DRSTP SPCTM school 2018
Introduction to tensor networks for QMBS

Exercise 3

This is an extra exercise, in case you are already done with exercise 2!

Problem 3.1 Repetition

Take some time to think about the following questions, to check if you have understood some of the key ideas.

1. What is the derivative of a tensor network with respect to a certain tensor in the network?
2. How can one perform a real-time evolution of an MPS?
3. In the energy minimization algorithm what type of equation do you need to solve to optimize one tensor, if the MPS is in mixed canonical form with respect to that tensor?
4. How can you get an MPO representation representing approximately $e^{-\tau \hat{H}}$ for a small time-step τ.

Problem 3.2 Infinite DMRG code

The goal of this exercise is to write an infinite DMRG code based on an MPO representation of the Hamiltonian.

1. Write a function which returns the 3 MPO tensors (left, middle, right) of transverse field Ising model:
$$\hat{H} = -\sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x. \tag{1}$$
Test the MPO construction by recomputing the Hamiltonian matrix of a 4-site system by multiplying the MPO tensors back together and reshaping upper legs together and lower legs together, respectively, to obtain a 16×16 matrix. Compute the lowest energy state using eigs. For $h = 1$ the lowest eigenvalue should be -4.758770.
2. Initialize H_L with the left MPO tensor, and H_R with the right MPO tensor.
3. Write a code which finds the new tensor T_m for two new sites added in the middle by solving the corresponding eigenvalue problem using eigs (cf. lecture), as in point 1 (you could actually reuse your code from there). Be careful with reshaping the legs of T_m. Note that due to roundoff errors you might need to symmetrize your matrix before using eigs (i.e. $H = (H + H')/2$).
4. Use an SVD to split the tensor T_m. Use the u and v tensors from the SVD to update the H_L and H_R parts. Reiterate points 3.-4. until convergence is reached (see below).

5. At each iteration k the corresponding eigen-energy is the total energy E_N of the system of length $N = 2+2*k$. Compute also the energy per site, E_s, and the energy per site in the center of the chain $E_{sc} = (E_N - E_{N-2})/2$. Check for convergence, e.g. in E_{sc}: if it drops below a certain tolerance (e.g. $\Delta E_{sc} < 10^{-7}$) then stop the simulation.

6. Test your code for $D = 8$, $h = 1.2$. You should obtain $E_{sc} = -1.419619$.

7. Check convergence of the energy as a function of the bond dimension, for $D = 2, 4, 6, ...20$ for $h = 0.8, h = 1, h = 1.2, h = 1.4$ by plotting the energy difference $E_D - E_{D=20}$ on a semi-logarithmic scale. What can you observe?

8. Compute the magnetization in the middle of the chain using the T_m tensor. The state typically does not break the symmetry spontaneously (because the system is still finite). To break the symmetry we can either apply a tiny external magnetic field or simply by fixing the boundary spins, e.g. to up. The latter can be achieved by projecting the physical degree of freedom in the initial H_L and H_R onto the up spin, i.e. multiplying it with matrices $diag([1, 0])$ on the physical legs on the boundary. Plot the magnetization as a function of h for different bond dimensions $D = 2, 4, 8, 12$ and observe the effect of a finite bond dimension on the order parameter, especially around the critical point $h = 1$. Note that to get the order parameter accurately close to the critical point you need to use a small tolerance, e.g. 10^{-10}.