Real-Space Renormalization Group for One-Dimensional Quantum Ising Models: CPU and GPU Parallel Implementations

1. Project Overview

The goal of this project is to implement and benchmark a real-space renormalization group (RG) algorithm for simple one-dimensional quantum systems. I've already studied and python developed this algorithm for a Quantum Information and Computing assignment.

2. Physical Model

The project will consider a one-dimensional quantum model with local interactions, the transverse-field Ising model. The Hamiltonian consists of nearest-neighbor interactions and local terms, allowing a natural decomposition of the system into small, independent blocks.

3. Renormalization Group Algorithm

The real-space RG procedure is based on the following steps:

- The system is partitioned into blocks of a small number of sites.
- For each block, a local Hamiltonian matrix of small dimension is constructed.
- The block Hamiltonian is diagonalized, and the lowest-energy eigenstates are selected.
- The system is projected onto a reduced Hilbert space, defining renormalized degrees of freedom.
- Effective interaction parameters are computed and used to build a renormalized Hamiltonian.
  This procedure is iterated multiple times until the convergence of the energy density.

4. Sequential CPU Implementation

A reference sequential CPU implementation will be developed to:

- validate the correctness of the RG procedure,
- provide a baseline for performance comparisons,
- clearly expose the algorithmic structure and data dependencies.
  All RG steps, including block Hamiltonian construction, diagonalization, and projection, will be implemented explicitly.

5. Parallel CPU Implementation

A parallel CPU version will be developed using OpenMP.
The parallelization strategy exploits the independence of the RG blocks during the diagonalization and projection steps.

6. GPU Implementation

A GPU-accelerated version will be implemented using CUDA programming.
Each RG block will be assigned to an independent GPU thread or thread group, performing:

- construction of the local Hamiltonian,
- diagonalization of complete Hamiltonian,
- selection of low-energy states.

The idea is to maintain the Hamiltonian on the GPU for all the steps required in such a way to move data as less as possible.

7. Benchmarking and Performance Analysis

The performance of the sequential CPU, parallel CPU, and GPU implementations will be compared.
Benchmarks will include:

- execution time per RG iteration,
- speedup as a function of the number of blocks,
- scalability with system size and number of RG iterations.

8. Validation and Physical Interpretation

The RG flow of the effective model parameters will be qualitatively compared with known theoretical results.
This validation step ensures the physical consistency of the implementation without requiring highly complex numerical techniques.

9. Notes

- I will select small blocks in such a way to be able to perform exact diagonalization.
- I will use the Cloud Veneto resources available to perform both the CPU's and GPU's calculations