FOR 1807 Winter School 2018 Marburg

Exact Diagonalization hands-on session

Exercise sheet

Alexander Wietek, Andreas Honecker

In the following exercises we will make use of the scripts `hamiltonian_tfi.py`, `hamiltonian_hb_staggered.py` and `hamiltonian_hb_xxz.py`. The three files contain each contain a function `get_hamiltonian_sparse` which creates the Hamiltonian matrix in a sparse matrix format. The following Hamiltonians are given

- `hamiltonian_tfi.py`: The *Transverse Field Ising* model,

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - h_x \sum_i \sigma^x, \tag{1}$$

  where σ^z and σ^x are the Pauli matrices, on a one-dimensional chain lattice (with periodic boundary conditions). This model does not conserve total S^z. Momentum conservation is not implemented.

- `hamiltonian_hb_staggered.py`: The *Heisenberg model in a staggered magnetic field* ,

$$H = J \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j + h_S \sum_i (-1)^i S^z, \tag{2}$$

  one-dimensional chain lattice (with periodic boundary conditions). This model does conserve total S^z. Momentum conservation is not implemented.

- `hamiltonian_hb_xxz.py`: The *Heisenberg model with Ising anisotropy* ,

$$H = J \sum_{\langle i,j \rangle} \left[S_i^z S_j^z + (1+\Delta) \left(S_i^x S_j^x + S_i^y S_j^y \right) \right] \tag{3}$$

  one-dimensional chain lattice (with periodic boundary conditions). This model does conserve total S^z. Momentum conservation is implemented.

An example how to use these functions for computations can be found in the file `example_groundstate_energy.py`. Internally, spin states are encoded in a binary representation, for example

$$|\uparrow\downarrow\uparrow\uparrow\rangle = (1011)_2 = 11. \tag{4}$$

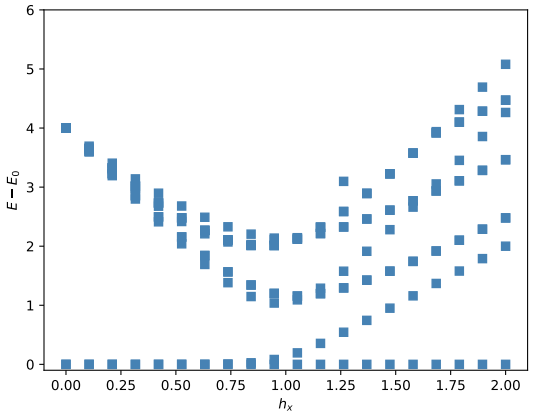
The scripts are kept short, to be easily accesible. Please just ask if you have any questions how they work.

Have fun with the exercises!

Exercise 1: Impementing a Hamiltonian, computing static correlations (basic)

(a) We consider the transverse field Ising model as defined by Eq. 1. Use the script `hamiltonian_tfi.py` to compute lowest 10 eigenvalues for $J = 1$ and for a range of values h_x from 0 to 2 for a chain of length $L = 12$. Plot the excitation energies as a function of h_x. The model has a phase transition from a ferromagnetic phase to a paramagnetic phase at $h_x = 1$. Can you detect the phase transition by investigating the energy spectra?

Result: The energies as a function of the transverse field are shown in the following plot:



Hints:

- Use the function `get_hamiltonian_sparse()` to create the sparse matrix of the Hamiltonian for a given h_x
- Create a scipy sparse matrix and compute low lying eigenvalues as shown in `example_groundstate_energy.py`.
- Perform the above steps for multiply values of h_x and plot the results using matplotlib.

(b) Alter the script `hamiltonian_tfi.py` to create the Hamiltonian of the spin-1/2 Heisenberg model,

$$H = J \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j, \tag{5}$$

and compute its ground state energy for a chain of length $L = 12$. Cross check the ground state energy with the script `hamiltonian_hb_staggered.py` for total $S^z = 0$.

As a supplement, investigate the behaviour of the ground state energy as a function of the system size L and check out the convergence to

$$E_0/L = -\log 2 + 1/4.$$

Result: The ground state energy for $L = 12$ as described above is given by

$$E_0/L = -0.44894924$$

Hints:

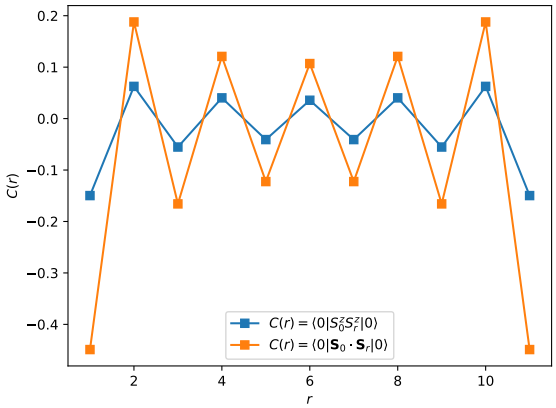
- We have to modify `get_hamiltonian_sparse()` in the file `hamiltonian_tfi.py`
- Replace lines 48-66 in this file by the implementation of the Heisenberg bond in the file `hamiltonian_hb_staggered.py`, lines 79-104 (omitting the staggered field).

(c) Compute the ground state spin correlations $\langle 0|S_0^z S_r^z|0\rangle$ and $\langle 0|\mathbf{S}_0 \cdot \mathbf{S}_r|0\rangle$ of the spin-1/2 Heisenberg model in Eq. 5 for a chain of length $L = 12$. Check, whether

$$\langle 0|\mathbf{S}_0 \cdot \mathbf{S}_r|0\rangle = 3\,\langle 0|S_0^z S_r^z|0\rangle\,.$$

This equality holds due to SU(2) rotational symmetry of the ground state $|0\rangle$.

Result: The ground state correlations as descibed above are shown in the following plot:



Hints:

- Create matrices for the operators $S_0^z S_r^z$ and $\mathbf{S}_0 \cdot \mathbf{S}_r$. This can be done analogously to creating the Hamiltonian.
- Use scipy and numpy multiplication routines to evaluate $\langle 0|S_0^z S_r^z|0\rangle$ and $\langle 0|\mathbf{S}_0 \cdot \mathbf{S}_r|0\rangle$ for different values of r.

Exercise 2: Hamiltonian symmetries (advanced)

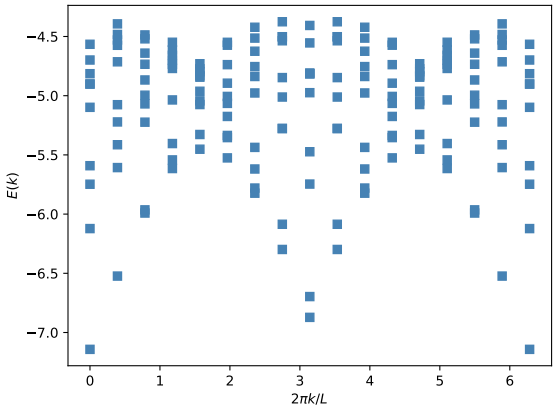
In this exercise we investigate the Heisenberg XXZ model Eq. 3. This model has a discrete translational symmetry $T : |\sigma_1, \sigma_2, \ldots, \sigma_L\rangle \mapsto |\sigma_L, \sigma_1, \ldots, \sigma_{L-1}\rangle$. Hence, a symmetrized basis with fixed momentum $2\pi k/L$ can be chosen. The script `hamiltonian_hb_xxz.py` creates the Hamiltonian of the Heisenberg model with Ising anisotropy as defined in Eq. 3 in the basis of momentum eigenstates.

$$|\boldsymbol{\sigma}^k\rangle = \frac{1}{N_{\boldsymbol{\sigma}}} \sum_{n=0}^{L-1} \mathrm{e}^{i2\pi kn/L} T^n |\boldsymbol{\sigma}\rangle ,$$

where $N_{\boldsymbol{\sigma}}$ denotes the normalization constant of the symmetrized state $|\boldsymbol{\sigma}^k\rangle$. k can be chosen as a parameter and acts as a label for the different momentum sectors.

1. Compute the lowest 10 eigenvalues of the Hamiltonian for each $k = 0, \ldots, L = 16$ and plot the energies as a function of momentum for the isotropic Heisenberg case.

   Result: The momentum resoved eigenvalues for $L = 16$ are shown in the following plot:



   Hints:

   - Use the function `get_hamiltonian_sparse()` to create the sparse matrix of the Hamiltonian for a given momentum k
   - Create a scipy sparse matrix and compute low lying eigenvalues as shown in `example_groundstate_energy.py`.
   - Perform the above steps for multiply values of k and plot the results using matplotlib. Notice that the actual momentum is given by $2\pi k/L$
   - If you want to understand in detail, how the symmetrized basis for translational symmetry is built up, have a look at the implementation `hamiltonian_hb_xxz.py` and Weiße and Fehske [2007].

2. We are now going to implement a spinflip symmetry for the transverse Field Ising model in Eq. 1. The spinflip symmetry flips all spins at all sites, $S : |\sigma_1, \sigma_2, \ldots, \sigma_L\rangle \mapsto |-\sigma_1, -\sigma_2, \ldots, -\sigma_L\rangle$. Check that this operator commutes with the Hamiltonian, $[H, S]$. The spinflip symmetry is simpler to utilize than translational symmetries.

   Evaluating matrix elements in the symmetrized basis: We will now describe how to evaluate the matrix elements in the spinflip symmetrized basis. The symmetrized basis states given by

$$|\boldsymbol{\sigma}^\gamma\rangle = \frac{1}{\sqrt{2}} \left(|\boldsymbol{\sigma}\rangle + \gamma S |\boldsymbol{\sigma}\rangle\right) .$$

$|\boldsymbol{\sigma}^\gamma\rangle$ denote the symmetrized states in the even ($\gamma = 1$) and odd ($\gamma = -1$) representation of the spinflip symmetry. Our goal for now is to evaluate matrix elements of the form

$$\langle \boldsymbol{\tau}^\gamma | H_k | \boldsymbol{\sigma}^\gamma \rangle \,.$$

To represent the symmetrized state on the computer we choose $|\tilde{\boldsymbol{\sigma}}\rangle = \min(|\boldsymbol{\sigma}\rangle, S\,|\boldsymbol{\sigma}\rangle)$ and call this state a representative. If we apply an offdiagonal, non-branching term (i.e. one spin configuration is only mapped to one single other configuration),

$$H_k\,|\tilde{\boldsymbol{\sigma}}\rangle = c_k\,|\boldsymbol{\tau}\rangle\,,$$

the state $|\boldsymbol{\tau}\rangle$ need not necessarily be a representative. Put differntly, $S\,|\boldsymbol{\tau}\rangle$ could be smaller than $|\boldsymbol{\tau}\rangle$. Let again $H_k\,|\tilde{\boldsymbol{\sigma}}\rangle = c_k\,|\boldsymbol{\tau}\rangle$. The matrix element between two representatives is given by

$$\langle \tilde{\boldsymbol{\tau}}^\gamma | H_k | \tilde{\boldsymbol{\sigma}}^\gamma \rangle = \begin{cases} c_k \text{ if } |\boldsymbol{\tau}\rangle = |\tilde{\boldsymbol{\tau}}\rangle \\ \gamma c_k \text{ if } |\boldsymbol{\tau}\rangle = S\,|\tilde{\boldsymbol{\tau}}\rangle \end{cases}$$

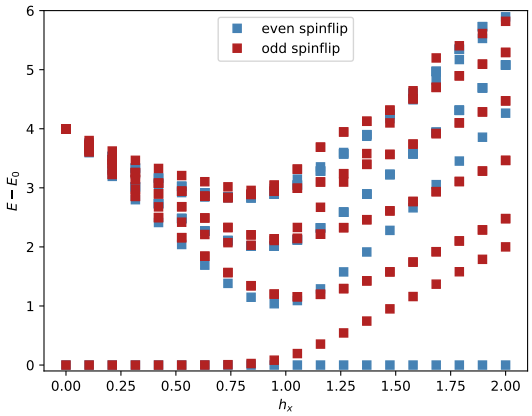
Check this equation!

How to implenent spinflip symmetry, step by step: Spinflip symmetry can be implemented in a manner analogous to the implementation of translational symmetry in `hamiltonian_hb_xxz.py`.

- Start with the function `get_hamiltonian_sparse()` from `hamiltonian_tfi.py` and add an additional parameter `gamma` which should be ± 1, depending whether we choose the even/odd representation.
- define a function `flip()` that flips all the spins for a given spin configuration
- define a function `get_representative()` that, for a given state $|\boldsymbol{\sigma}\rangle$ returns $|\boldsymbol{\sigma}\rangle$ if $|\boldsymbol{\sigma}\rangle < S\,|\boldsymbol{\sigma}\rangle$ and $S\,|\boldsymbol{\sigma}\rangle$ else. You can use the function `flip()`. This function should also return as a second value `gamma` if the flipped state is returned and 1 else.
- We need to create a list of representatives. To do so run through all spin configurations and save those who are representatives to a list.
- Instead of looping over all spin configurations, we now only loop over the representatives to create the Hamiltonian. This yields the dimensional reduction.
- The evaluation of diagonal elements remains unaltered.
- The off-diagonal elements need to be computed differently. After the transverse field bond is applied, we need to find the representative of this state. Use the function `get_representative()` to do so. This function should also return `gamma` if the state has been flipped.
- The coefficient is now given by `gamma` $\cdot\, h_x$. The column and row indices are determined by the location of both states in the list of representatives.

Follwing these instructions, we can now compute eigenvalues of the Hamiltonian in the even and odd spinflip sector. Compute the lowest 10 eigenvalues for a range $h_x = 0, \ldots, 2$ and both symmetry sectors. Plot the excitation energies and compare to exercise 1a). Why are the even and the odd sector degenerate in the ferromagnetic case?

Result: The excitation energies for the even/odd symmetry sectors are shown in the following figure:



References

Erik Koch. The lanczos method. *The LDA+ DMFT approach to strongly correlated materials*, 2011. `https://www.cond-mat.de/events/correl11/manuscripts/koch.pdf`.

Alexander Weiße and Holger Fehske. Exact diagonalization techniques, lecture notes in physics. 2007. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.627.8371&rep=rep1&type=pdf`.