# Advanced numerical methods in many-body physics

Exercise 9

---

**Problem 9.1 Imaginary time evolution algorithm with MPS**

The goal of this exercise is to implement an imaginary time evolution algorithm starting from a random initial MPS in order to obtain the ground state of a local Hamiltonian. The idea is to apply the local imaginary time evolution operator successively between all nearest-neighbor pairs, and then use an SVD to bring the time-evolved state back into MPS form, with the bond dimension truncated down to $D$. We can do this by sweeping from left to right and then back from right to left as shown in Fig. 6.11(a) in the lecture notes, corresponding to a second order scheme. The idea is to keep the MPS in mixed canonical form on the fly, by absorbing the singular values to the right during the right sweep and to the left during the left sweep. As an example we will consider the $S = 1/2$ Heisenberg model of a $N_s = 16$ site system and study the energy as a function of the bond dimension $D$ of the MPS and the imaginary time step $\tau$.
As a starting point you can make use of the code in *mps_energy.py* which computes the energy of an MPS based on an MPO. The MPS is stored in a list, with MPS tensors of size $d \times D \times D$, with the physical index being the first index, the left auxiliary leg the second index, and the right auxiliary leg the third index. Note we use an MPS tensor of size $2 \times 1 \times D$ tensor on the left boundary and a $2 \times D \times 1$ tensor at the right boundary, because this simplifies the ITE code if each tensor has the same number of legs.

1. Initialize a random MPS for the $N_s = 16$ site system and compute its initial energy (see *mps_energy.py*).

2. Initialize the 2-site imaginary time evolution gate $\exp(-\frac{\tau}{2}H_2)$, with $H_2$ the 2-site Heisenberg Hamiltonian term. You can do this by taking the matrix exponential (not element-wise exponential) of the 2-site Hamiltonian matrix, and by reshaping it into a $2 \times 2 \times 2 \times 2$ tensor. Be careful with the labelling of the legs (the correct order of the legs is 1: upper left, 2:upper right, 3: lower left, 4: lower right). As a check you can apply the gate subsequently to a 2-site wave function and it should converge to a singlet state.

3. Write a function which contracts two tensors with a two-body time-evolution operator (using *tensordot*, *einsum*, or *ncon*) and which performs an SVD, keeping $D$ singular values, with the option to absorb the singular values either on the left (used during the left sweep) or on the right (used during the right sweep). Be careful that you permute your resulting tensors after the SVD accordingly. Normalize the singular values by the norm. Note that if the wave function is in its Schmidt decomposed (mixed-canonical) form then the norm is simply computed by $norm = \sqrt{Tr(s * s)}$.

4. Next we use this function to apply the 2-site imaginary time evolution operator successively between all nearest-neighbor pairs. Write a function to perform a sweep from left-to-right, and then back from right-to-left. This will correspond to one time step $\tau$ (second order Trotter-Suzuki decomposition). Note that you can keep the MPS in the correct mixed-canonical form on-the-fly by absorbing the singular values to the right during the right sweep and to the left during the left sweep (initially it is not in canonical form, but it will be after the first complete sweep).

5. Call this function $M/2$ times, where $M$ is the maximal number of time steps (i.e. $\beta = M * \tau$). Check for convergence in the energy (you can recompute the energy, e.g., every 10 sweeps and stop the algorithm if it drops below a certain tolerance.)

6. Use your code to study the ground state energy for $N = 16$ as a function of $D$, for time steps $\tau = 0.1$, $\tau = 0.05$ and $\tau = 0.01$, and compare your results with the exact ground state energy $E_0 = -6.9117371455750956J$. To speed up convergence you can use the solution at smaller $D$ as an initial state for the simulations at larger $D$ instead of restarting from scratch (random tensors). Plot the deviation from the exact grounds state energy as a function of $1/D$.