Topological Matter School TMS'18 – San Sebastian (Spain) **31.08.2018**

Hands-on Session on Matrix Product States

Matteo Rizzi – Johannes Gutenberg University Mainz
(see also the theoretical lectures by F. Pollmann this morning)
matteo.rizzi@uni-mainz.de frank.pollmann@tum.de

0: Preliminaries

(a) Make sure you have Python installed on your laptop (at best version 2.7.*), and possibly a visual interface for it, so that you can launch iPython (e.g., Anaconda provides it). If you have never used python before, no worries, it will be easier than you imagine: keep only in mind that tabs have a precise meaning, so if you get an error, you most probably misaligned some lines ...

(b) Create a folder and copy inside the files you find in the Dropbox of TMS18: in particular, the file "svd_robust.py" provides a patch for some shortcomings of the native python routine for singular value decomposition. It relies on having some kind of numerical libraries installed (MKL, LAPACK, BLAS, ...) which you almost certainly have (e.g., if you have MatLab or other programs).

If you get error messages related to this later on, please tell it loud :-)

(c) We fix the following conventions for the labelling of tensors in the program:
- $B[i_s]$ will be the MPS tensor at site i_s (within the unit cell), and has axes (i.e., indices) $[i, a, b]$ = (physical, left virtual, right virtual);
- $S[i_s]$ are schmidt values between sites $(i_s, i_s + 1)$, and have therefore a single axis $[a]$ = virtual;
- $H_{\text{bond}}[i_s]$ is the bond hamiltonian between $(i_s, i_s + 1)$ with only physical axes $[i_1, i_2, j_1, j_2]$ = (out left, out right, in left, in right);
- $U_{\text{bond}}[i_s] = \exp(-\delta H_{\text{bond}}[i_s])$ will be the (imaginary) time evolution operator of a bond – just set $\delta \in \mathbb{R}^+$ or $\delta \in \mathrm{i}\,\mathbb{R}^+$ for your scopes;
- O will be an observable, defined on a site or on a bond, therefore having only physical axes $[i, j]$ = (phys. out, phys. in) or $[i_1, i_2, j_1, j_2]$ (as H_{bond} above);
- W will be the MPO tensor, and has axes $[a, b, i, j]$ = (virtual left, virtual right, physical out, physical in) – in principle it could also depend on the site, but let us neglect it for the moment, since we use it only for observables here (and not to deal with Hamiltonians);

(d) As you might have noticed, most files are only templates with a skeleton of the needed routines, so we have to build a working code together – yes, "hands-on"!

1: Implementing the basic operations & testing them on Ising model

(a) EXERCISE: Open the file "itebd_functions_template.py" and go to the routine "update". Here we want to apply the time-evolution operator U (in its Trotter form) on two sites of the chain at a time (even/odd links): thus, we need to contract U together with the MPS tensors $B[j]$ and $B[j+1]$ (without forgetting the Schmidt values $s[j]$) and then perform a SVD to compress the information again. [If you wonder what it does mean, recall the examples of this morning.]

(b) EXERCISE: Then move to the routines "*_expectation_value" (*= "site" and "bond") to implement the couple of simple contractions needed to measure observables out of the MPS state. Same in the routine "correlation_length", where you need to compute the transfer matrix and get its eigenvalues. Now we are ready to test our algorithm onto a physical model!

(c) Open the iPython notebook "groundstate_ising.ipynb" by typing "ipython notebook" on your shell. If you do not have such a graphical interface installed, then simply open "groundstate_ising.py" with an editor of your choice, and you will later run it by "./groundstate_ising.py" on your shell :-)

(d) Have a look at the general structure of the code: it simply sets the simulation parameters, define the spin operators and the Ising Hamiltonian, and then perform a long imaginary time evolution, which exponentially suppresses excited states and lead towards the ground state wavefunction. Finally, it measures entanglement entropy and spectrum, local observables and the (slowest) correlation length.

(e) EXERCISE: Pick up a coupling parameter g, say between 0 and 2.
Produce plots showing how fast the energy converges to its exact value as a function of imaginary time (and of bond dimension χ).
Plot the entanglement spectrum (i.e., the Schmidt values) at some chosen coupling: does it decay exponentially as predicted in the morning lecture? What happens around $g = 1$?
Finally plot magnetization, entanglement entropy of the semi-infinite chain, correlation length, etc. as a function of the coupling parameter g.

(f) EXERCISE*: If you were fast with the above, and you feel fit with MPS, then implement a subroutine that computes explicitly correlation functions of two given observables at an arbitrary distance.
Moreover, you could also compute the entanglement entropy of a open interval of a given length ℓ: how is its density matrix related to the transfer matrix you computed above for obtaining the correlation length?

2: A SPT example: Haldane phase in $S = 1$ chains

Now let us move to something topological, namely take a 1D spin-1 chain undergoing a bilinear-biquadratic model of the form $H = \cos(\theta) \sum_j \vec{S}_j \cdot \vec{S}_{j+1} + \sin(\theta) \sum_j \left(\vec{S}_j \cdot \vec{S}_{j+1} \right)^2$. At $\theta = 0$, this is simply the usual Heisenberg antiferromagnetic (AFM) model, while at $\theta = \arctan(1/3)$ it provides the paradigmatic exactly solvable AKLT point.

(a) EXERCISE: Repeat what you did for the Ising model before. Now, however, give a deeper look to the entanglement spectrum for couplings in the region $\theta \in [-\pi/4, \pi/4]$ around the Heisenberg AFM: do you notice any pattern? And outside this range? Can you relate this to what explained in the morning lecture about SPT phases?

(b) EXERCISE*: Take the routine about correlation length, where the standard transfer matrix was computed, and modify it (after having duplicated it!) to compute the mixed transfer matrix for the operators $U_x = \exp -i\pi S_x$ and $U_z = \exp -i\pi S_z$ defining the $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry protecting this topological phase. As explained in the lecture, the dominant eigenvectors V_x, V_z are the representation of the symmetry operators acting on the virtual links. Perform $\chi \operatorname{Tr} \left(V_x V_z V_x^\dagger V_z^\dagger \right)$ to get the phase factor related to the $\mathbb{Z}_2 \times \mathbb{Z}_2$ projective representation, as explained in the lecture this morning.

3: Pumping in the Rice-Mele model

Finally, let us consider the prototypical Rice-Mele model (i.e., the Su-Schrieffer-Heger model plus staggered chemical potential):
$H = t \sum_j \left(c_{j,A}^\dagger c_{j,B} + \text{h.c.} \right) + t' \sum_j \left(c_{j,B}^\dagger c_{j+1,A} + \text{h.c.} \right) + u \sum_j \left(c_{j,A}^\dagger c_{j,A} - c_{j,B}^\dagger c_{j,B} \right)$.
Notice that we can represent the fermionic operators with on-site spin matrices, as long as we are considering only local and nearest-neighbour terms – what should we instead do for longer range terms?

We want to perform a pumping cycle of the kind $t = 1,\ t' = t_0 + \cos\phi,\ u = \sin\phi$, where ϕ is a parameter going from 0 to 2π, and see whether we can spot the topological regime of the underlying SSH model ($u = 0$) by looking at the pumped charge per cycle.

We need to invoke the files "itebd_pump.py" and "pump_ricemele.py" now.

(a) EXERCISE: First go around the cycle in a quasi-static way, i.e., compute the ground state for different ϕ's starting from a close-by solution (iteratively apply iTEBD with imaginary time without re-initialising B, s). Use the routine called "n_right" to extract the amount of extra charge present in the right hand side of the chain (with respect to the uniform average background – ref. in file).
Plot n_r as a function of ϕ for different values of t_0, say smaller or greater than 2: what do you notice? Can you relate it to some theoretical analysis of the model?

(b) EXERCISE: Do the same as above with real-time evolution: what happens if you are impatient and set a too short total time for the pumping cycle?

(c) EXERCISE*: Now add a interacting term $H_\text{I} = V \sum_j (n_{j,A} n_{j,B} + n_{j,B} n_{j+1,A})$, and redo all the above: is the topological regime stable? Is there a critical V?

This is, of course, only a very basic example: if you instead use a model describing a fractional Chern insulator, as discussed briefly in the lecture, you will be able to see fractional pumping (i.e., more cycles to pump a single charge).