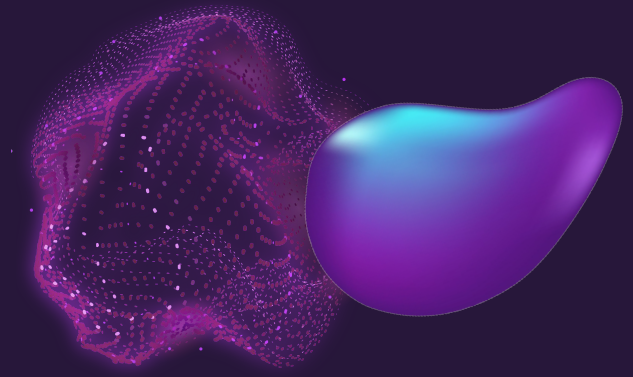


Ising Model

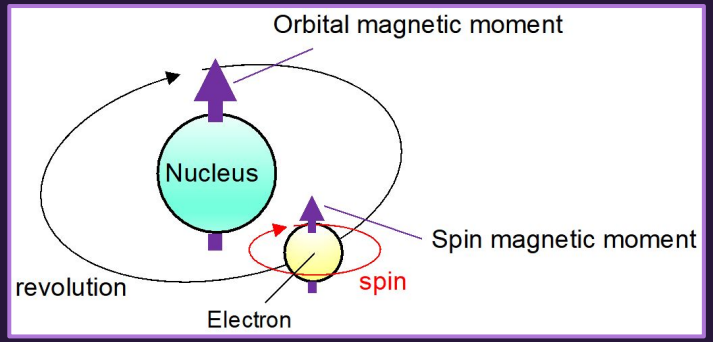
with Parallel Implementations

Taj Gillin

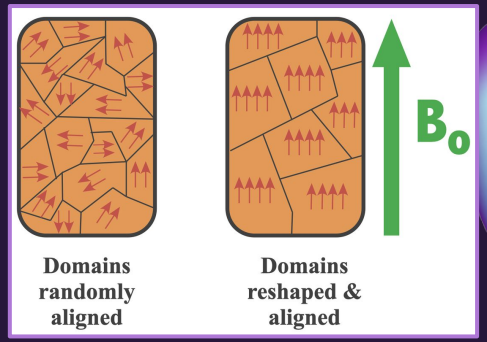
01 Background

Ferromagnetism and the Ising Model

Ferromagnets

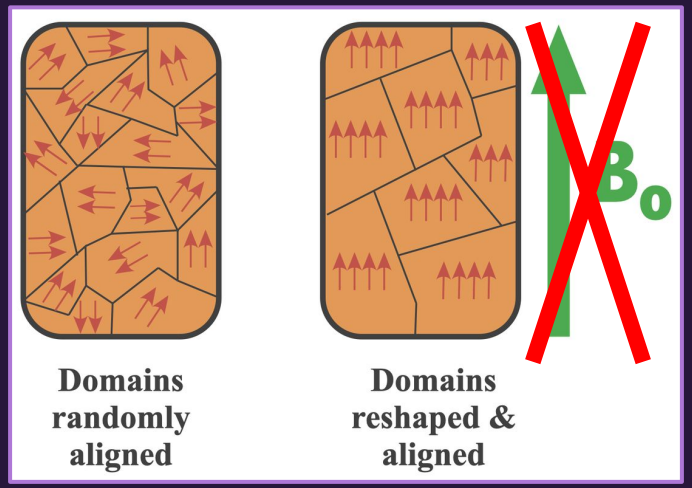


Atoms have magnetic moments



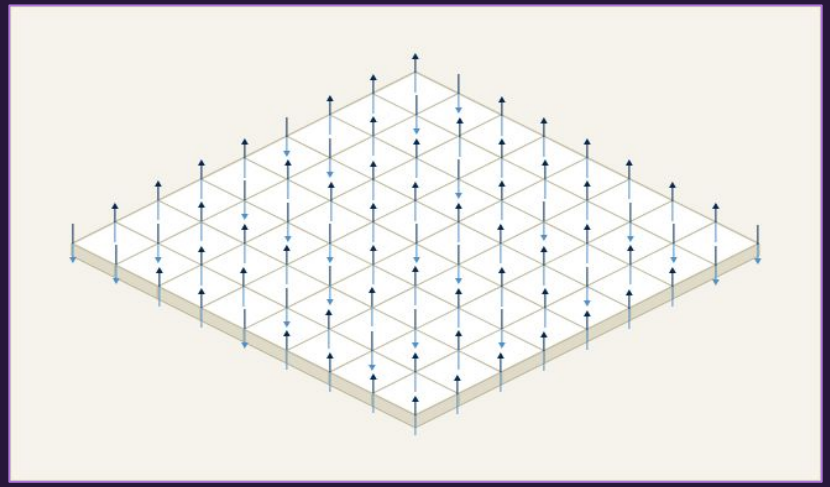
Aligns with magnetic field below critical temperature

Ferromagnets



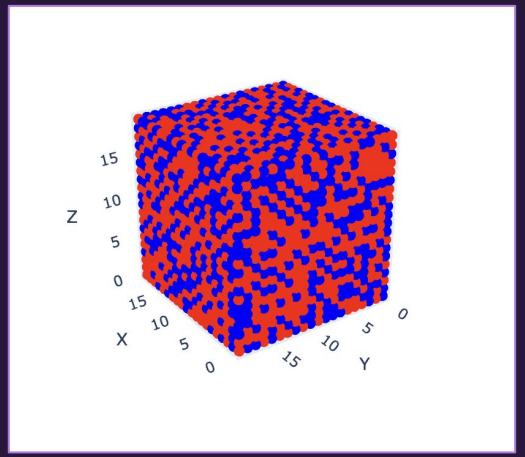
At low temperatures, atoms align with each other and spontaneously polarize!

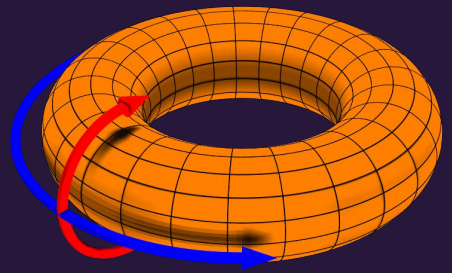
Ising Model



- Spins on a lattice, each with ±1
- Interaction: spins favor alignment with neighbors
- Simple but captures essence of phase transitions and ferromagnetism

Ising Model



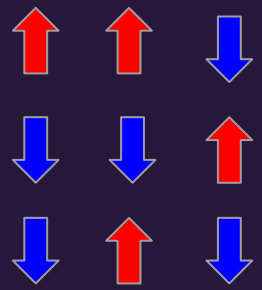


But in 4d!

Monte Carlo (Metropolis) Algorithm

1. Randomly pick a spin
2. Compute energy change ΔE
3. If ΔE < 0, flip
   a. Else, flip with probability exp(−ΔE/kT)
4. Repeat

$$\Delta E_i = -\sum \square J s_i s \square - h s_i$$



02 Approach

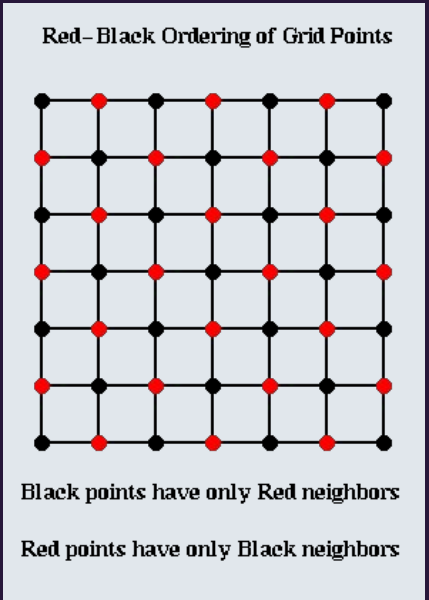
Non-parallel and parallel implementations

Sweeping Method vs Random Updates

Instead of randomly picking spins, sweep over the whole lattice and visit every spin

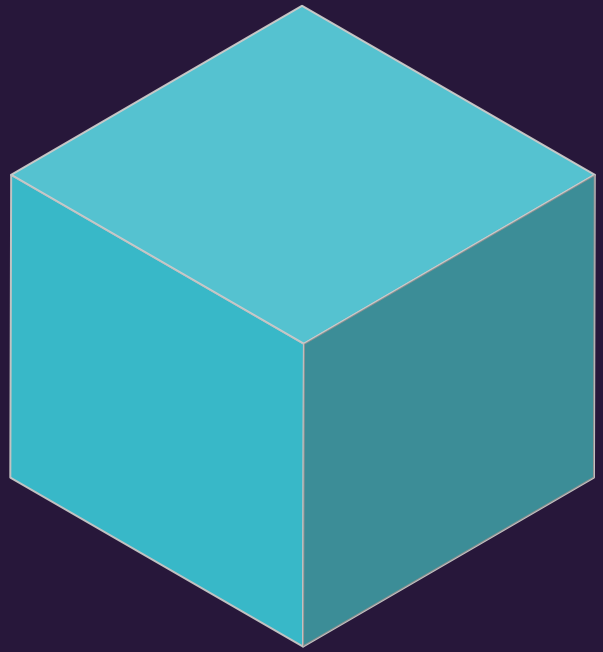
Less randomness, but still has proper statistical sampling and achieves equilibrium

Red-Black Updates

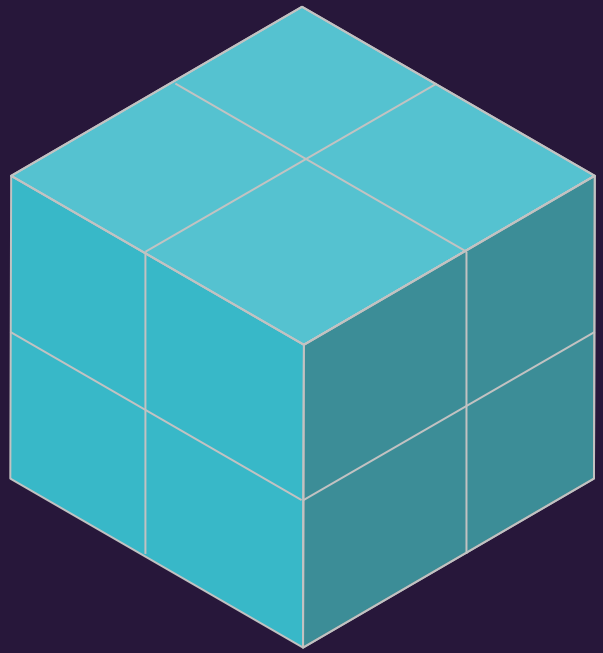


With sweeping, helps improve reliability and prevent pattern formation (see later)

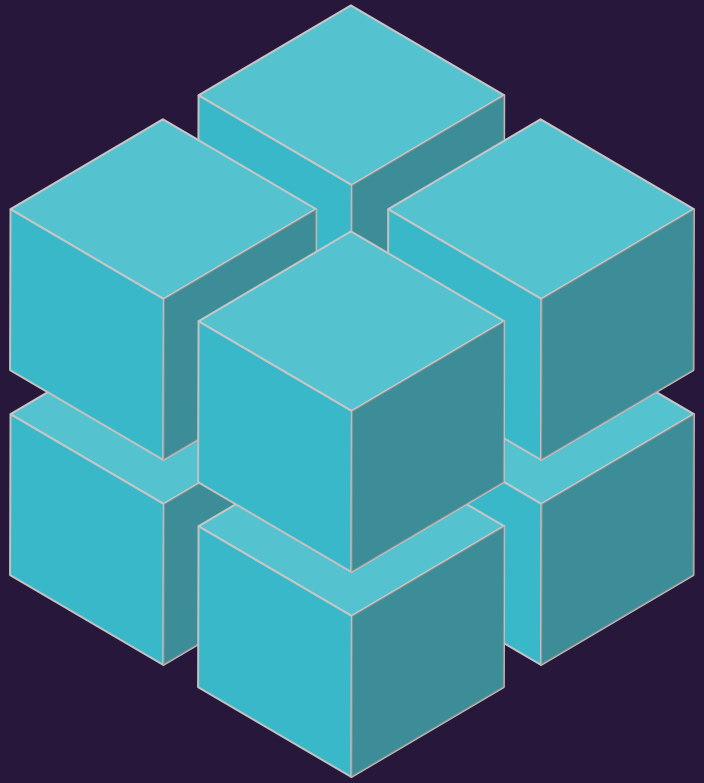
MPI Domain Decomposition



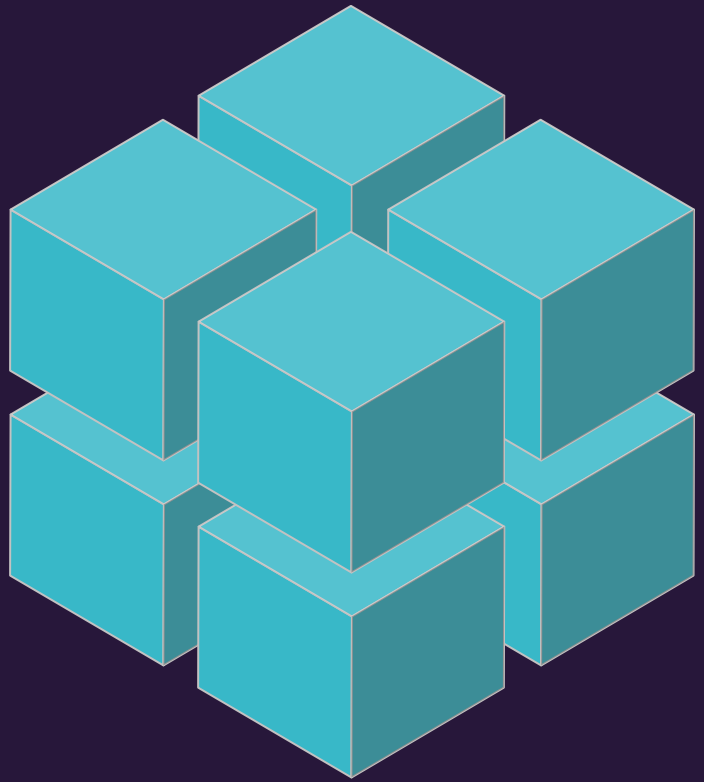
MPI Domain Decomposition



MPI Domain Decomposition

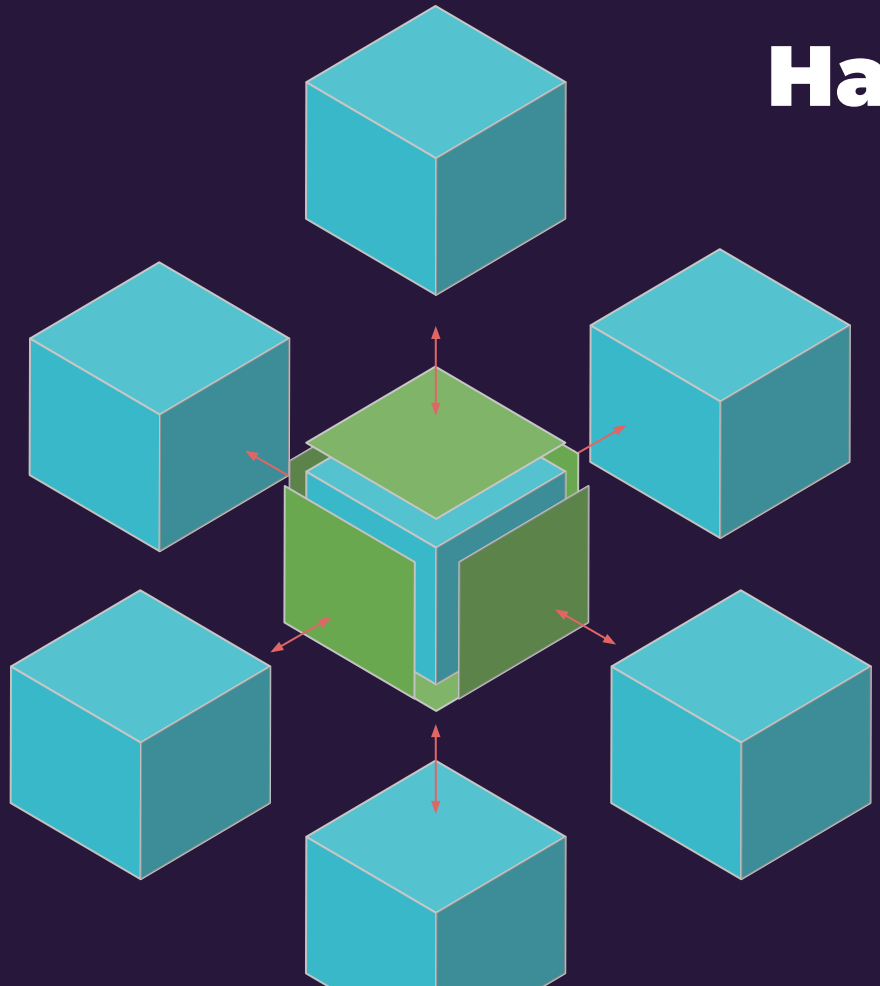


MPI Domain Decomposition



- Decompose LxLxL lattice into sub-lattices, each handled by an MPI rank
- Each rank stores a portion of the lattice plus halo (ghost) cells
- Halo exchanges ensure each process sees correct neighbor spins

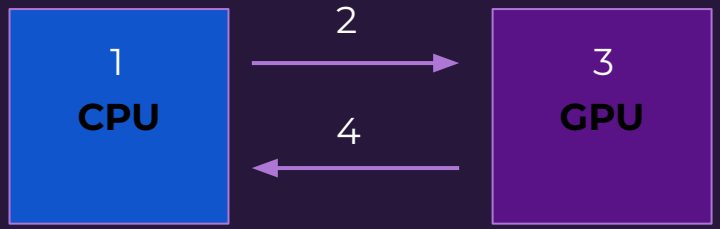
Halo Exchange



- Neighbor ranks exchange boundary data (planes)
- Synchronized before next update

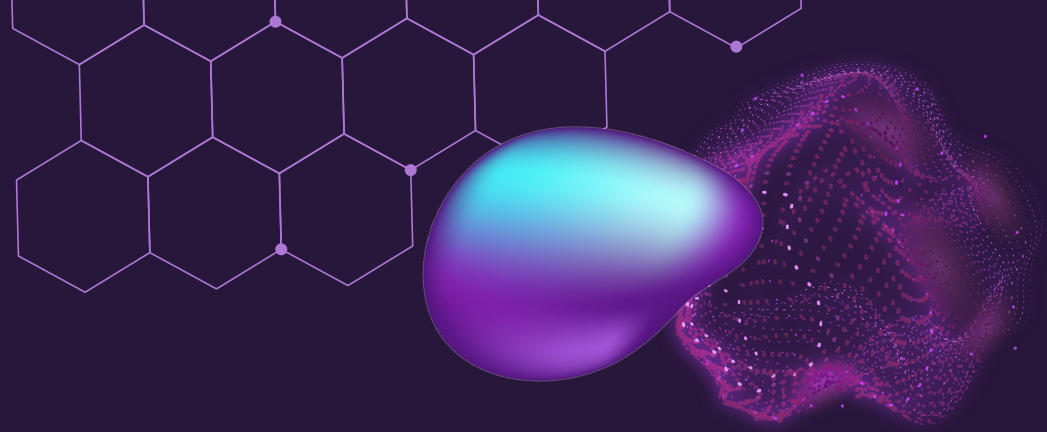
GPU Kernel

- Each MPI rank individually runs kernels
- Spin updates, energy and magnetism are offloaded to GPU
- (1) Halo is exchanged on CPU, (2) data is copied to GPU, (3) calculations performed, (4) then edges copied back
- Operations seen in rocprof

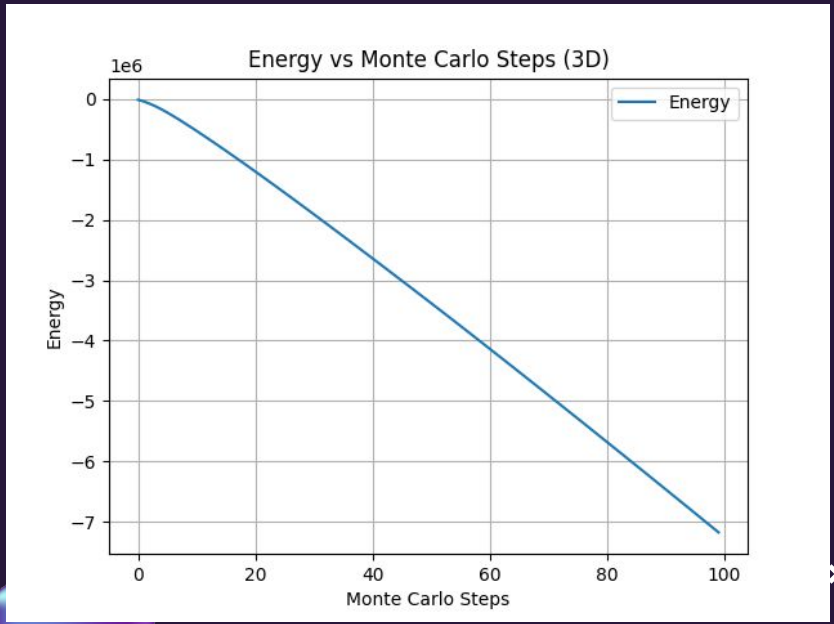


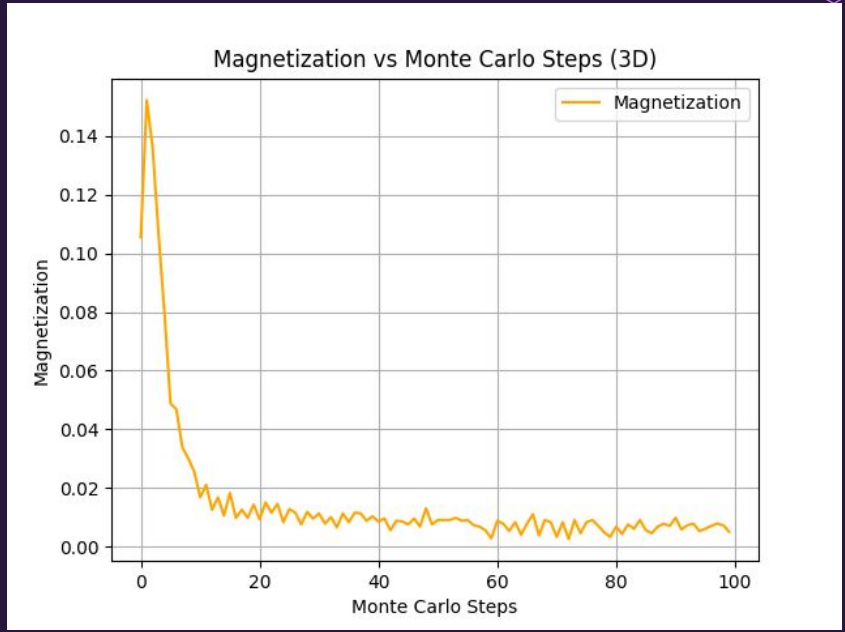
03 Results

Issues and Speedup



Serial Implementation



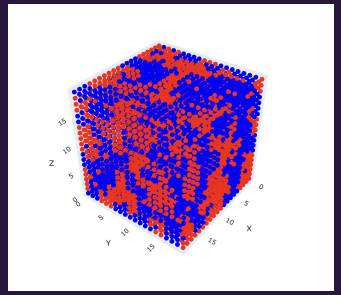


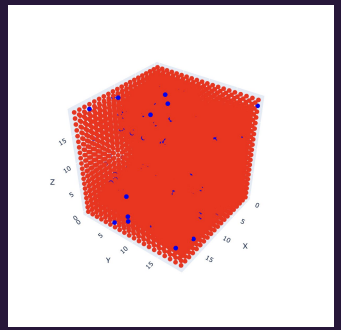
Serial Implementation

https://tajgillin.neocities.org/cpu/even

https://tajgillin.neocities.org/cpu/odd

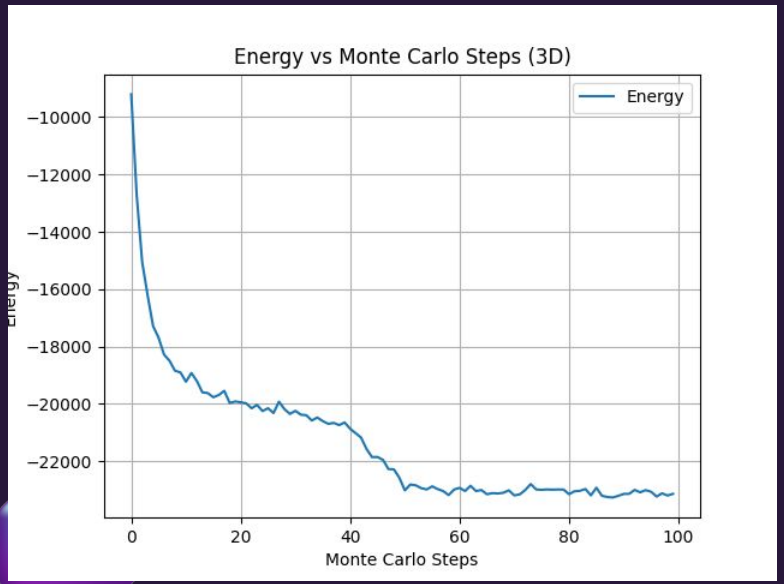
Red Black

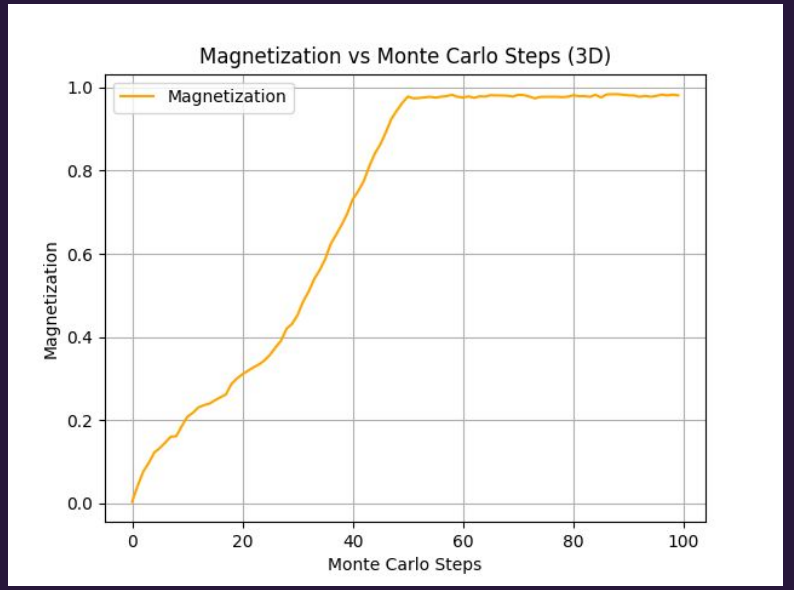




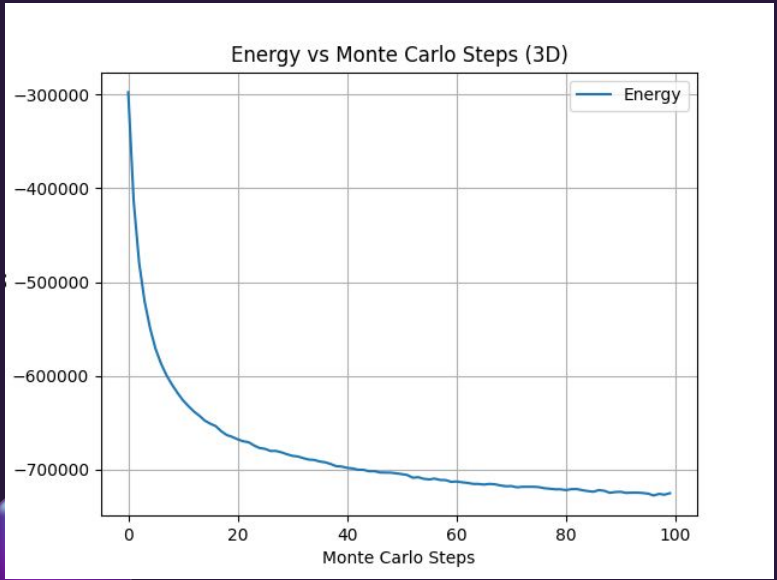
https://tajgillin.neocities.org/redblack/rb

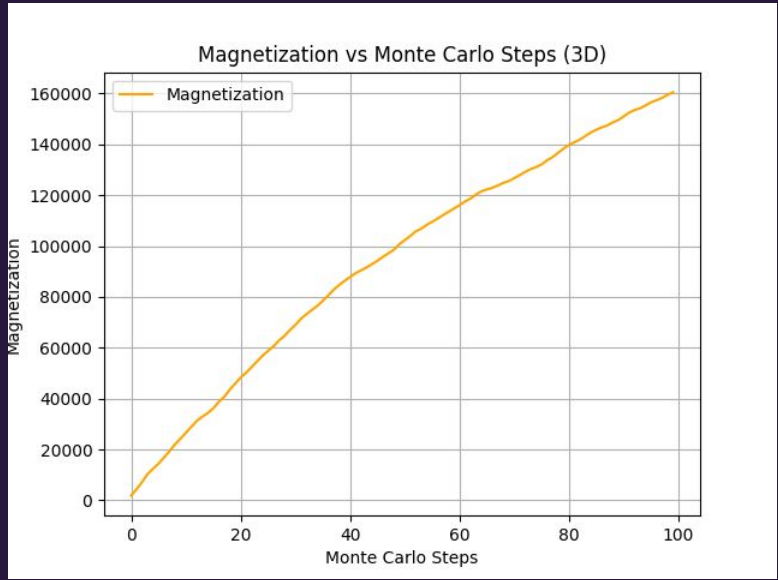
Serial Implementation w Red Black



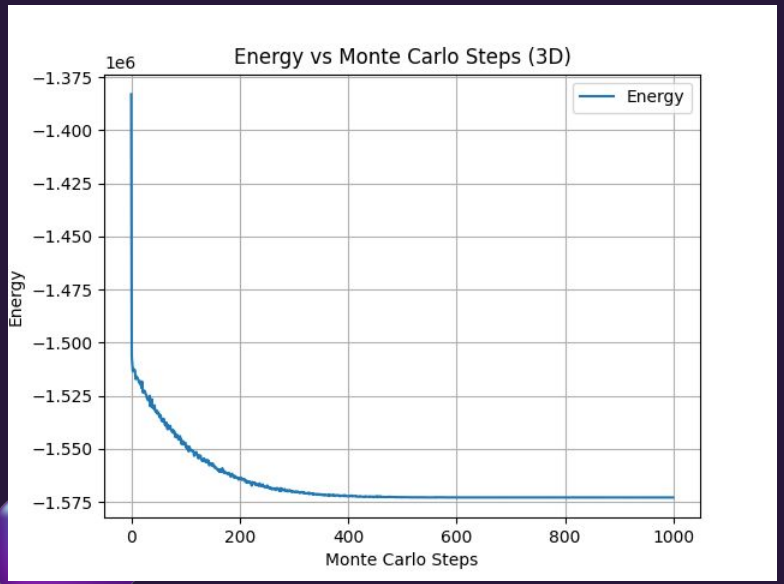


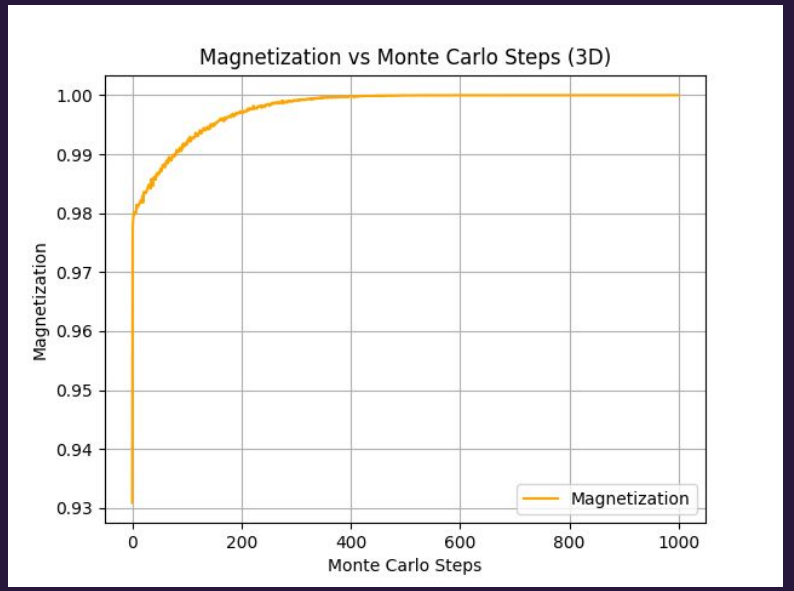
MPI w Red Black





MPI w GPU Kernels



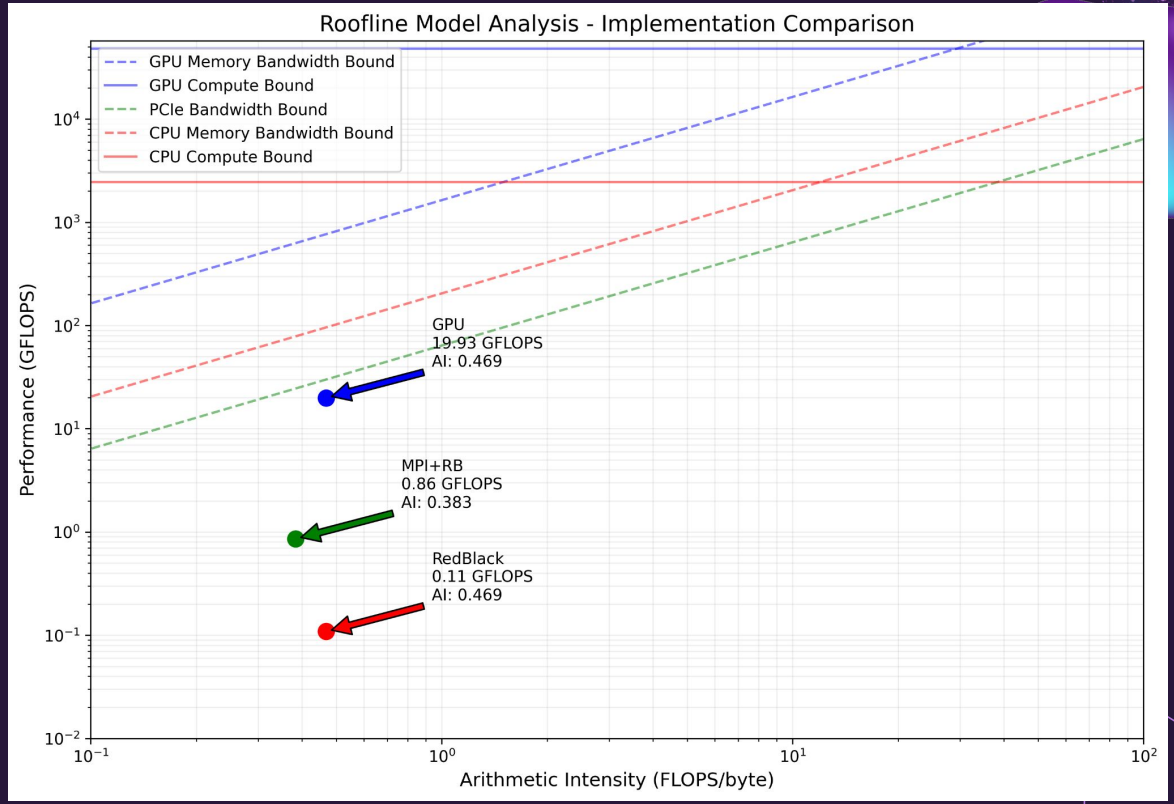


Comparison

64x64x64, 1000 steps, T = 2.5, J = 1

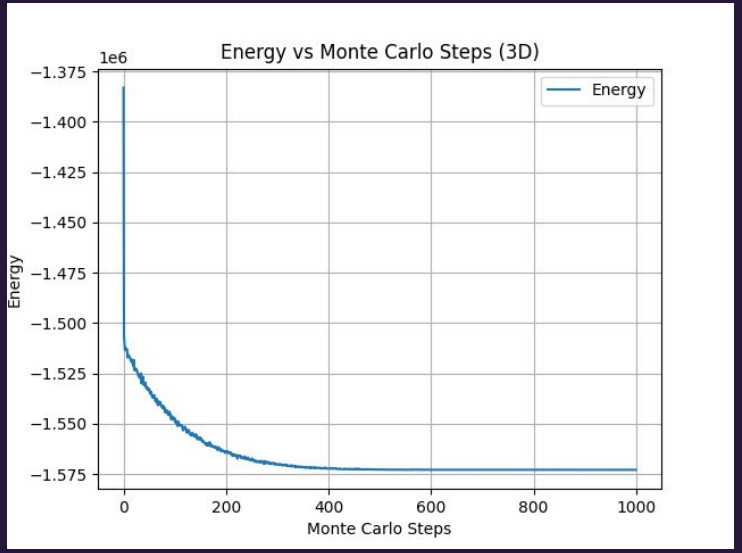
Serial	36.2 seconds
MPI v1	52 seconds
MPI v2 (1 rank)	35.6 seconds
MPI v2 (8 ranks)	5.8 seconds
GPU	0.95 seconds
GPU (256^3)	28.6 seconds

Roofline Model

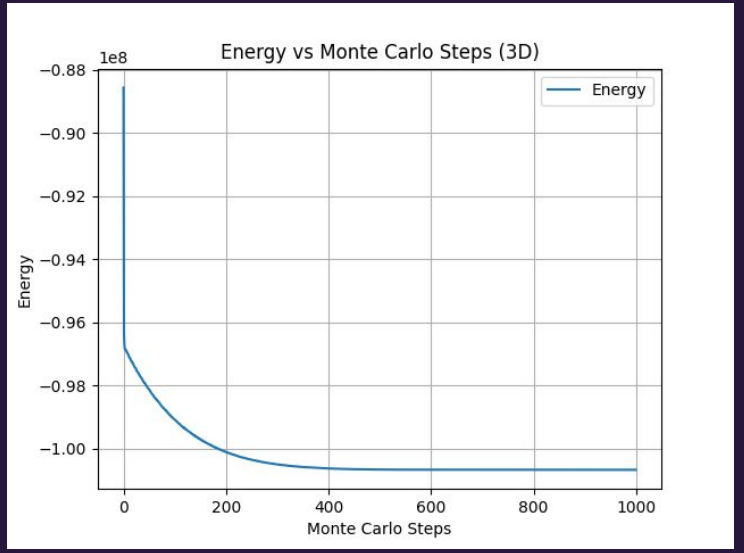


CPU: MI250X HBM2e
GPU: AMD EPYC 7V13

Scalability



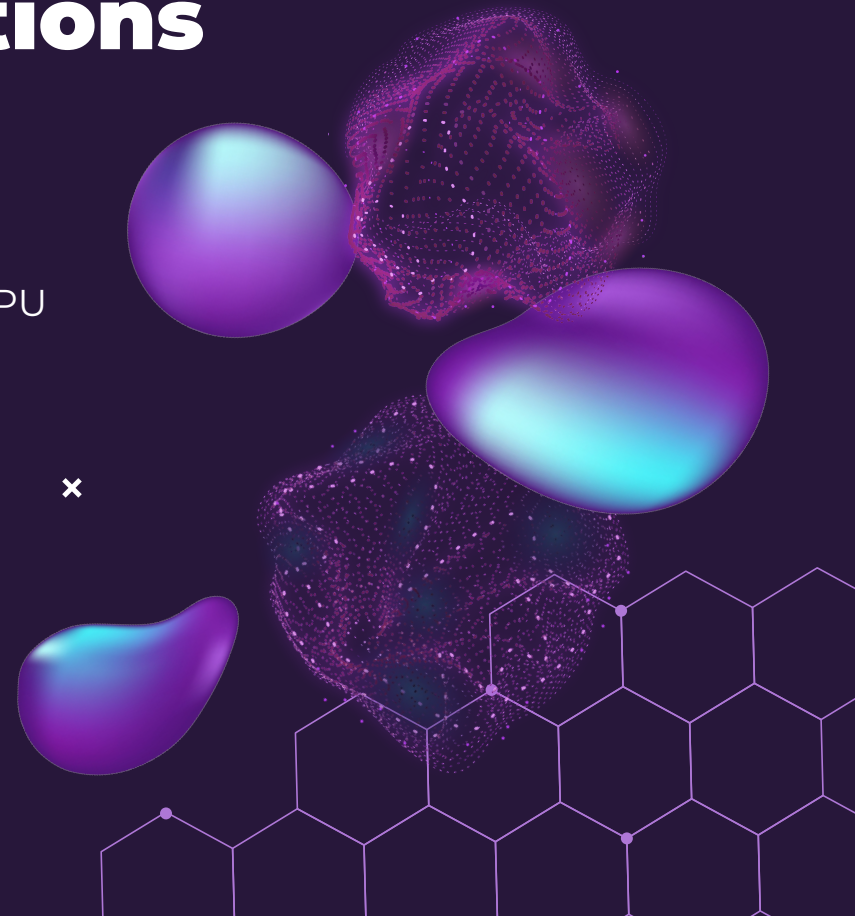
64^3 ~262k



256^3 ~17m

Optimizations

- Shared GPU memory (0.95 → 0.71 seconds) for threads within block with shared memory
- GPU aware halo process, share directly from GPU to GPU (→ 0.28 seconds)
- With 256^3, brought from 28.6 seconds to 5.6 seconds



Future Directions

- Pinned memory (currently using device), faster transfers and communication
- Streams, compute edges, then start halo exchange while computing inside

Thank you!