Presentation of Assignment 8

Quantum Information and Computing, A.Y. 2022/2023

Paolo Zinesi

09/01/2023

Theoretical introduction

We consider again a one-dimensional system with N $\frac{1}{2}$-spins, described by

$$\hat{H} = \lambda \sum_{i=1}^{N} \sigma_i^z + \sum_{i=1}^{N-1} \sigma_i^x \sigma_{i+1}^x .$$

Our goal is to find the ground state (g.s.) energy of a many-body quantum system described by the Hamiltonian $\hat{H}$.

In the previous assignment, it was not possible to exactly simulate systems with $N > 14$ because of memory limitations. With **approximate algorithms** we are able to simulate larger systems by performing some simplifications. We follow three approaches:

- **Mean Field**. The total wavefunction is factorized into a product of wavefunctions of each spin. The interaction between spins is usually oversimplified.
- **Real Space Renormalization Group (RSRG)**. Using a coarse-graining iterative procedure, at each step the system size is doubled and the total Hamiltonian is reduced to an effective Hamiltonian by keeping the lowest 2^N eigenenergies.
- **Infinite Density Matrix Renormalization Group (Infinite DMRG)**. Two spins are added at each iteration step and the total Hamiltonian is reduced to an effective one by keeping the largest m populations of the reduced density matrices.

Code development - Real Space RG

Algorithm:

1. Choose the number N of initial subsystems.
2. Initialize $\mathcal{H}_N^{(0)}$ with the original Hamiltonian and the interaction terms $\mathcal{A}_N^{(0)} = \mathbb{I}_{N-1} \otimes \sigma_x$ and $\mathcal{B}_N^{(0)} = \sigma_x \otimes \mathbb{I}_{N-1}$. Set $k = 0$.
3. Construct the matrix $\mathcal{H}_{2N}^{(k)} = \mathcal{H}_N^{(k)} \otimes \mathbb{I}_N + \mathbb{I}_N \otimes \mathcal{H}_N^{(k)} + \mathcal{A}_N^{(k)} \otimes \mathcal{B}_N^{(k)}$.
4. Diagonalize $\mathcal{H}_{2N}^{(k)}$ and construct the projector $\mathcal{P}^{(k)}$ that spans the subspace of the lowest 2^N eigenvalues.
5. Define matrices of the new iteration step as:
   - $\mathcal{H}_N^{(k+1)} \longleftarrow \frac{1}{2}\mathcal{P}^{(k)\dagger}\, \mathcal{H}_{2N}^{(k)}\, \mathcal{P}^{(k)} = \frac{1}{2}\mathrm{diag}\left(E_0, E_1, \ldots, E_{2^N-1}\right)$
   - $\mathcal{A}_N^{(k+1)} \longleftarrow \frac{1}{\sqrt{2}}\mathcal{P}^{(k)\dagger}\left(\mathbb{I}_N \otimes \mathcal{A}_N^{(k)}\right)\mathcal{P}^{(k)}$
   - $\mathcal{B}_N^{(k+1)} \longleftarrow \frac{1}{\sqrt{2}}\mathcal{P}^{(k)\dagger}\left(\mathcal{B}_N \otimes \mathbb{I}_N\right)\mathcal{P}^{(k)}$
6. Compute $\varepsilon^{(k+1)} = E_0/2N$
7. If stopping condition is not met, set $k \leftarrow k+1$ and go to step 3.

Notes

- The algorithm stops when a maximum number of iterations is reached or when the update in energy $\Delta\varepsilon^{(k)} = |\varepsilon^{(k)} - \varepsilon^{(k-1)}|$ goes below a given threshold value.
- In our implementation, we make use of the subroutines contained in ManyBodyUtils_mod.f90 to manage tensor product operations and of the LAPACK subroutines DSYEVX and DGEMM to perform, respectively, diagonalization and matrix-matrix multiplication.

Code development - Infinite Density Matrix RG

Algorithm:

1. Choose the number m_{max} of most populated states to keep at each iteration.
2. Initialize $\mathcal{H}_i^{(0)} = \sigma_i^z$ (for $i = 1, 2, 3, 4$) and $\mathcal{H}_{i,i+1}^{(0)} = \sigma_i^x \sigma_{i+1}^z$ (for $i = 1, 2, 3$). Set $m^{(0)} = 2$ and $k = 0$.
3. Set $m^{(k+1)} = \min(m_{max}, 2 \cdot m^{(k)})$.
4. Construct the matrix $\mathcal{H}^{(k)} = \sum_{i=1}^{4} \mathcal{H}_i^{(k)} + \sum_{i=1}^{3} \mathcal{H}_{i,i+1}^{(k)}$.
5. Find the ground state $|\Psi^{(k)}\rangle$ with energy $E_0^{(k)}$ of $\mathcal{H}^{(k)}$ and construct the density matrix $\rho^{(k)} = |\Psi^{(k)}\rangle\langle\Psi^{(k)}|$.
6. Compute $\varepsilon^{(k)} = E_0^{(k)}/(4 + 2k)$
7. Compute the reduced density matrices $\rho_L^{(k)} = \underset{\mathrm{R}}{\mathrm{Tr}}\left[\rho^{(k)}\right]$ and $\rho_R^{(k)} = \underset{\mathrm{L}}{\mathrm{Tr}}\left[\rho^{(k)}\right]$.
8. Diagonalize $\rho_L^{(k)}$ and construct the projector $\mathcal{P}_L^{(k)}$ that spans the subspace of the largest $m^{(k+1)}$ populations (i.e, the $m^{(k+1)}$ largest eigenvalues). Repeat for $\rho_R^{(k)}$.
9. Define matrices of the new iteration step as:
   - $\mathcal{H}_1^{(k+1)} \longleftarrow \mathcal{P}_L^{(k)\dagger} \left(\mathcal{H}_1^{(k)} \otimes \mathcal{I}_2 + \mathbb{I}_{m^{(k)}} \otimes \mathcal{H}_2^{(k)} + \mathcal{H}_{12}^{(k)} \right) \mathcal{P}_L^{(k)}$
   - $\mathcal{H}_2^{(k+1)} \longleftarrow \mathcal{H}_2^{(k)} = \lambda \sigma_z$
   - $\mathcal{H}_{12}^{(k+1)} \longleftarrow \left(\mathcal{P}_L^{(k)\dagger} \left(\mathbb{I}_{m^{(k)}} \otimes \sigma_x \right) \mathcal{P}_L^{(k)} \right) \otimes \sigma_x$
   - $\mathcal{H}_{23}^{(k+1)} \longleftarrow \mathcal{H}_{23}^{(k)} = \sigma_x \sigma_x$
   - Repeat similarly for $\mathcal{H}_3^{(k+1)}$, $\mathcal{H}_4^{(k+1)}$, $\mathcal{H}_{34}^{(k+1)}$ (with proper symmetrizations).
10. If stopping condition is not met, set $k \leftarrow k + 1$ and go to step 3.

Results - Real Space RG

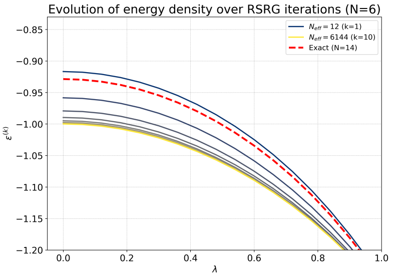

Figure 1: Evolution of $\varepsilon^{(k)}$.

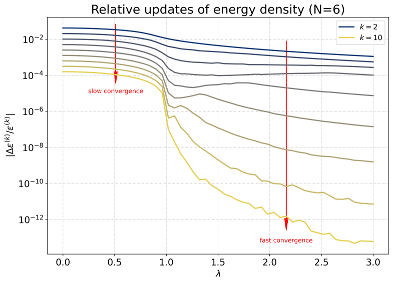

Figure 2: Relative updates of $\varepsilon^{(k)}$.

In **Figure 1** the evolution of the ground state (g.s.) energy density estimate is shown for different iteration steps. At each iteration the system of initial size N is doubled in size, so that the effective system dimension at k-th iteration is $N_{eff} = 2^k N$. The true g.s. energy density of a system with $N = 14$ is plotted in dashed lines for comparison, and is consistent with the RSRG results.

In **Figure 2** the updates of energy density are plotted as a function of λ. A difference between $\lambda < 1$ and $\lambda > 1$ is clearly visible in the convergence speed of the algorithm, suggesting a possible change of behaviour (maybe a phase transition?) of the energy density for $\boldsymbol{\lambda = 1}$.

Results - Infinite Density Matrix RG

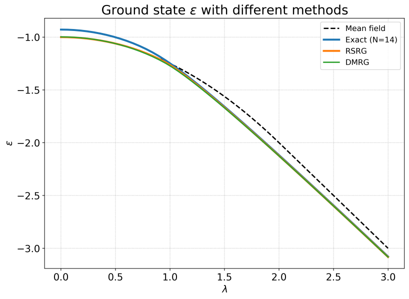

Figure 3: Computation of ε.

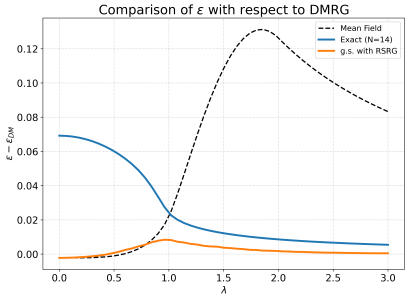

Figure 4: Absolute difference $\varepsilon - \varepsilon_{DM}$.

The ground state energy densities of RSRG and DMRG algorithms stay below the MF solution for $\lambda \gtrsim 0.7$. For $\lambda < 1$, the three solutions are comparable. The exact solution with $N = 14$ is plotted for comparison.

The Renormalization Group results presented in Figures 3,4 are obtained with a precision of 10^{-10} and are similar within a relative difference of 1%. The main difference is in the number of iterations for convergence: **$\mathbf{10-30}$ iterations for RSRG** with $N = 5$ (depending on λ) of duration $\Delta t = 0.5\,s$ each, and **$\mathbf{70-250}$ iterations for DMRG** with $m_{max} = 16$ and similar Δt. The RSRG algorithm converges quickly, but DMRG gives more accurate results for systems with boundary conditions.