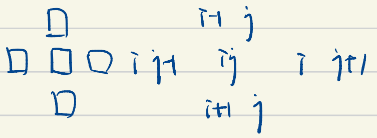
Attached written notes are my pseudo code for the problems.

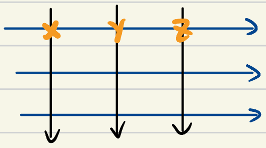
10.9 a).



(j-1 j) (j j+1)

(i-1 i) (i i+1)).

$\begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix} \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix}$



(i, j).

j j+1 j+2 ... N

j j+1 j+2 ... N

$(x+y)+(y+z)$
$= x+2y+z$

```
for i in len(N):
    for j in len(N-1):

        s[i,j] * s[i,j+1]
```

b).

old: $-1 \cdot 1 \cdot (-1) = +1$

$\begin{matrix} -1 & \boxed{-1} & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{matrix} \rightarrow$

new: $-1 \cdot 1 = -1$

$\begin{matrix} -1 & \boxed{1} & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{matrix}$

$-1 - (+1) = -2$

2 · new.

```
After calculating the ΔE.
 if ΔE < 0
     config[i][j] x = -1

  if ΔE >= 0 :
      if random ( ) < $e^{-ΔE/kT}$
         config[i][j] x = -1

      else
         config[i][j] x = 1.

  return (config).
```

c).

```
        M = [ ]
def magnetization ( ):
```

```
                    ...      (same as part b).

          m = np.sum (config).
          M.append ( m)
          return (M).
```

d).

```
      def  animation (   ):
                   ...

           img. set_data (config)
```

XY model

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

```
For all rows : for j in range(N):
                   for i in range (N-1):
                        energy += cos (config[j, i] - config
                                       [j, i+1]
```

same thing for all columns.

```
MCMC :
     for n in (steps):
          i = random
          j = random

          θ = config [i][j]
          θ' = (0, 2π)
          copy = np.copy (config).
          config [i][j] = θ'

          ΔE = energy (config) - energy(copy).

                  ...
```

same in 10.9

```
T=0.2
print (energy(runmcmc( )).

Energy = [ ]
T_ = np.arange (0.2, 1.6, 0.1).
for T in T_ :
     energy(runmcmc(config)))
     Energy.append
```

56 + 54 + 5 = 115

115/117

PHYS 304 AS11

Xiyue Shen[*]
Haverford College Department of Physics
(Dated: May 3, 2024)

I collaborated with Yang. I have two python files.

PHYS_304_AS11_10.9_Xiyue_Shen.py file is the Ising model problem, labeled as 10.9 in the textbook.

PHYS_304_AS11_XY Model_Xiyue_Shen.py is the XY model problem in the assignment instruction.

1. THE ISING MODEL

1.1. part a

For this part, we aim to calculate the total energy using an Ising model. The energy equation is given by,

$$E = -J \sum_{<ij>} s_i s_j, \tag{1}$$

where J is a constant determining the material's property (the interaction between atoms), either ferromagnetism or antiferromagnetic.

I used two loops to define the total energy function. One loop for going through all rows; for every row, the interaction between each pair is calculated, ie,

$$config[j][i] \times config[j][i+1],$$

where $config$ is my lattice model, j represents the row and i represents the column.

For every column, we have,

$$config[j][i] \times config[j+1][i]$$

1.2. part b

For this part, we aim to code up the Metropolis algorithm. In this part, I define a function called "flip_energy", which takes the configuration of Ising model as the input. Firstly, we choose a random spin, such as config[i][j]. Then, calculate the energy of this specific spin. Then, the negative of twice the energy value is the cost to flip the spin. Flipping the spin is favored if the energy is smaller than zero. If the energy is bigger than zero, we introduce a random value between zero and one. Then, compare the probability of this cost of energy (Boltzmann distribution) with the random number. If the probability is higher than the random number, then the flip could happen; if not, the flip won't happen. Very different from the classical world, though the energy is not favored, there's still some possibility for the thing to happen.

1.3. part c&d

In part c, we will calculate the magnetization. The code is exactly the same as part b. I added two things: one is to put everything in a loop in terms of steps, and the other is to sum up all spins at the end but inside the loop. The temperature and Boltzmann constant are set to be one. The magnetization is shown below. In the long term, since the temperature is pretty low, once we trigger a single flip in the system, they tend to balance it, resulting in an overall shift toward a specific spin state. Because the temperature is low, the spins tend to occupy the lowest energy state. They will align toward the same direction, wither upward or downward. In figrure 1 and figure 2, we observe that the magnetization goes -400 and 400. Both plots show some fluctuation at the beginning but a trend of all spins toward either spinning upward or downward in the final state.

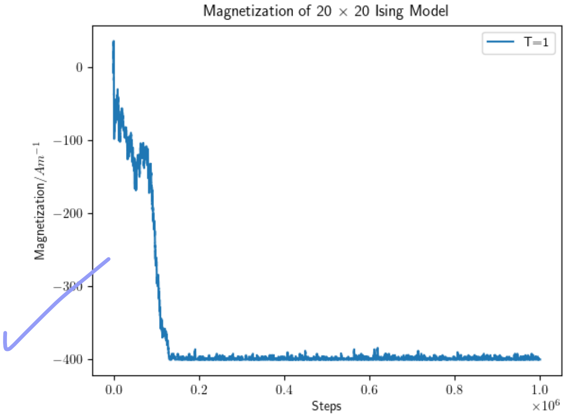


FIG. 1: [Magnetization of 20 × 20 lattice gas at T=1, $m = -400$]

[*] Electronic address: xshen2@brynmawr.edu; URL: Optionalhomepage

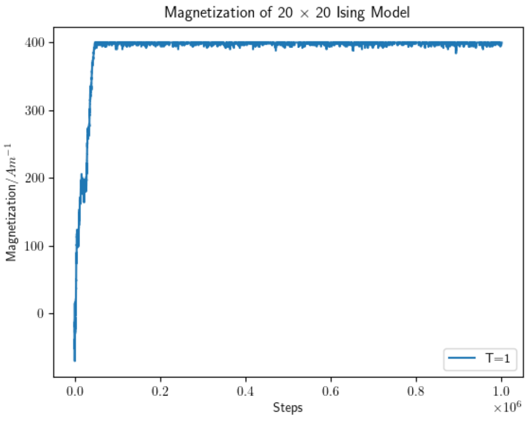


FIG. 2: [Magnetization of 20×20 lattice gas at T=1, $m = 400$]

1.4. part e

For part e, an animation of 20×20 lattice square is created. When $T = 0$, the lattice initially shows an equal distribution of spinning up and spinning down. As time goes on, one of the colors will turn into another one completely, as shown by figure 3. The whole square exhibits the same color. However, as we increase the temperature, the final state will have an equal number of colors, as shown by figure 4. At low temperatures, the configuration tends to the lowest energy state, which is aligned up, while at high temperatures, the neighboring spins tend to have different directions, which leads to higher entropy.

2. XY MODEL

Similar to the Ising model, the XY model enables the spin to have a tilted angle. And the energy is,

$$E = -Jcos(\theta_i - \theta_j), \tag{2}$$

J is the constant that specifies the interaction between atoms.

When $T = 0.2$, the energy after running for 10000 times is -724.211. In figure 5, the x-axis represents the varying temperature. The y-axis is the energy. We observe that the energy starts from the lowest energy and then gradually increases. This agrees with the statement that as temperature increases the system is provided with more thermal energy, so the change to have the spin-flip is higher, resulting in a higher-energy state.

At low temperatures, the spins tend to align in the same direction, which gives a negative energy according to equation 2. As temperature increases, the disorder in the system increases as well. This will lead to a random distribution of spin up and spin down. In this case, the average energy and magnetization is zero.

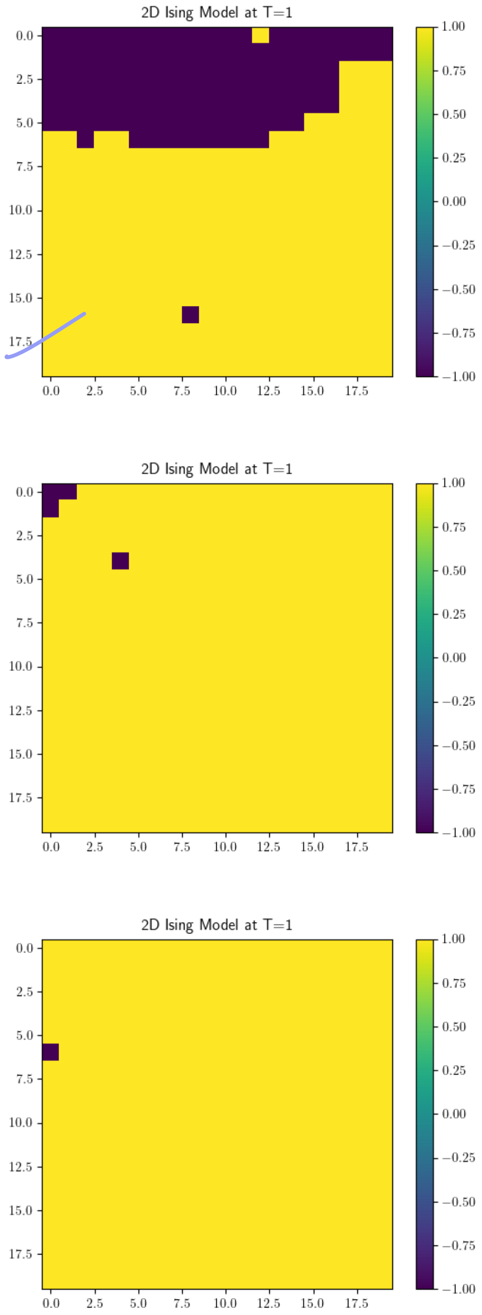


FIG. 3: [Animation of magnetization at T=1]

3. ISING MODEL

The Ising model is a statistical model used to describe a thermodynamic system. The universality theory can be applied to many different systems, spanning from condensed matter to biophysics. A basic setup is that we will have a lattice square, say 20 × 20, and each spot will fill with either +1 or −1. +1 represents the spin pointing upward, while −1 is the pointing downward spin. We

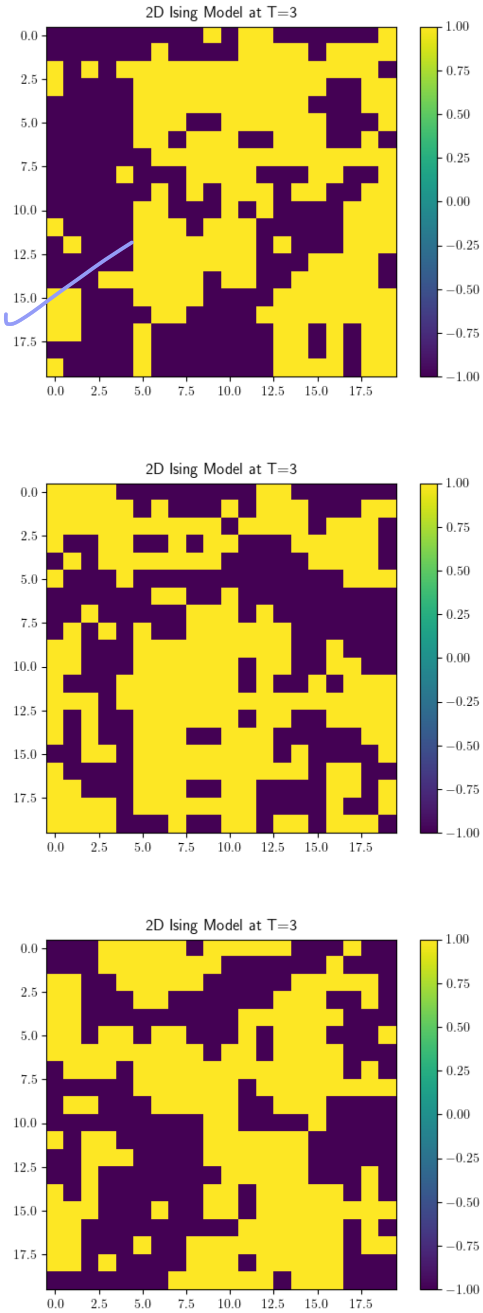


FIG. 4: [Animation of magnetization at T=3]

can also use 0 and 1 to indicate the existence of atoms in other systems. For such a system, the Hamiltonian consists of two parts. One is the external field term, and the other is the interaction between neighboring spins. And the equation is,

$$H = -J \sum_{i,j} s_j s_j - \sum_{i} h_j s_j, \tag{3}$$

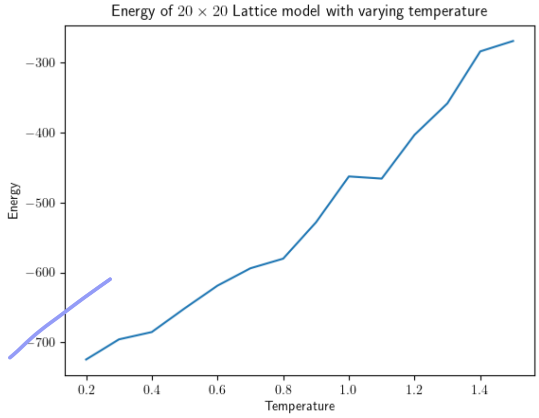


FIG. 5: [Energy of XY model with change of temperature]

where J specifies the system is ferromagnetism or anti-ferromagnetic, h_j is the external magnetic field.

4. SURVEY

✓ .5

I spent roughly 6 hours on this homework. I'm now more familiar with the Ising model. I like the first one most, since it introduces the Ising model coding step by step. I think this set is about the right length.

5. UNGRADED PART

I have done all of the required and ungraded work.

10.9

56/56

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)

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

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

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

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

#2 54/56

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)

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

↳ energies need to be divided by 20·20 −1

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

. 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) expand description of the final trend −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)