Simulating the Ising Model - CompPhys Homework 1

-Aleena Sibi(s6alsibi) and Kriti Baweja(s6krbawe)

Questions

Que-1) Discuss the Physical meaning of J, in particular the sign of J and the role it plays in magnets?
Ans-1) J in the expression represents the interaction strength(coupling) between two spins. The sign of J will determine the kind of interaction between the spins i.e. if its is attractive or repulsive, if J is positive the spins will tend to align with each other and if J is negative the spins tend align antiparallel to each other because it is energetically favourable.
This is also seen in the case of magnets where the potential energy of the (classical) magnetic dipoles is given by **-m.B** where **m** is the magnetic dipole moment and **B** is the external magnetic field. The (classical) dipole magnets have the lowest energy when they are parallel to **B** and that is when its potential energy is the minimum and vice versa.
J>0 is analogus to classical magnets as the spins tend to align with each other.

Que-2) Clarify what it means to have periodic boundary conditions?
Ans-2) Periodic boundary conditions in one dimension essentially means that the first spin is interacting with the last spin i.e. our lattice is circular or a ring. There are no end points.

Que-3) what are the relevant dimensionless ratio in the problem?
Ans-3) The Relevant dimensionless ratios are:
i) J/T
ii) h/T

```
In [47]:  import numpy as np
          import random as r
          import matplotlib.pyplot as plt
          from tqdm import tqdm
```

$$H(s) = -J \sum_{\langle x,y \rangle} s_x s_y - h \sum_x s_x$$

Role of J
- Strength of coupling between the spins
- The sign of J decides the kind of interaction between the spins
- There are also systems where J depends on the spin pair we select???
- Choosing J = 1 implies what? => When spins are aligned, the energy is negative (attractive??) and when not aligned, it's positive (repulsive???).

Role of h
- Measure of the external magnetic(?) field applied

- Trying to see with differetn values of h. <\li>
- h ∈ [−1, 1]

In [86]:
```python
N = 10
J = 1.
spins = [np.random.choice([-1,1]) for i in range(N)]
spin1 = np.copy(spins)
```

In [87]:
```python
J = 1.
T = 1.
def hastings(h,k,spins):
    dM = 0
    N = np.shape(spins)[0]
    dE = 2*J*spins[k]*spins[k-1] + 2*J*spins[k]*spins[(k+1)%N] + 2*h*spins[k]
    if (dE < 0 or np.random.uniform(0,1) < np.exp(-dE/T)):

        dM = -2*spins[k]
        spins[k] *= -1

        #print(dM)
    return dM

def mExactN(h,N,J=1):
    s = np.sinh(h)
    c = np.cosh(h)
    expm4J = np.exp(-4*J)
    ratio = (c-np.sqrt(s**2+expm4J))/(c+np.sqrt(s**2+expm4J))
    answer = (1-ratio**N)/(1+ratio**N)
    answer *= s/np.sqrt(s**2+expm4J)
    return answer

# exact answer in thermodynamic limit
def mExact(h,J=1):
    s = np.sinh(h)
    c = np.cosh(h)
    expm4J = np.exp(-4*J)
    answer = s/np.sqrt(s**2+expm4J)
    return answer
```

$\langle m \rangle$ vs h

In [88]:
```python
Nsim = 10000 #Number of simulations that we have to run
harr = np.linspace(-1,1,101)

Marr = []

for i in tqdm(range(len(harr))):
    spins = spin1
    M = np.sum(spins)
    Mp = 0

    for k in range(Nsim):
        p = r.randint(0,N-1)
        #print(k)
        dm = hastings(harr[i],p,spins)
        M = M + dm

        if( k>3000):
            Mp += M/N

    Marr.append(Mp/7000)
```

```
100%|███████████████████████████████████|
      █████████| 101/101 [00:20<00:00,  4.98it/s]
```

```
In [109]: plt.figure(figsize=(10,6))
          plt.title("<m> vs N")
          plt.xlabel("h")
          plt.ylabel("<m>")
          plt.plot(harr,Marr,'.',label = 'Simulation')
          plt.plot(harr,mExactN(harr,N),label = 'Theoretical Curve (N = 20)')
          plt.plot(harr,mExact(harr),'--',label = 'Theoretical Curve (N->$\infty$)')
          plt.legend()
```

Out[109]: <matplotlib.legend.Legend at 0x235be91bd60>
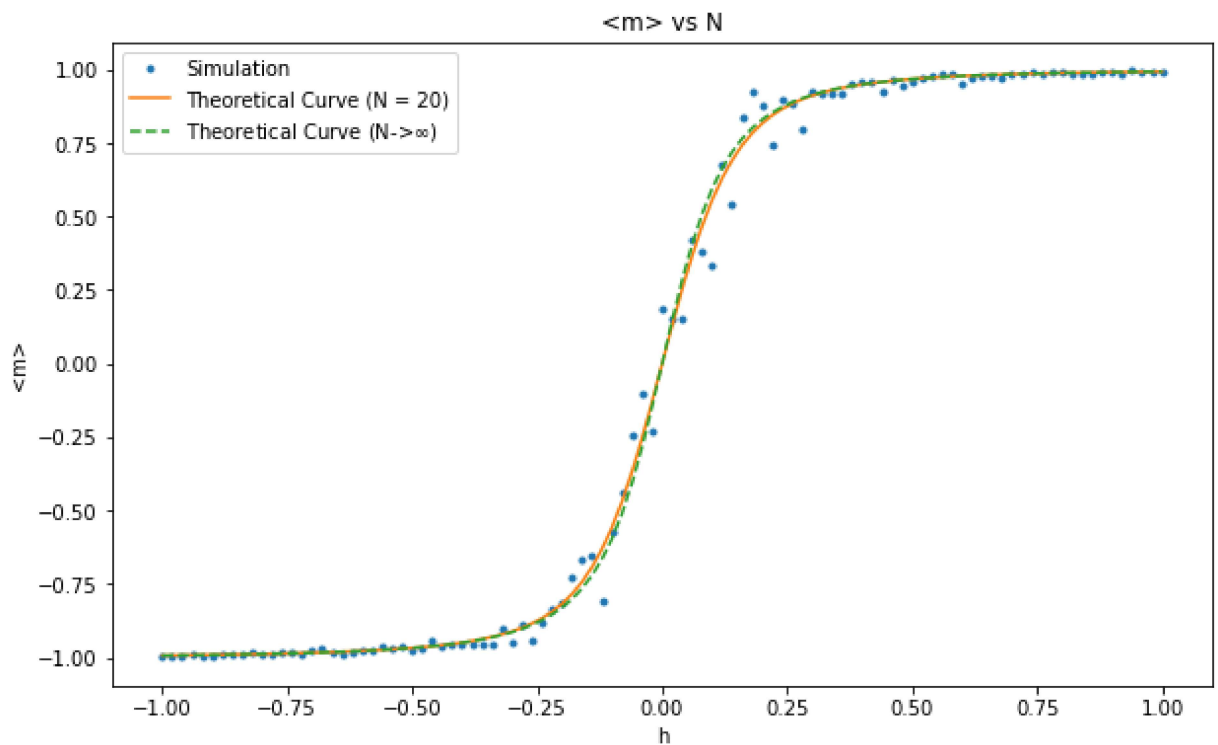
The data points obtained via sampling agree very well with both the analytical curve, despite having such a small lattice size. As expected, for lower values of h (h ~ [-0.25,0.25]), the magnetization tends to 1, i.e. higher the field strength, the more likely it is for the spins to align themselves with the field.

$\langle m \rangle$ vs N

In [121]:

```python
Nsim = 10000 #Number of simulations that we have to run
Narr = np.arange(1,20,1)

Marr1 = []

for i in tqdm(range(len(Narr))):
    spin = [np.random.choice([-1,1]) for i in range(Narr[i])]
    M = np.sum(spin)
    Mp = 0

    for k in range(Nsim):
        p = r.randint(0,Narr[i]-1)
        dm = hastings(1.,p,spin)
        M = M + dm

        if( k>3000):
            Mp += M/Narr[i]

    Marr1.append(Mp/6999)
```

```
100%|████████████████████████████████████████|
█████████| 19/19 [00:06<00:00,  3.00it/s]
```

In [122]:
```python
plt.figure(figsize=(8,8))
plt.title("<m> vs N")
plt.xlabel("N")
plt.ylabel("<m>")
plt.plot(Narr,Marr1,'.')
```

Out[122]: [<matplotlib.lines.Line2D at 0x235bebc3760>]
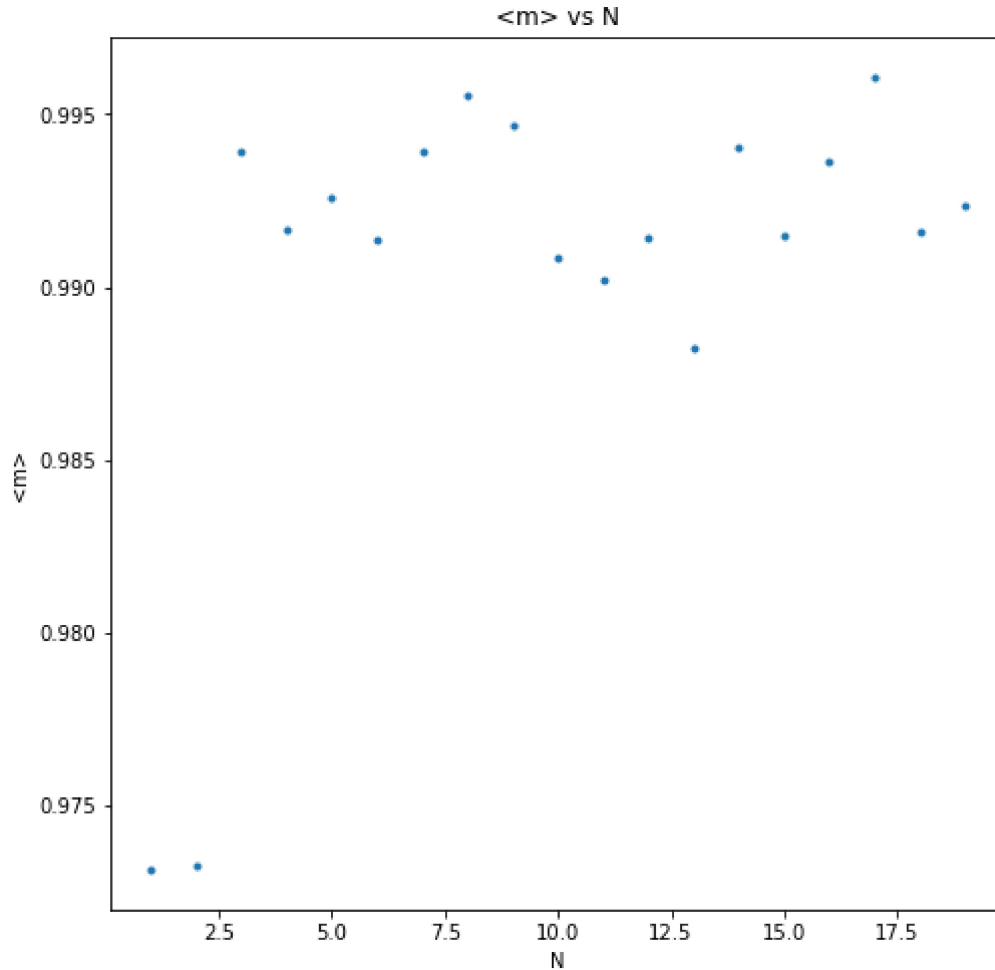
Here, we have chosen h = 1. (We did try to plot for other values of h, but haven't shown any of it here), As expected, since the field is very high, the spins are aligned with the field (and hence have magnetization ~ 1), As N increases, we can see the magnetization slowly decreasing.