51+55+5=111

111/117

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

$$E = -J \sum s_i s_j \tag{1}$$

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 θ which can be between 0 and 2π. In the XY model, the energy is given by the equation:

$$E = -J \sum \cos(\theta_i - \theta_j) \tag{2}$$

[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 20×20 matrix of randomly assigned

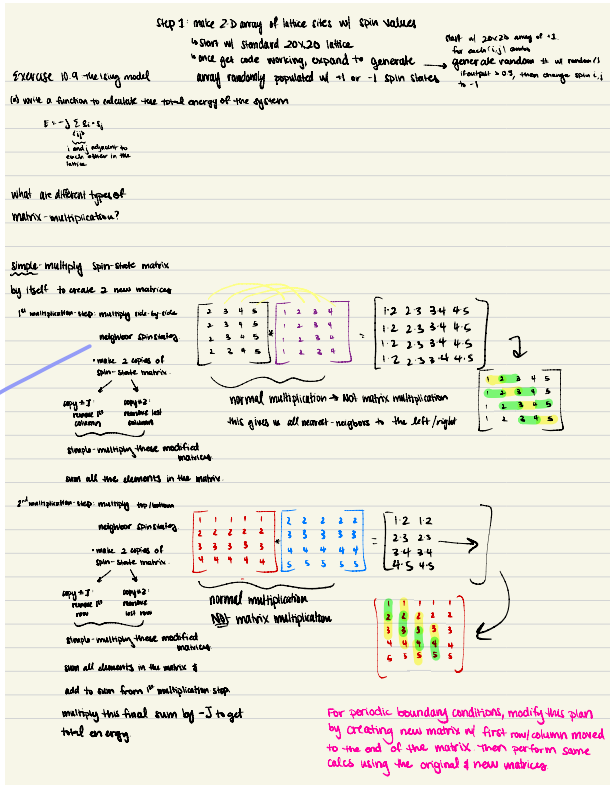

FIG. 1: [Pseudocode of the method used to calculate the Ising model energy of a two-dimensional lattice. Special emphasis is given to the calculation of the products of spin states of neighboring lattice sites. Method with and without implementation of periodic boundary conditions is discussed.]

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

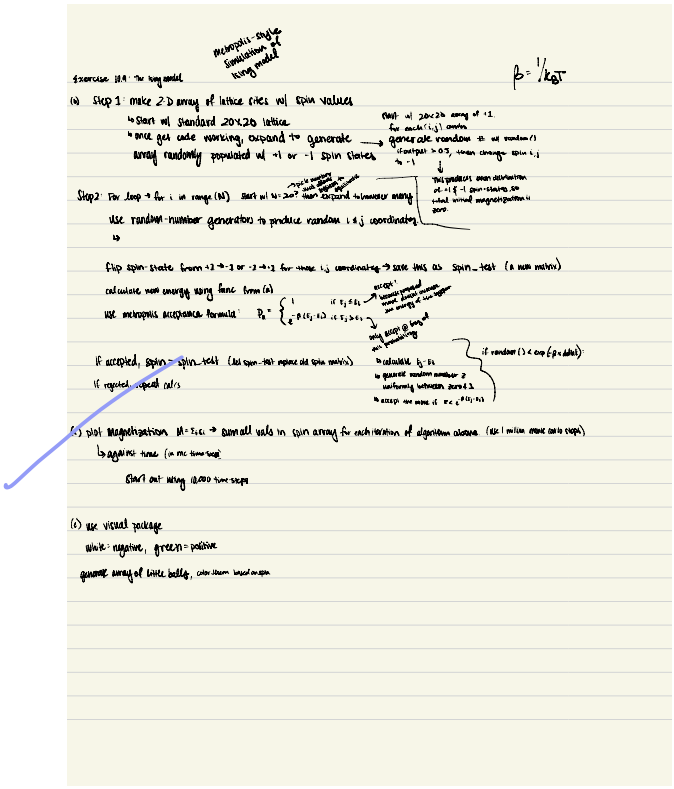

FIG. 2: [Pseudocode of implementation of a Metropolis-style simulation of the Ising model for a two-dimensional lattice.]

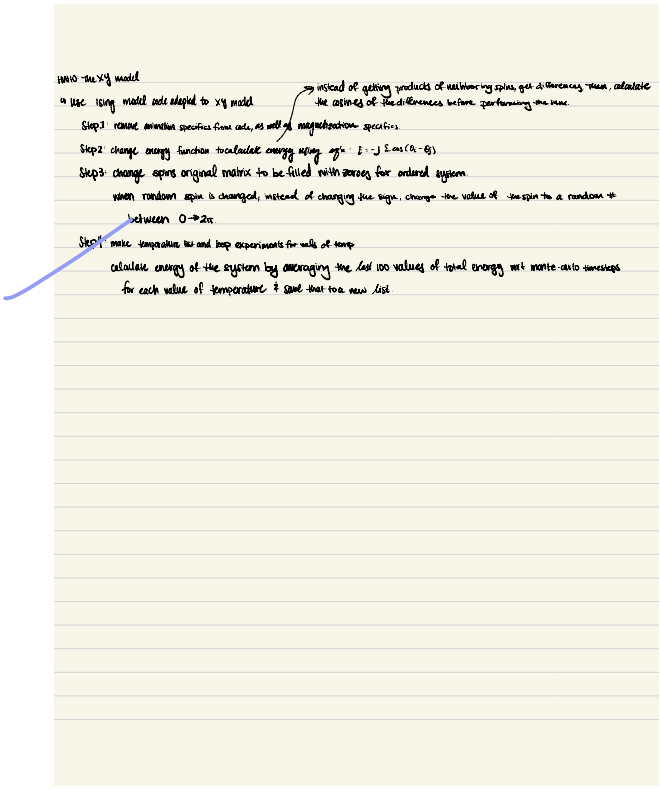

FIG. 3: [Pseudocode of the adaptation of a Metropolis-style simulation of the Ising model to the XY model for a two-dimensional lattice.]

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 2π 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.

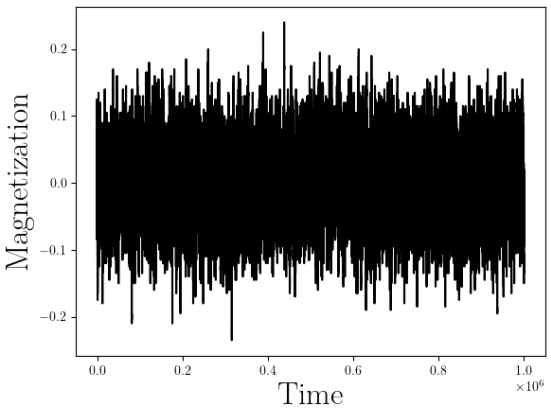

FIG. 4: Magnetization of a 20×20 lattice system was calculated implementing the Ising model using a Metropolis simulation over 1000000 monte-carlo time-steps for $J = 1$, $T = 1$ and $s_i = \pm 1$

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 ± 0.2. This means that spontaneous magnetic moments arise easily 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

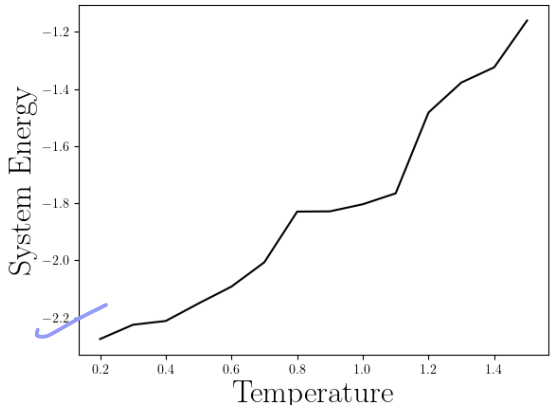

FIG. 5: System Energy as a function of temperature for an XY model of a 20×20 lattice system. Metropolis simulation ran over 10000 monte-carlo time-steps for $J = 1$, and $0 \leq \theta_i < 2\pi$, and the system energy was determined by the normalized average over the last 50 time-steps.

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 until 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.

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.

10.9 51/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

2 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points)

2 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) (+3 points)

missing animation -1

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

1 6. Is the output answer correct? (+4 points).

too much noise, magnetizations should have larger magnitudes, -3

3 7. Is the code readable? (+3 points)

no animation

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

---

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

3 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named?

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

3 . Are key figures and numbers from the problem given? (+ 3 points)

4 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3 . Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

1 . Is a brief explanation of physical context given? (+2 points) need animation discussion -1

1 . If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3 . Is the algorithm used explicitly stated and justified? (+3 points)

2 . When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)

#2 55/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

2 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points)

3 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) (+3 points)

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

2 6. Is the output answer correct? (+4 points). small error in energy function

3 7. Is the code readable? (+3 points) -1

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

3 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named?

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

3 . Are key figures and numbers from the problem given? (+ 3 points)

4 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3 . Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

2 . Is a brief explanation of physical context given? (+2 points)

1 . If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3 . Is the algorithm used explicitly stated and justified? (+3 points)

2 . When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2 . Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)

1/2 . Are collaborators clearly acknowledged? (+1 point)

. Are any outside references appropriately cited? (+2 point)