

# HW10 Write-Up

Nina Martinez Diers\*  
*Bryn Mawr College Department of Physics*  
(Dated: May 3, 2024)

In this homework, algorithms to look at the magnetic properties of Ising and XY models of a lattice using Metropolis style Markov-chain Monte Carlo simulations.

## 1. INTRODUCTION

In this homework, we examine the evolution in energy and magnetization of a system that can develop a spontaneous magnetic moment using a Metropolis-style simulation. We perform the simulation with two different models for a 2-dimensional lattice structure: the Ising model and the XY model.

In the Ising model, the energy of a two-dimensional lattice is

$$E = -J \sum s_i s_j \quad (1)$$

where  $i$  and  $j$  denote lattice sites and  $s_i$  and  $s_j$  are the spins at those lattice sites [1]. In the Ising model, atoms can either have spin-up or spin-down, denoted by spin equal to positive or negative 1.

Conversely, in the XY model, spins are given an angle  $\theta$  which can be between 0 and  $2\pi$ . In the XY model, the energy is given by the equation:

$$E = -J \sum \cos(\theta_i - \theta_j) \quad (2)$$

[2] The magnetization of the lattice in either model is found from the sum of the spin states,  $M = \sum s_i$ . The more the spin states are in alignment, the higher the magnetization of the system.

## 2. EXPERIMENT

In the algorithm for the Ising model, a function was defined to calculate the energy, implementing periodic boundary conditions. This was done by creating two matrices of the spin states and superimposing them to perform multiplication (element-by-element multiplication instead of traditional matrix multiplication). This was done twice, once to calculate side-by-side neighbor products, and once to calculate top-and-bottom neighbor products, as explained in Figure 1. The elements of both product matrices were summed and multiplied by  $-J$  to find the total energy of the spin matrix.

Using the energy function, a Metropolis-style Markov-chain Monte-carlo simulation was performed (Fig. 2). Starting with a  $20 \times 20$  matrix of randomly assigned

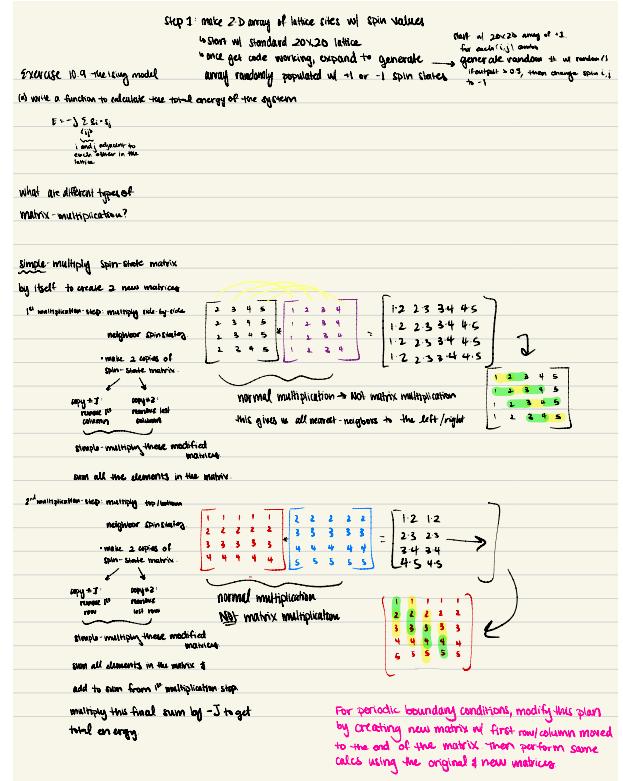


FIG. 1: [Pseudocode of the method used to calculate the Ising model energy of a two-dimensional lattice. Special emphasis is given to the calculation of the products of spin states of neighboring lattice sites. Method with and without implementation of periodic boundary conditions is discussed.]

spin states, random coordinates were generated to flip the spin of that element. If the energy of the new system was less than or equal to the energy of the old system, the change was accepted. If the energy of the new system was greater than the energy of the old system, it was accepted with a frequency of  $e^{-\beta(E_{new} - E_{old})}$ , where  $\beta = \frac{1}{k_B T}$  for temperature  $T$ . Temperature was set to 1 with units in which the boltzmann constant equaled 1, and  $J = 1$  was used. If the change was accepted, the new system was adopted, but if change was rejected, the old system was kept, with the specified element's spin reverting back to its previous state. This was performed for 1 million monte-carlo steps.

In the algorithm for the XY-model, a very similar algorithm was implemented, but using new functions to calculate the energy and commands to propose new spin

\*Electronic address: [nmartinezd@brynmawr.edu](mailto:nmartinezd@brynmawr.edu)

*Metropolis-style  
Simulation of  
Ising model*

(a) Step 1 make 2D array of lattice sites w/ spin values  
 ↳ Start w/ Standard  $20 \times 20$  lattice.  
 ↳ Once get code working, expand to generate  
 array randomly populated w/  $+1$  or  $-1$  spin states.

Step 2: For loop  $\rightarrow$  for  $i$  in range (10) Start w/  $20 \times 20$  array of  $+1$ .  
 ↳ for each  $(i,j)$  coords → generate random  $\rightarrow$  if random() > 0.5, then change spin  $s_{i,j}$  to  $-1$ .  
 ↳ If spin state, between  $+2 \rightarrow -2$  or  $-2 \rightarrow +2$  for same  $i,j$ , commanding  $\rightarrow$  save this as Spin-test (a num matrix).  
 calculate new energy using formula from (a)  
 use metropolis acceptance formula:  $P = \begin{cases} 1 & \text{if } \Delta E \leq 0 \\ e^{-\Delta E / kT} & \text{if } \Delta E > 0 \end{cases}$

If accepted,  $s_{i,j} = \text{Spin-test}$  (old spin - test replace old spin matrix).  
 If rejected, repeat code.

(c) plot magnetization,  $M = \frac{1}{N} \sum s_i \rightarrow$  sum all vals in spin array for each iteration of algorithm above. ( $10^6$  million monte carlo steps)  
 ↳ average time (in one dimension)  
 ↳ Start out using 10000 time steps

(d) use visual package  
 white=negative, green=positive  
 generate array of little balls, color them based on

FIG. 2: [Pseudocode of implementation of a Metropolis-style simulation of the Ising model for a two-dimensional lattice.]

states (Fig. 3). The experiment started using an ordered system, with all spin angles equal to zero. When a new spin was proposed, a random number with uniform distribution between 0 and 1 was generated and scaled by a factor of  $2\pi$  to generate the new proposed spin-state.

However, because by populating the spin matrix with floats, a significant amount of error was introduced by the previous method of copying the entire matrix to a new matrix and then replacing the entire old matrix with the test matrix if the new spin state was accepted. To counteract this cumulative error, the original value for the randomly chosen lattice site was saved under a new variable before the spin at that site was changed experimentally. If the change was rejected, the original value replaced the test value in the matrix. A simulation using the XY-model was run for 10,000 monte-carlo time-steps for varied temperature. The system energy was determined by averaging the final 50 values of total system energy normalized over the total number of lattice sites.

### 3. RESULTS

Although the average magnetization of the Ising-model system was consistently equal to zero, the spontaneous magnetic moment of the lattice was highly varied, with the magnetization oscillating heavily between  $\pm 0.2$ . This means that spontaneous magnetic moments arise easily

*HAN: the XY model*

use Ising model and adapt to XY model  
 ↳ instead of getting products of neighboring spins, get differences → instead, calculate the indices of the differences before performing one more.

Step 1: reuse function specifics from code, as well as magnetization specific.

Step 2: change energy function to calculate energy using  $E = J \cdot \sum s_i s_j$

Step 3: change spins original matrix to be filled with zeroes for ordered system.

When random spin is changed, instead of changing its sign, change the value of  $s_{i,j}$  to a random  $\theta$  between  $0 \rightarrow 2\pi$ .

Step 4: make temperature list and keep experiments for walls of temp.  
 calculate energy of the system by averaging the last 100 values of total energy, not monte-carlo timestep.  
 for each value of temperature  $T$  add that to a new list.

FIG. 3: [Pseudocode of the adaptation of a Metropolis-style simulation of the Ising model to the XY model for a two-dimensional lattice.]

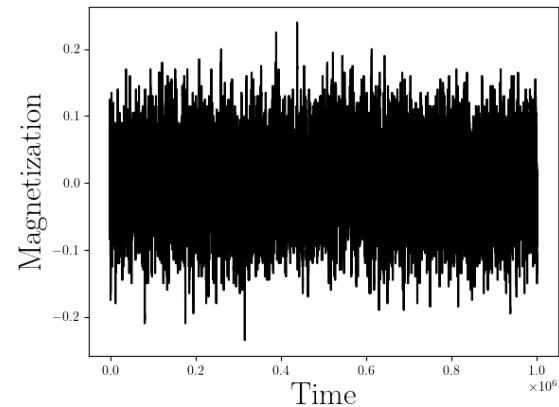


FIG. 4: Magnetization of a  $20 \times 20$  lattice system was calculated implementing the Ising model using a Metropolis simulation over 1000000 monte-carlo time-steps for  $J = 1$ ,  $T = 1$  and  $s_i = \pm 1$

in the system, but they are so fleeting that the bulk behavior observed is no net magnetization.

When the simulation of the Ising model of the lattice is animated to observe the changing spin states, the spins appear to change in bursts in between periods of more prolonged stagnation. The frequency of accepts appears

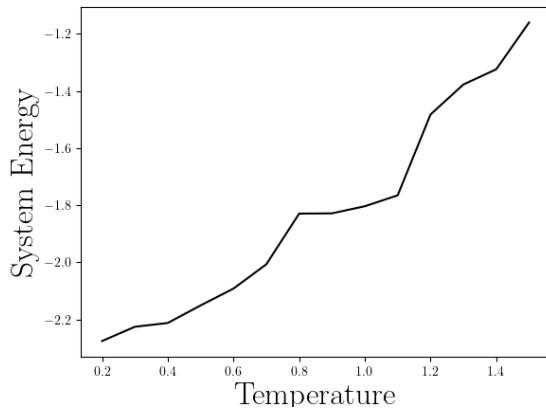


FIG. 5: System Energy as a function of temperature for an XY model of a  $20 \times 20$  lattice system. Metropolis simulation ran over 10000 monte-carlo time-steps for  $J = 1$ , and  $0 \leq \theta_i < 2\pi$ , and the system energy was determined by the normalized average over the last 50 time-steps.

higher at the beginning of the experiment, decreasing in frequency towards the end of the experiment. This behavior may be by the system reaching energetically-stable spin configurations that makes it less likely to accept a new spin state. When one spin changes, it may cause some instability that increases the energy of the system temporarily as a series of spin changes occur un-

til a new more energetically favorable configuration is reached. This process may also simply be explained by the randomness of the accept/reject determination process of the algorithm. When temperature is increased, the spins change more frequently. This can be explained by an increase in the total entropy of the system with the temperature increase. Increasing the temperature also increases the probability that the change will get accepted since by the Metropolis style simulation the acceptance probability  $e^{-\beta*(E_{new}-E_{old})}$  when the energy of the new system is higher than than the energy of the old system is more likely to lead to an acceptance when the temperature is higher.

When the energy of the XY-model system is plotted as a function of temperature, it is observed that the energy increases with respect to temperature (Fig. 5). This is consistent with the expected behavior, as temperature often correlates with an increase in the energy of a system.

#### 4. CONCLUSIONS

Temperature increase leads to an increase in the frequency of spin state changes and system energy. Additionally, while average magnetization is zero, spontaneous magnetization is highly varied but is not noticed in bulk behavior because the change is so fleeting.

- 
- [1] M. Newman, *Computational Physics* (2013), revised and expanded ed., ISBN 978-1-4801-4551-1.
  - [2] URL [https://www.compadre.org/stpbook/statistical-mechanics-2/ex9\\_3.cfm](https://www.compadre.org/stpbook/statistical-mechanics-2/ex9_3.cfm).

#### Appendix A: Statement of Collaborators

I did not collaborate on this assignment.

#### Appendix B: Survey Question

Pseudocode took 2 hours. Coding took 10 hours. Write-up took 3 hours. I was really excited to figure

out the method to calculate the energy using multiplication and manipulation of arrays. This assignment helped me to develop an understanding of Monte Carlo methods and re-emphasized the concept of cumulative errors in computers and the need to restructure the approach of the algorithm to accommodate for that. I thought the most interesting problem was the animation of the Ising model. The assignment was definitely on the longer side for me.