DRSTP SPCTM school 2018
Introduction to tensor networks for QMBS

Exercise 4

Problem 4.1 Repetition

Take some time to think about the following questions.

1. Is it possible to use tensor network techniques to compute properties at finite temperature?
2. What scaling of the entanglement entropy can a MERA reproduce?
3. What is the maximal correlation length that an MPS / a MERA can represent (using a finite bond dimension)?
4. Imagine you have two subsystems A and B with n legs (indices) between A and B of bond dimension D. What is the maximal entanglement entropy between A and B?
5. Why can a MERA for a very large system be efficiently contracted?
6. Does every critical system in 1D have a log(L) scaling of the entanglement entropy?
7. Does every critical system in 2D have a L log(L) scaling of the entanglement entropy?
8. Draw one layer of the 3-to-1 MERA. What properties do the tensors fulfill?
9. Can an MPS be used for 2D wave functions at all?
10. What's the main problem when using an MPS for 2D systems?
11. Why do people typically use long cylinders for 2D MPS calculations?
12. What type of entanglement scaling does a PEPS reproduce?
13. What tensor network ansatz would you use on a large 2D honeycomb lattice?
14. What type of infinite PEPS ansatz would you choose to represent an antiferromagnetic state?
15. How does the computational cost scale as a function of system size to contract a PEPS in an exact way?
16. Describe one way to contract a 2D network in an approximate way (the main idea).
17. How can you represent the partition function of the 2D classical Ising model as a tensor network?
18. What's the main problem of using the simple iTEBD method (based on an SVD for the truncation of a bond) in 2D?

Problem 4.2 Corner-transfer matrix method for the 2D classical Ising model

The aim of this exercise is to get familiar with a powerful contraction method for 2D tensor networks: the corner-transfer matrix renormalization group method.[1] As an example we will consider the partition function of the ferromagnetic classical 2D Ising model which can be represented as a 2D tensor network. The Hamiltonian of the 2D Ising model reads

$$H = \sum_{\langle i,j \rangle} H_b(s_i, s_j) = -\sum_{\langle i,j \rangle} s_i s_j, \tag{1}$$

with $s_i \in \{+1, -1\}$ and where the sum goes over nearest neighbors. The partition function is given by

$$Z(\beta) = \sum_{\{c\}} \exp(-\beta H(c)) = \sum_{\{c\}} \prod_{\langle i,j \rangle} \exp(-\beta H_b(s_i, s_j)), \tag{2}$$

where $\beta = 1/T$ is the inverse temperature and where the sum goes over all possible spin configurations c. The exact solution of the order parameter (magnetization) is given by

$$m(\beta) = \frac{\sum_{\{c\}} s_r \exp(-\beta H(c))}{Z} = (1 - [\sinh(2\beta)]^{-4})^{1/8}, \tag{3}$$

for $\beta > \beta_c$, with critical value $\beta_c = \log(1+\sqrt{2})/2 \approx 0.44069$, and $m = 0$ for $\beta \leq \beta_c$. Here s_r denotes the spin on an arbitrary reference site r (since the system is translationally invariant).

Implement the following steps:

- Define the following tensor: $Q_{ss'} = \exp(-\beta H_b(s, s'))$. The partition function can be expressed by an infinite 2D tensor network made of the tensor:

$$a_{ijkl} = \sum_s \left(\sqrt{Q}\right)_{is} \left(\sqrt{Q}\right)_{js} \left(\sqrt{Q}\right)_{ks} \left(\sqrt{Q}\right)_{ls} \tag{4}$$

  The numerator of Eq.[3] can be expressed by a similar tensor network, except that the tensor on the reference site r is different:

$$b_{ijkl} = \sum_s s \left(\sqrt{Q}\right)_{is} \left(\sqrt{Q}\right)_{js} \left(\sqrt{Q}\right)_{ks} \left(\sqrt{Q}\right)_{ls} \tag{5}$$

  Construct these tensors (e.g. using ncon. Use *sqrtm* to compute the square root).

- Implement the corner-transfer matrix method (for a 2D isotropic tensor network) as discussed during the lecture. The boundary tensors can be initialized randomly (symmetrized appropriately). Note that thanks to the symmetry we only need to keep track of two different tensors: a corner tensor C and an edge tensor T. Note that before performing the SVD (or eig) symmetrize the matrix ($M = (M + M')/2$), since it might not be perfectly symmetric due to round-off errors. After each CTM iteration renormalize the boundary tensors by dividing them by their largest number (in magnitude). As a convergence check we can check how the (normalized) spectrum of singular values changes. If the change is below a certain tolerance, the CTM scheme has converged.

- Plot the order parameter m as a function of β in the vicinity of β_c for different values of χ (e.g. $\chi = 2, 4, 8, 16$) and compare your result with the exact solution.

[1] T. Nishino and K. Okunishi, J. Phys. Soc. Jpn. 65, 891894 (1996)

Problem 4.3 Corner-transfer matrix method for 2D quantum systems

Use your CTM code to evaluate the norm of a 2D quantum wave function (with translational and rotational invariance), where the a_{ijkl} tensors are obtained from contracting bra- and ket- iPEPS tensors. The provided function *get_symtensor(c)* can be used to obtain a symmetrized $D = 2$ iPEPS tensor A_p^{ijkl} (depending on 12 parameters c), where the physical index is the first index. As an example, we consider the transverse Ising model, which in addition to above Hamiltonian has a transverse magnetic field with magnitude λ. (You can use the provided function *get_H_trans_ising(lambda)*).

- Write a function to compute the 2-site reduced density matrix on two neighboring sites (don't forget to normalize it) using the (converged) environment tensors C and T, and the iPEPS tensor A. Use this function to compute the energy of your wave function, e.g. for $\lambda = 1$, given by some parameters c (you can try random c or play around with the parameters by hand).

- Let's attempt a brute-force optimization of the parameters c. This can be done, e.g. using the Matlab function *fmincon* (we can assume that all the parameters are within the interval $[-1, 1]$). Use this to compute the best $D = 2$ energy for $\lambda = 1, 2, 3, 4$ (the critical point is at $\lambda_c \approx 3.0444$).