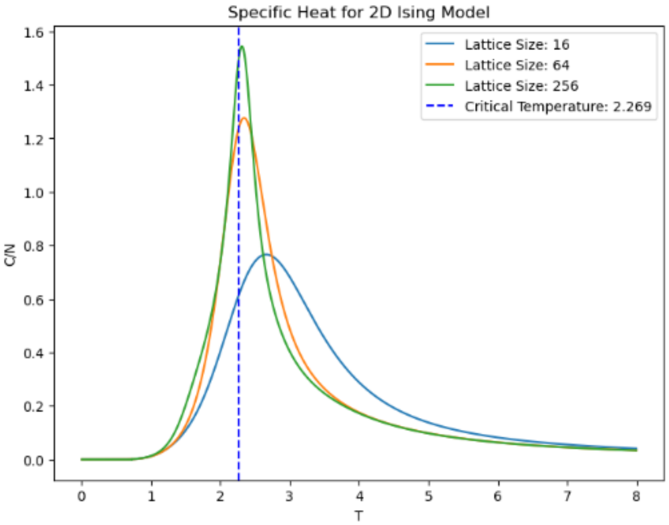


Figure 4. Specific Heat per site for various 4 by 4, 8 by 8 and 16 by 16 grid of spins.

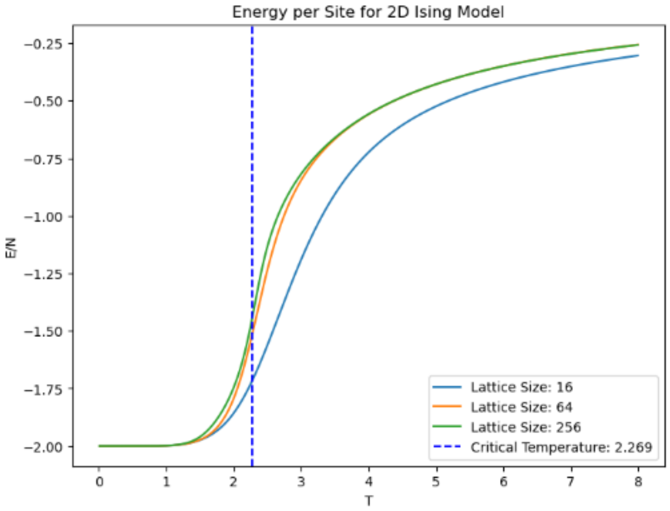


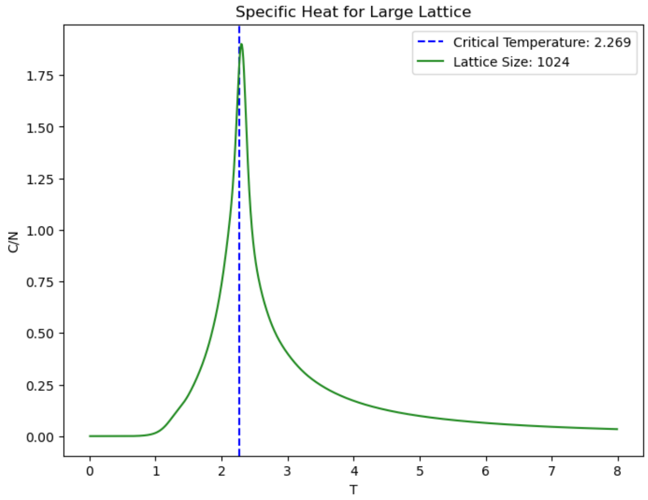
Figure 5. Average Energy per site for various 4 by 4, 8 by 8 and 16 by 16 grid of spins.

Since the definitions of a phase transition might be more pronounced for larger systems, I ran the algorithm overnight with over 1 billion sweeps for the 32 by 32 lattice. If the peak is more pronounced then my hypothesis of lattice size playing a role in the severity of the peak is true.

To be thorough, the treatment for these larger lattices needed special care due to the incredibly large numbers that we are working with. The average energy to calculate the specific heat had to be modified in the following way,

$$\langle E \rangle = \frac{\sum_E g(E) E e^{-(E+E_{max})/T}}{\sum_E g(E) e^{-(E+E_{max})/T}} . \tag{15}$$

Where E_{max} is the maximum energy possible for the lattice (alternating spins). This avoids taking exponentials of the order of positive thousands (e to a large negative value is ok because it approaches 0, but a large positive value just races to infinity and reaches the overflow for floats in python). Using this updated definition (which is completely equivalent) we produce the plot, For this 32x32 lattice we can see the peak is becoming more pronounced. This means we are approaching a second-order phase transition since this sharp peak indicates a discontinuity in the second order of the Free Energy. It is also known that the 2D Ising model does have a second-order phase transition but not a first by Ehrenfest's definition.



4.3.1 A Concern of the Algorithm

This sub-sub-section explores a potential bug I found when generating this large lattice. In the original paper by Wang and Landau [5], they show the density of states for the 32x32 lattice. I noticed that my plot for the density of states was quite different so it raised a lot of alarms and had to do a bunch of checks on my code. Here is the figure comparing the two,

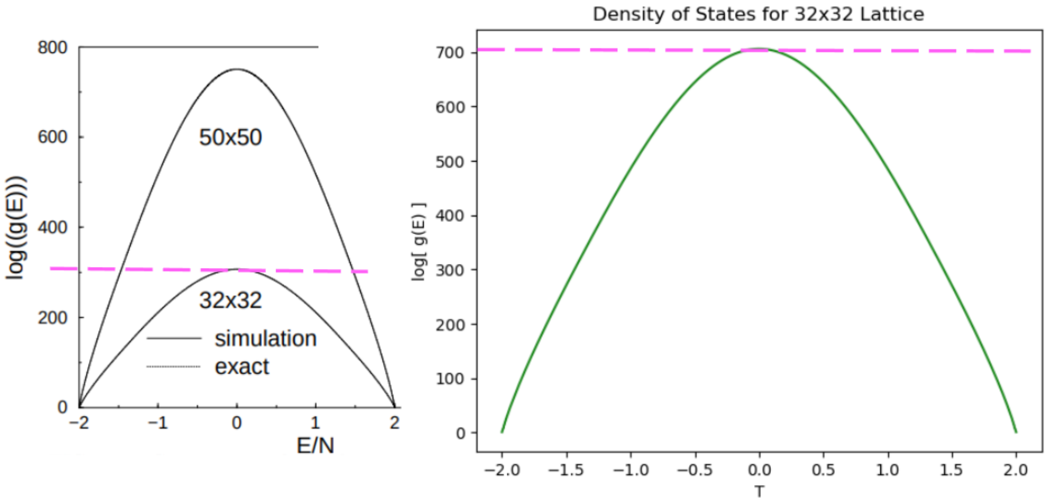


Figure 6. On the left is the Wang and Landau plot for the density of states. On the right is my plot.

A convincing check is to see if the total number states is 2^{N^2}. Since $N = 32$, we shall work in logarithms so we compare the natural logarithm of the sum of the density of states to $N^2 ln(2)$. Doing so I obtain 709.78 from the algorithm and the analytical is 708.93. It seems like my algorithm is working correctly in this regard and it fits to the analytical solution well for the Free Energy. I could not figure out the discrepancy between our calculations. The next step would be to program the analytical solution for the density of states and see if it compares.

5 Conclusions

In conclusion, the Wang Landau algorithm offers a fast and easily implemented algorithm to calculate the density of states. This is useful for systems that have "rough energy landscapes" and performs much better than a Metropolis-Hastings type algorithm for these types of systems.

I have implemented this algorithm as well as a couple of ways to optimize the algorithm. The next step in terms of optimization would be to calculate the density of states over a small range of overlapping energy spectrums and stitch them all together.

I tested the algorithm on the famous 2D Ising model which serves as a "fruit fly", as David Landau says, for most Monte Carlo algorithms in Condensed Matter. Then the phase transitions of the lattice were explored for the model and how lattice size might contribute.

6 Appendix: Why Does the Wang and Landau Algorithm Work?

A fundamental requirement of a Markov Chain Monte Carlo Method is that it should satisfy "the detailed balance equation." This is an important equation in Monte Carlo methods and in physics (actually where it originated!) because it is a statement about equilibrium. Before I write down the detailed balance equation for the algorithm I will show why a flat histogram is produced.

Suppose we have a configuration of our lattice X and X might share the energies with many other configurations. Wang and Landau choose the probability $P(X)$ to be in state X to be proportional to the inverse of the density of states [2],

$$P(X) \propto \frac{1}{g(H(X))}. \tag{16}$$

Now the number of lattice configurations X that share the same energy E_i is the density of states $g(E_i)$. Thus the probability $P(E)$ that a certain lattice configuration X has a certain energy E_i is,

$$P(E) = P(X)g(E_i) = constant \tag{17}$$

This information was given by [2]. So they observed that if a random walk in energy space is performed with a probability $\frac{1}{g(E)}$ then a flat histogram is generated (which is what I showed above with (13)) (This is information in the sentence was given by [5]). An important idea (I think) is that we are equally likely to visit any energy value (and generate a flat histogram) but whether we accept the change (dictated by the acceptance criteria) is what builds our density of states. The probability we transition from one energy $E_{current}$ state to the next $E_{proposed}$ is given by (following Landau's book and his paper to piece the transition rate/probability [4] [5]),

$$W_{E_{current} \to E_{proposed}} = \frac{P(X_{proposed})}{P(X_{current})} = min(\frac{g(E_{current})}{g(E_{proposed})}, 1). \tag{18}$$

This is the origin of the acceptance ratio in the algorithm (the min makes sure the probability doesn't go over 1). For the Metropolis Hastings algorithm $W = e^{-\Delta E/k_B T}$ with P given by (3). The requirement for the transition probability and $P(X)$ is that it agrees with the detailed balance equation. The detailed balance equation is (from [4]),

$$P(X_{current})W_{E_{current} \to E_{proposed}} = P(X_{proposed})W_{E_{proposed} \to E_{current}}. \tag{19}$$

The detailed balance equation is a statement for Markov Chains (where the probability of where you move next only depends on where you are now) so moving in between the current and proposed state. This is where I have confusion because if you plug in W it can be reduced that $P(X_{proposed}) = P(X_{current})$. I don't think that statement is generally true for the algorithm or any Metropolis Hasting algorithm.

References

[1] Eric O. Lebigot. Uncertainties python package. URL `https://pythonhosted.org/uncertainties/`.

[2] J. P. Felipe Moreno, Sergio Davis. A portable and flexible implementation of the wang–landau algorithm in order to determine the density of states. *Cornell University*, 2021. doi: https://doi.org/10.1016/j.cpc.2022.108283.

[3] D. P. Landau. Tutorial 1 - replica-exchange wang-landau sampling - lecture 1 - david landau. URL `https://www.youtube.com/watch?v=_ilTEy_w8h8&t=1818s`.

[4] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, 2014.

[5] D. P. Landau and F. Wang. An efficient, multiple range random walk algorithm to calculate the density of states. 2008. doi: https://doi.org/10.1103%2FPhysRevLett.86.2050.

[6] T. Ojeda-May and M. E. Garcia. Electric field-driven disruption of a native -sheet protein conformation and generation of a helix-structure. *National Library of Medicine*, 2010. doi: https://doi.org/10.1016%2Fj.bpj.2010.04.040.

[7] K. Sitarachu and M. Bachmann. Phase transitions in the two-dimensional ising model from the microcanonical perspective, 2020.

[8] Wei Cai. Stanford university introduction to statistical mechanics, handout 12. ising model. URL `http://micro.stanford.edu/~caiwei/me334/Chap12_Ising_Model_v04.pdf`.