Dr. Kiryl Pakrouski ETH Zürich
Department of Physics
Date: 15.03.2022

Computational Quantum Physics
Exercise Sheet 3

Responsible TA: Karin Sim, simkarin@phys.ethz.ch

Exercise 1. Transverse field Heisenberg model: exact diagonalisation
Consider the transverse field Ising model given by the Hamiltonian

$$\hat{H} = \sum_{i=0}^{N-1} \sigma_i^z \sigma_{i+1}^z + h^x \sum_i \sigma_i^x. \tag{1}$$

In this exercise we impose periodic boundary conditions. You will need to perform the exact diagonalisation and plot the low-energy spectrum of the model as the function of h^x. To this end, you will construct the Hamiltonian matrix in the σ^z basis of the 2^N–dimensional Hilbert space.

Part A: Dense matrix approach

1. Consider a reasonably large system size $N_{\text{spins}} \geqslant 10$ but that your machine can still handle. Construct Hamiltonian in the σ^z basis using the `np.kron` function. Recall that
$$\sigma_i^z \sigma_{i+1}^z \equiv \sigma_0^0 \otimes \ldots \otimes \sigma_{i-1}^0 \otimes \sigma_i^z \otimes \sigma_{i+1}^z \otimes \sigma_{i+2}^0 \otimes \ldots \otimes \sigma_{N-1}^0.$$
2. Consider 40 evenly distributed values of h^x in the interval $[-2, 2]$.
3. For every field h^x plot the lowest 10 energy levels.
4. To make the effect of phase transition at $|h^x| = 1$ more visible, plot the *difference* between 10 lowest energy levels and the ground state energy.

Hint: In the `.ipynb` exercise file you will find the template for this pipeline, which you can use. The functions `np.linalg.eigh`, `np.linspace` and `np.kron` will be necessary.

Part B: Sparse matrix
Instead of constructing a dense matrix, you can also use the bit representation to implement the action of the Hamiltonian in terms of bitwise operations. This function can be used as a scipy linear operator `scipy.sparse.linalg.LinearOperator` in the Lanczos eigenvalue solver `scipy.sparse.linalg.eigsh`. This procedure is particularly useful for larger problems.
To employ the bitstring represenation: the basis element number k should be written in the binary representation to get the individual spin orientations. For instance, the basis element $k = 415 = 256+128+16+8+4+2+1$ with $N = 10$ spins would correspond to the bitstring (0, 1, 1, 0, 0, 1, 1, 1, 1, 1), meaning, 0-th, 3-rd and 4-th spins are down, others up.

1. As a start, write the functions to convert between a spin configuration and its corresponding index
2. Implement the action of the Hamiltonian as a bitwise operation (you might find it helpful to consider the diagonal and the non-diagonal parts separately)

Part C: Symmetry operators
The transverse fiend Ising Hamiltonian has many symmetries. Finding the symmetry essentially means finding another basis where the Hamiltonian takes the block-diagonal form. Among the possible symmetries, there are:

1. Flipping all spins with the operator $\hat{A} = \otimes_i \sigma_i^x$,
2. Total magnetisation conservation with the operator $\hat{M} = \sum_i \sigma_i^z$ (only at $h^x = 0$),
3. Translation symmetry $\hat{T} : i \to i + 1 \bmod N$ with a rather non-local operator form.

If the Hamiltonian has a symmetry $[\hat{H}, \hat{A}] = 0$ (block-diagonal form in some basis), $\hat{H}$ and $\hat{A}$ can be diagonalized simultaneously. This means that the Hamiltonian does not mix the eigenstates from sectors with different eigenvalues of $\hat{A}$. This fact allows one to search for $\hat{H}$ eigenstates in the sectors of $\hat{A}$ eigenvalues one-by-one, which might significantly reduce the ED-complexity.
As the end of the first exercise, please constuct the operators for the symmetries (1) and (2) and explicitly check that $[\hat{H}, \hat{A}] = 0$ for both of them.

Exercise 2. Transverse field Heisenberg model: exact diagonalisation with symmetries
If the Hamiltonian commutes with certain unitary symmetry operator $\hat{A}$ that maps any state (bitstring) into one another state, this can be unilized in order to decrease the Hilbert space size of the problem in the following way. For every state (bitstring) $|s\rangle$, define its *orbit* as $O(|s\rangle) = \text{set}(\hat{A}^n|s\rangle)$, e.g. set of all possible different bitstrings that can be constructed out of $|s\rangle$ by applying $\hat{A}$ arbitrarily many times.
This defines an *equivalence relation*: all the bitstrings split into non-intersecting orbits (equivalence classes) $O_1, O_2, \dots O_m$. Since $\hat{H}$ and $\hat{A}$ can be diagonalized simultaneously, let us fix some specific eigenvalue λ of $\hat{A}$ and for each orbit O_α construct one $\hat{A}$ eigenstate as

$$|\xi_\alpha^\lambda\rangle = \frac{1}{\sqrt{N_\alpha}} \sum_{i=0}^{N_\alpha - 1} \lambda^{-i} \hat{A}^i |c_\alpha\rangle.$$

Here $|c_\alpha\rangle$ is the *orbit represenrative* — any priorly chosen state of the O_α orbit. It can be shown that the states $|\xi_\alpha^\lambda\rangle$ united over all eigenvalues λ and orbits α form a new basis in the Hilbert space.
Finally, in this basis it can be shown that $\langle \xi_\alpha^{\lambda_1} | \hat{H} | \xi_\beta^{\lambda_2} \rangle \propto \delta_{\lambda_1 \lambda_2}$: the Hamiltonian takes the *block-diagonal* form, different $\hat{A}$ eigenvalues do not mix that the problem can be solved for every λ independently.
In this exercise we will work with the translational symmetry given by

$$\hat{T} : i \to i + 1.$$

For any spin configuration, this operator shifts all spins one position forward (recall that we assume periodic boundary conditions). Since $T^N = 1$, the possible eigenvalues are

$$\lambda_k = \exp\left(2\pi i \frac{k}{N}\right).$$

Note that each orbit can have only specific eigenvalues, namely, we require that

$$\lambda^{|\mathcal{O}_\alpha|} = 1,$$

which is quite obvious: if one shifted the state so many times that it comes into itself (length of the orbit $|\mathcal{O}_\alpha|$), the amplitude should be the same.
Now we are ready to rewrite the Hamiltonian in the new basis of $|\xi_\alpha^\lambda\rangle$. Since the Hamiltonian has the block diagonal form in the ξ–basis, we will write a routine that will construct the block of a given eigenvalue λ. The algorithm for this might be as follows:

1. Loop over all orbits and find those that satisfy the $\lambda^{|\mathcal{O}|} = 1$ property. Write down the list of representatives. The length of this list would be the block dimension.

2. Act Hamiltonian on every $|\xi_\alpha^\lambda\rangle$. Luckily, we know the exact formula for every $|\xi_\alpha^\lambda\rangle$ as the decomposition into bitstring basis states (see above). Let us consider explicitly the action of the Hamiltonian components:

   (a) The action of $\hat{H}_J$ on the bitstring is trivial (diagonal). It leaves all bitstrings in their orbits, which means that $\hat{H}_J$ does not mix different orbits and is diagonal in the ξ–basis:

   $$\langle \xi_\alpha^\lambda | \hat{H}_J | \xi_\beta^\lambda \rangle = \delta_{\alpha\beta} \frac{1}{N_\alpha} \sum_{j=0}^{|\mathcal{O}_\alpha|-1} \langle T^j c_\alpha | \hat{H}_J | T^j c_\alpha \rangle = \delta_{\alpha\beta} \langle c_\alpha | \hat{H}_J | c_\alpha \rangle, \tag{2}$$

   where we used $|\lambda| = 1$ and the translational invariance of $\hat{H}_J$.

   (b) The $\hat{H}_h = h^x \sum_i \sigma_i^x$ field term flips spins and kicks bitstring to another orbit. To write down its matrix elements, let us fix some specific spin σ_s^x. Then spins of the orbit $\mathcal{O}_\alpha$ are mapped to some other orbits. For instance, $\sigma_x^s T^j |c_\alpha\rangle = T^{\zeta(j)} |c_{\gamma(j)}\rangle$. So, for every j one has to define the resulting orbit index $\gamma(j)$ and the distance to representative $\zeta(j)$. Then the matrix element reads

   $$\langle \xi_\beta^\lambda | \sigma_s^x | \xi_\alpha^\lambda \rangle = \frac{1}{\sqrt{N_\alpha N_\beta}} \sum_{j=0}^{|\mathcal{O}_\alpha|-1} \lambda^{j-\zeta(j)} \delta_{\beta\gamma(j)} \langle T^{\zeta(j)} c_\beta | T^j c_\alpha \rangle. \tag{3}$$

   Note that for any selected s there will be projections onto orbitals that are not compatible with λ. There is no surprise though: the Hamiltonian becomes translationally symmetric only after taking the sum over s. So, below we will neglect such contributions, but keeping in mind that they vanish only if the sum over s is performed.

1. Define orbits and eigenstates. Define a systematic way to take the representative of every orbit.

2. As a start, you need to write two functions that will map from bitstrings and back. You will also need the action of operator $\hat{T}$ on a state (given as an index). It is convenient to define $\hat{T}^n$ as an operator for an arbitrary power n.

3. Now we need to define the lookup table: it maps index to the index of its representative in the orbit. We will also need the list of unique representatives for each orbit. There is no obvious way to systematically define each orbit representative. So, a possible approach would be to iterate over all spin configurations and for each of them check if it corresponds to any already defined orbit.

4. Start with the listing of orbits that are compatible with a specific eigenvalue λ. Store them as a list of representatives for λ.

5. Check the resulting block dimensions and sum them to compare with the size of the full Hilbert space. What is the size of the largest block?

6. Write a routine to calculate the diagonal part of $\hat{H}^{\xi}$ and the non-diagonal part (set $h^x = 1$) given the above formulae. You will need to define the shift function $\zeta(j)$ and the orbit function $\gamma(j)$. To check your implementation, compare your resulting H-matrix for the case of $N = 4, k = 0, 1$ with your analytic results from Exercise 2, Question 2.

7. Plot the ground state in each λ–segment. Note that the energy levels for λ and λ^* are degenerate, so one will only see $N/2 + 1$ energy levels.

8. Measure the performance gain: how much faster it is to diagonalise two blocks independently than the initial Hamiltonian as a whole?

9. Verify that the energy spectrum agrees with the one obtained in Exercise 1.

Hint: The functions `np.linalg.eigh`, `np.linspace`, `np.kron`, `np.unique`, `np.roll`, `np.where`, `np.any` will be necessary.