Week Four PHY-480

Lewis

September 2025

Answers to Questions

1. Magnetic Phase Transitions

A **magnetic phase transition** occurs when a magnetic material moves from an ordered state to a disordered state as temperature increases. The key quantity used to describe this is referred to as magnetization which is used to track the orders.

- **First-order transition:** The magnetization changes discontinuously. When a critical temperature is hit, there is a jump in magnetization, and the system can potentially interact with energy by absorbing or releasing it.
- **Second-order transition:** The magnetization changes continuously. Its derivative with respect to temperature diverges at the transition point. There is no latent heat, but fluctuations will dominate the behavior near the critical temperature.

2. Complete Graph and Adjacency Matrix

A **complete graph** with N vertices is a graph where every vertex is connected to every other vertex. For such a graph, the adjacency matrix has:

- Zeros on the diagonal as we do not want any self interactions.
- Ones everywhere else as we want distinct pairs of vertices to be connected.

For a complete graph with 4 vertices (K_4), the adjacency matrix is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Which clearly shows what was outlined above, that the zeros are on the diagnol angle and the ones fill everywhere else.

3. Python Code for Metropolis Monte Carlo on a Complete Graph

The Ising model Hamiltonian for a complete graph with ferromagnetic interactions is:

$$H = -J \sum_{i<j} S_i S_j$$

where $S_i = \pm 1$. Below is pseudocode for implementing the Metropolis Monte Carlo method as requested by the question.

```
1. Define system parameters:
      N = number of spins
      J = interaction strength (ferromagnetic if J > 0)
      T = temperature
      steps = number of Monte Carlo updates

2. Initialize the system:
      Assign each spin randomly as +1 (up) or -1 (down).

3. Define energy change rule:
      For a trial flip of spin i:
          Compute E = 2 * J * (spin_i) * (sum of all spins) - 2 * J * (spin_i)^2

4. Begin Monte Carlo simulation:
      Repeat for 'steps' iterations:
          a. Randomly select a spin i.
          b. Calculate E for flipping this spin.
          c. Apply Metropolis acceptance criterion:
                - If E < 0 (energy decreases), accept the flip.
                - Otherwise, accept with probability exp(-E / T).
          d. If accepted, flip spin i (spin_i → -spin_i).
          e. Measure magnetization M = (sum of all spins) / N.
          f. Record M for statistics.

5. After all steps:
      Compute the average magnetization over the simulation.
      Report ⟨M⟩ as the equilibrium magnetization of the system.
```

Explanation:

- At each step, we make a random spin and a trial flip is made like flipping a coin.
- The energy change ΔE is computed, and the flip is accepted with probability $\exp(-\Delta E / k_B T)$.
- The magnetization is computed as $M = \frac{1}{N} \sum_i S_i$ and tracked across the simulation.
- Averaging M over many steps provides the equilibrium magnetization at the given temperature.

4. Detailed Balance: Metropolis and Heat Bath

The detailed balance equation is:

$$\frac{w_{l\to k}}{w_{k\to l}} = e^{-\beta(E_k - E_l)} = x$$

where $x = \exp[-\beta(E_k - E_l)]$.

- **Metropolis condition:**

  $$F(x) = \min(x, 1)$$

  If $x < 1$, then $w_{l\to k} = x$ and $w_{k\to l} = 1$, so:

  $$\frac{w_{l\to k}}{w_{k\to l}} = \frac{x}{1} = x$$

  If $x > 1$, then $w_{l\to k} = 1$ and $w_{k\to l} = 1/x$, so:

  $$\frac{w_{l\to k}}{w_{k\to l}} = \frac{1}{1/x} = x$$

  Thus the Metropolis algorithm satisfies detailed balance.

- **Heat bath condition:**

  $$F(x) = \frac{x}{1+x}$$

  Then,

  $$\frac{F(x)}{F(1/x)} = \frac{\frac{x}{1+x}}{\frac{1/x}{1+1/x}} = \frac{x}{1+x} \cdot \frac{1+x}{1} = x$$

  Hence the Heat Bath method also satisfies detailed balance.

5. Python Code for Periodic Boundary Conditions

Below you will find a Python function that implements periodic boundary conditions for a square lattice with nearest-neighbor interactions as requested in question 2.

```
Define function get_neighbors(i, j, L):
a. Purpose: return the nearest neighbors of a site (i, j) on an L × L lattice with periodic boundary

For a given site (i, j):
a. Compute "up" neighbor → ((i - 1) mod L, j)
b. Compute "down" neighbor → ((i + 1) mod L, j)
c. Compute "left" neighbor → (i, (j - 1) mod L)
d. Compute "right" neighbor → (i, (j + 1) mod L)

Return the list of neighbors:
[up, down, left, right]

Example usage:
Set lattice size L = 4
Call get_neighbors(0, 0, L)
Output: list of 4 neighbors, wrapped around edges of lattice
```

This ensures that if a site is at the boundary, its neighbor wraps around to the opposite edge, effectively placing the lattice on a torus.

6. Python Code for Magnetization vs Temperature

```
Define function ising_simulation(L, T, steps, J=1):
    a. Initialize spins randomly:
       Create an L x L lattice where each spin is +1 or -1 at random.
    b. Define energy change rule:
       Define energy_change(i, j):
           - Let s = spin at site (i, j)
           - Find the four nearest neighbors of (i, j) with periodic boundaries
           - interaction = sum of spins at neighbor sites
           - E = 2 * J * s * interaction
           - Return E
    c. Prepare measurement list:
       magnetization = empty list
    d. Monte Carlo simulation:
       For step in 1 to steps:
           i.   Choose random site (i, j)
           ii.  Compute E = energy_change(i, j)
           iii. Metropolis acceptance:
                If E < 0 OR random(0,1) < exp(-E / T):
                    Flip spin at (i, j)
           iv.  Measurement:
                If step is a multiple of (L * L):
                    M = absolute value of (sum of all spins) / (L * L)
                    Append M to magnetization
    e. Return average magnetization:
       mean(magnetization)


Example usage:

    - Set L = 20
    - Define temps = array of 10 values between 1.0 and 4.0
    - Print list of (temperature, magnetization) pairs
```

Explanation:

- Spins are arranged on an $L \times L$ square lattice.
- Periodic boundaries ensure every site has 4 neighbors.
- Energy change is computed when flipping a spin.
- The Metropolis rule decides whether to accept a flip.
- Magnetization $M = \frac{1}{N} \left| \sum_i S_i \right|$ is measured and averaged over sweeps.