5 2D Ising Model

5.1 Introduction

The 2D Ising model is a statistical mechanics framework originally developed to describe ferromagnetic systems. The system is represented as a lattice of spins, σ_i, where each spin can take a value of $+1$ (up) or -1 (down). The two possible values indicate whether two spins i and j are aligned, and thus parallel ($\sigma_i \cdot \sigma_j = +1$), or anti-parallel ($\sigma_i \cdot \sigma_j = -1$). A system of two spins is considered to be in a lower energetic state if the two magnetic moments are aligned.
The energy of a given spin configuration is governed by the Hamiltonian:

$$E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - h \sum_i \sigma_i$$

Here:

- J is the interaction strength between neighboring spins σ_i and σ_j.
- h is the external magnetic field.
- $\langle i, j \rangle$ indicates summation over all pairs of neighboring spins.

Another key property of the Ising model is magnetization, which represents the overall alignment of spins in the system. It is defined as:

$$M = \frac{1}{N^2} \sum_i \sigma_i$$

where N is the dimension of the lattice. The magnetization serves as an order parameter for phase transitions in the system.
The model can be evolved dynamically via the Metropolis-Hasting algorithm. At each iteration, a spin is randomly selected, and the change in energy, ΔE, is computed if the spin is flipped:

$$\Delta E = 2J\sigma_i \sum_j \sigma_j + 2h\sigma_i$$

where $\sum_j$ runs over the nearest neighbors of σ_i. In two dimensions, the sum runs over the 4 nearest spins, whereas in three dimensions this sum would run over the 6 nearest spins.
If the new configuration is energetically favored over the previous state ($\Delta E \leq 0$), the new configuration is kept.
Otherwise, we compare the transition probability p with a random number r (generated from 0 and 1), and accept the new configuration if $r \leq p$.

$$p = e^{-\Delta E / k_B T}$$

where k_B is the Boltzmann constant and T is the temperature. By simulating this evolution, the model captures temperature-dependent behavior, such as phase transitions.

5.2 Task

You are tasked with implementing and simulating the 2D Ising model using GPUs. The project includes the following components:

Dataset Generation

Initialize the dataset as a 2D grid of spins (σ) of size $N \times N$, and ensure the simulation can handle different grid sizes N. The spin initialization can be *cold*, where all spins are set to $+1$ or -1, or *hot*, where each spin is randomly set to $+1$ or -1. Ensure reproducibility by using a fixed random seed if populating the spin lattice randomly. Implement periodic boundary conditions to allow spins at the edges of the grid to interact with their counterparts on the opposite edges.

Implementation of the Ising Model

Implement the Model evolution dynamics by simulating the model for a large number of steps, starting from the chosen configuration, and compute macroscopic properties like magnetization and energy at each step.
At every step, the algorithm must perform:

- *Energy Calculation*: Compute the energy of the system based on the Hamiltonian.
- *Spin Updates*: Use the Metropolis algorithm to iteratively update spins.

After implementing the algorithm for the CPU and testing its behavior, implement the spin updates in parallel using directives and in CUDA.
Compare the runtime performance of the CPU, parallel, and CUDA implementations for a (manageably) large grid size N, and a given number of simulation steps.
Provide a visualization of the spin configurations (e.g., a heatmap where colors represent spin orientations) at the beginning and the end of the procedure to demonstrate the evolution of the system.

5.3 Extra

Hints

- Once a first CUDA implementation is performed, consider using shared memory to optimize energy calculations, especially for local spin interactions.
- Consider possible choices of kernel/block/grid subdivisions to account for the geometry of the lattice.
- Define a strategy to update spins in parallel to avoid race conditions.
- When developing for GPU, start with smaller grid sizes (e.g., $N = 32$) to debug and validate your code before scaling up.

Bonus Features

If willing to explore the topic further, consider the addition of Simulated Annealing by implementing a temperature annealing schedule where T decreases gradually over time:

$$T(t) = T_0 \cdot e^{-\alpha t}$$

where T_0 is the initial temperature, α is the cooling rate, and t is the iteration step.
Compare how annealing affects the final spin configuration and properties like magnetization.