Computational Tools for Quantum Many-Body Physics (WS 24/25)
Sheet 7 – Reading assignment

Jan Budich

Online: Mon, 27 Jan 2025
self paced

The purpose of this self paced reading assignment is to install and become familiar with the basic features of the C++ (also available for Julia) library ITensor. ITensor implements contraction of tensor diagrams in a quite user-friendly way, and offers already built-in routines for the DMRG optimization of finite matrix product states (MPS). Support in case of questions is provided by Tim Pokart (tim.pokart@tu-dresden.de) and Jan Budich (jan.budich@tu-dresden.de).

As a concrete case study, we consider a spin-1 chain model described by the Hamiltonian

$$H = \sum_j \left[\cos\theta \; \vec{S}_j \cdot \vec{S}_{j+1} + \sin\theta \; \left(\vec{S}_j \cdot \vec{S}_{j+1} \right)^2 \right] \tag{1}$$

where $\vec{S}_j = \left(S_j^x \;\; S_j^y \;\; S_j^z \right)$ are the spin-1 operators satisfying the angular momentum algebra $\left[S_j^\mu, S_{j'}^\nu \right] = i \, \delta_{j,j'} \epsilon^{\mu\nu\rho} S_j^\rho$ ($\hbar$ is set equal to 1) and the angle θ controls the strength and the sign of the bilinear and the biquadratic terms.

Hint: The ITensor libraries are well documented on the website `itensor.org`, where also various tutorials and sample codes for performing a basic DMRG simulation can be found.

1. **Installing and trying out ITensor**
   Download and compile the ITensor libraries. At `itensor.org` detailed information on the installation can be found. The dependencies of ITensor libraties are the BLAS/LAPACK libraries (`netlib.org`) or equivalently the Intel MKL libraries (`software.intel.com`), for highly optimized linear algebra routines. After the installation is completed, the following links to ITensor pages within the website are very useful:

   - A basic introduction to the notion of tensor diagrams and their implementation in ITensor is given in `itensor.org/Learn/The ITensor Book`. Read the first five chapters (links) which are fundamental to understand the building blocks of ITensor libraries.
   - To go a bit more in depth, take a look at the first four links in `itensor.org/Learn/Tutorials`, which again contain an introduction to the notion of tensor diagrams and their contraction, but supplemented by the practical implementation with ITensor.
   - A comprehensive documentation about the ITensor classes and the related routines is provided under `itensor.org/Learn/In-depth Documentation`.

   Furthermore, the installation folder of ITensor contains a directory `tutorial/` with working codes for several basic tasks. Take a look at the first two examples, in `tutorial/01_one_site/01_tutorial_code.cc` and `tutorial/02_two_site/02_tutorial_code.cc`, which implement the initialization of the state of one and two spin 1/2 site(s), respectively, and the calculation of local expectation values.

   Check that the ITensor libraries work properly by compiling and running the driver code `dmrg.cc` in the `sample/` directory found in the installation folder. This code calculates the ground-state of a nearest-neighbor Heisenberg model on a chain of either spin-1 or spin-1/2 sites. In the case of spin-1/2 sites, the result for a small chain can be checked against the exact diagonalization results from the fourth exercise sheet.

2. **Create the Hamiltonian as an MPO**
   Now, we aim at finding the ground state of the Hamiltonian (1) in the MPS variational space. The first step is to implement the Hamiltonian (1) in ITensor language. For the DMRG optimization, ITensor needs the Hamiltonian to be expressed as a matrix product operator (MPO). To get familiar with the concept of MPO, read the related section in [1] and the documentation at `itensor.org/Learn/Tutorials/Matrix Product Operators (MPO)`. The construction of an MPO representation of a given operator is performed in ITensor by the routine `AutoMPO` (related information at `itensor.org/Learn/Tutorials/Introduction to AutoMPO`) through which the practical implementation of the Hamiltonian in the code resembles hand-written notation.

Write out the single bilinear and biquadratic terms of Eq. (1) in terms of the raising/lowering operators S_j^+/S_j^-. Follow the example at `itensor.org/Learn/In-depth Documentation/DMRG`: Write an ITensor driver code that loads the `SiteSet SpinOne(N)`, a class containing the local spin-1 operators acting on the lattice sites, and implements the Hamiltonian (1) as a MPO.

3. **DMRG optimization from initial random MPS**
   Check that the Hamiltonian (1) commutes with $S_z = \sum_j S_j^z$. This means that the total magnetization M along z is a good quantum number. The `SiteSet SpinOne(N)` in ITensor already keeps track of this quantum number, but the initial MPS must have a well defined value of M in order to take full advantage of this symmetry in the DMRG optimization. Once the value of M has been specified (by the user), a random MPS belonging to the chosen magnetization sector may be initialized as a (random) product state in the S_j^z basis, where N_+ sites with spin $+1$, N_- sites with spin -1, and N_0 sites with spin 0, s.t. $N_+ + N_- + N_0 = N$ (N being the total number of sites) and $N_+ - N_- = M$, are randomly distributed over the chain. For MPS with well defined quantum numbers it is recommended to use the class `IQMPS` instead of `MPS`, to account for the sparsity of the quantum number blocks. Once the MPS is initialized, call the `dmrg` routine (see `itensor.org/Learn/In-depth Documentation/DMRG`) for performing the DMRG optimization of the state. Working driver codes for DMRG of different models can be also found in the directory `sample/` in the installation folder of ITensor. Read the tutorial at `itensor.org/Learn/Tutorials/Choosing DMRG Parameters` about proper choices of the DMRG parameters like number of sweeps, maximum bond dimension and truncation error, and `itensor.org/Learn/In-depth Documentation/Sweeps` on how to practically implement these choices.

   Compute the ground state energy for a chain with $N = 24$ sites for the following values of θ, corresponding to different phases of the model: $\theta = 0$ (Haldane anti-ferromagnetic phase), $\theta = 3\pi/8$ (quadrupolar phase), $\theta = \pi$ (ferromagnetic phase) and $\theta = 3\pi/2$ (dimer phase). Which magnetization sector does the ground state belong to in these four different cases? Check that the results are converged with respect to the number of sweeps and the maximum bond dimension used.

4. **(Bonus 1) Von Neumann entropy for chain bipartitions**
   From the ground states computed in the four different phases, calculate the von Neumann entanglement entropy

$$S_{\text{vN}}(\ell_A) = -\text{tr}\left(\rho_A \log \rho_A\right)$$

   of the subsystem A consisting in the first ℓ_A sites of the chain. Read the page at `itensor.org/Code Formulas/Compute entanglement entropy` to find the most efficient and compact way for calculating S_{vN}. Plot $S_{\text{vN}}(\ell_A)$ as a function of $\ell_A = 1, ..., N-1$, for the four different phases. Qualitatively, which differences can be observed between these four different values of θ?

5. **(Bonus 2) Correlation functions from MPS**
   Read the tutorials `itensor.org/Code Formulas/Measure a correlator from an MPS wavefunction` and `itensor.org/Tutorials/Calculating two-operator correlation functions` about how to efficiently calculate correlation functions of the form $\langle\psi|\mathcal{A}_i\mathcal{B}_j|\psi\rangle$ between two operators $\mathcal{A}_i$ and $\mathcal{B}_j$ acting on sites i and j respectively, over a MPS $|\psi\rangle$. For the four values of θ used, from the ground state MPS $|\psi_{\text{GS}}(\theta)\rangle$ obtained from DMRG calculate

$$C_{S^zS^z}(i,j) = \langle\psi_{\text{GS}}(\theta)|S_i^zS_j^z|\psi_{\text{GS}}(\theta)\rangle - \langle\psi_{\text{GS}}(\theta)|S_i^z|\psi_{\text{GS}}(\theta)\rangle\langle\psi_{\text{GS}}(\theta)|S_j^z|\psi_{\text{GS}}(\theta)\rangle$$
$$C_{S^+S^-}(i,j) = \langle\psi_{\text{GS}}(\theta)|S_i^+S_j^-|\psi_{\text{GS}}(\theta)\rangle - \langle\psi_{\text{GS}}(\theta)|S_i^+|\psi_{\text{GS}}(\theta)\rangle\langle\psi_{\text{GS}}(\theta)|S_j^-|\psi_{\text{GS}}(\theta)\rangle$$

   and plot these quantities as a function of $|i-j|$ by e.g. fixing $i = N/4$ and letting $j = i, ..., 3N/4$.

References

[1] U. Schollwoeck, *The density-matrix renormalization group in the age of matrix product states*, Annals of Physics **326**, 96 (2011).