Ecole Normale Supérieure de Lyon — Université de Lyon
Master: Sciences de la Matière, A. A. 2024-2025

Computational Quantum Physics (M2) — *T. Roscilde, F. Mezzacapo, F. Caleca, S. Bocini*

General rule for the TD sessions: the TD sessions are fully *hands-on* – namely, in every TD session you are supposed to *write computer codes* to learn about the phenomenology and efficiency of important algorithms for problems in quantum physics. You should choose a programming platform (Python, Matlab, Mathematica, C, Fortran, etc.), and you should be able to plot your results in the form of two-dimensional functions $y = f(x)$ (using matplotlib in Python, the plotting utilities of Matlab and Mathematica, Gnuplot, etc.), or occasionally in a more complicated form. We assume that you have some familiarity with at least one programming platform; if this is not the case, you should be able to familiarize yourself rapidly *e.g.* by attending online tutorials.

TD1: Exact diagonalization of the $S = 1/2$ XXZ spin chain
(a.k.a. one-dimensional harcore bosons or spinless fermions)

In this exercise sheet, we shall use exact diagonalization to learn about the physics of the one-dimensional XXZ model of interacting quantum spins defined on a closed ring. We will explore the use of the Lanczos algorithm for the ground-state properties and low-energy excitations; and also consider the non-equilibrium dynamics starting from a relevant initial state.

The Hamiltonian of the XXZ spin chain – describing several quasi-one-dimensional magnetic materials as well as some important synthetic quantum systems – reads

$$H = -J \sum_{i=1}^{L} \left(S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z \right) \quad (1)$$

$$= -J \sum_{i} \left[\frac{1}{2} \left(S_i^+ S_{i+1}^- + S_i^- S_{i+1}^+ \right) + \Delta \; S_i^z S_{i+1}^z \right] \quad (2)$$

Here S_i^μ ($\mu = +, -, z$) are spin-1/2 operators, such that $S^z|\uparrow\rangle = 1/2|\uparrow\rangle$, $S^z|\downarrow\rangle = -1/2|\downarrow\rangle$, $S^+|\downarrow\rangle = |\uparrow\rangle$, $S^-|\uparrow\rangle = |\downarrow\rangle$.

The XXZ spin chain is equivalent to a chain of impenetrable bosons (a.k.a. hardcore bosons), with operators $b^+ = S^+, b = S^-, b^\dagger b = S^z + 1/2$, and with the constraint $(b^+)^2 = b^2 = 0$, namely each site of the chain admits only two occupation numbers, $n = 0, 1$. The bosonic Hamiltonian takes the form

$$H = -J \sum_{i=1}^{L} \left[\frac{1}{2} \left(b_i^\dagger b_{i+1} + b_{i+1}^\dagger b_i \right) + \Delta \; (n_i - 1/2)(n_{i+1} - 1/2) \right] . \quad (3)$$

This same model can also be mapped onto a chain of spinless fermions via a nonlocal (Jordan-Wigner transformation), but this aspect goes beyond the interest of the present exercise sheet.

In the following we will consider the above models as cast on an L-site ring with periodic boundary conditions, namely $L + 1 \equiv 1$, and L an *even* number . We consider the zero magnetization sector of the Hilbert space, i.e. $\sum_i S_i^z = 0$ – or, equivalently for the case of bosons, the case of half filling, $\sum_i b_i^\dagger b_i = N = L/2$.

Throughout the rest of the exercise, you can fix $J = 1$, setting the energy unit; while Δ can vary from large and negative to large and positive, and the physics will strongly depend on its value.

1 Building the Hamiltonian matrix and fully diagonalizing it on a small ring

1.1

Start your diagonalization code by a part enumerating the $D = \binom{L}{L/2}$ basis vectors spanning the zero-magnetization (or half-filling) sector of the many-body Hilbert space. In order to store the states in memory, you can use e.g. a binary encoding of a Fock state for bosons:

$$\text{Example}: \quad 1\ 0\ 1\ 1\ 0\ 0 \rightarrow 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 + 0*2^4 + 0*2^5 = 13$$

You can store the basis using a `state(index)` vector, with `index=1,...,D` , which returns the binary encoding for the `index`-th state.
Ideally you should write your code by leaving L as a variable which can be modified in the header of your program (but remember that realistically you will be limited to sizes $L \lesssim 20$).

1.2

For a small lattice ($L \leq 10$) write the full matrix, and diagonalize it with a library built in your coding platform (with python you can use `numpy.linalg.eig` under `numpy`; with C/C++/Fortran you can use the Lapack library, etc.).

1.3

Check that the ground state satisfies translational invariance, by calculating e.g. the correlation function $\langle S^z_i S^z_{i+1} \rangle$ and show that it is independent of the site i.

1.4

Repeat the diagonalization for a grid of values of Δ, going from -4 to 4 (with special focus on the $[-1, 1]$ interval), and plot $E_1 - E_0$ (the energy gap between the ground state and the first excited state) as a function of Δ, and this for various system sizes, e.g. $L = 4, 6, 8, 10$. What do you observe?

1.5

You can also plot the nearest-neighbor correlations $\langle S^z_i S^z_{i+1} \rangle$ and $\langle S^x_i S^x_{i+1} \rangle$ as a function of Δ. What do you observe? And is there any correlation between this behavior and the one of the spectral gap?

From this point onward, you can choose to work on one of the two successive optional tasks – which both allow you to go beyond the simple full diagonalization you implemented so far.

2 Option 1: using the Lanczos method for the low-lying spectrum on bigger rings

You can now work with bigger system sizes (maybe going up to $L = 16$...) by using the Lanczos approach targeting the lowest eigenvalues.

2.1

Build the Lanczos basis from the orthonormalization of Krylov vectors: choosing an initial vector $|\phi\rangle$ from the basis (suggestion: use $|101010...\rangle$), build the sequence of M vectors ($k = 0, ..., M-1$):

$$\begin{aligned} |\psi_0\rangle &= |\phi\rangle \\ \beta_1|\psi_1\rangle &= H|\psi_0\rangle - \langle\psi_0|H|\psi_0\rangle|\psi_0\rangle \\ \beta_2|\psi_2\rangle &= H|\psi_1\rangle - \langle\psi_1|H|\psi_1\rangle|\psi_1\rangle - \langle\psi_0|H|\psi_1\rangle|\psi_0\rangle \\ &... \\ \beta_k|\psi_k\rangle &= H|\psi_{k-1}\rangle - \langle\psi_{k-1}|H|\psi_{k-1}\rangle|\psi_{k-1}\rangle - \langle\psi_{k-2}|H|\psi_{k-1}\rangle|\psi_{k-2}\rangle \\ &... \end{aligned}$$

where β_k is the normalization vector.
The application of the matrix H onto the $|\psi_k\rangle$ vectors can be calculated by a matrix-vector multiplication (if you have built the matrix in the first place), OR it can be calculated on the fly, without ever building the matrix (which may occupy too much memory).
The reduced Hamiltonian matrix to be diagonalized has the tridiagonal form

$$(H)_{\text{red}} = \begin{pmatrix} \langle\psi_0|H|\psi_0\rangle & \langle\psi_0|H|\psi_1\rangle & 0 & 0 & ... & 0 \\ \langle\psi_1|H|\psi_0\rangle & \langle\psi_1|H|\psi_1\rangle & \langle\psi_1|H|\psi_2\rangle & 0 & ... & 0 \\ 0 & \langle\psi_2|H|\psi_1\rangle & \langle\psi_2|H|\psi_2\rangle & \langle\psi_2|H|\psi_3\rangle & ... & 0 \\ 0 & 0 & ... & ... & ... & 0 \end{pmatrix} \tag{4}$$

2.2

By monitoring the value of the two lowest eigevalues (E_0 and E_1) of the reduced matrix, increase M until you have reached convergence. Check that for $L \lesssim 10$ you recover the same result as at question 1.4.

2.3

Extend the results found at question 1.4 by going to bigger system sizes ($L = 12, 14, ...$).

3 Option 2: implementing the translation symmetry

3.1

Build an array `translation(index)` that associates to each of the basis states (with index `index`)a basis state of which the `index`-th state is a translation (e.g. 101100 is a translation of 001011). In practice you can proceed as follows:

1. Set `translation(index)=-1` for all the states;
2. Scan the states with label `index` one by one; if `translation(index)=-1` , then `translation(index)=index` .
3. Scan the states with index `jndex>index` ; if `state(jndex)` is a translation of `state(index)` , then `translation(jndex)=index` .

With this procedure you will be able to identify a subset of $O(D/L)$ basis states $|\psi_l\rangle$ from which all other states can be produced by translation, namely the states $|\psi_l\rangle$ such that `translation(index)=index` .

3.2

From the representative states $|\psi_l\rangle$, build a reduced basis of zero-momentum states

$$|\psi_l^{(0)}\rangle = \frac{1}{\sqrt{L}} \sum_{d=0}^{L-1} T_d |\psi_l\rangle \tag{5}$$

where T_d is the operator translating a state by a distance d on the ring.

3.3

Build the Hamiltonian matrix using these states only, in order to diagonalize it in the zero-momentum sector. Check that the ground state energy you find for small system sizes is the same as what found from the full diagonalization at question 1.4.

(If you wonder why the ground state should have zero momentum, think about what bosons typically do in their ground state...).

---

4 Bonus: studying the non-equilibrium dynamics

In this bonus section, we would like to consider the non-equilibrium evolution starting from a state which is *not* an Hamiltonian eigenstate – namely, we would like to simulate a so-called *quantum quench*.
In the following, we shall choose the initial state $|\psi(0)\rangle = |101010...\rangle$, and we would like to monitor how its density/spin pattern evolves for different values of Δ.

4.1

To calculate the time evolution under Hamiltonian H, use the full diagonalization at question 1.4, namely the knowledge of the spectrum $H|\phi_m\rangle = E_m|\phi_m\rangle$, to evolve $|\psi(0)\rangle$ as

$$|\psi(t)\rangle = e^{-iHt}\ |\psi(0)\rangle = \sum_m\ e^{-iE_m t}\ \langle\psi_m|\psi(0)\rangle\ |\psi_m\rangle\ . \tag{6}$$

4.2

Calculate the evolution of the *imbalance* $I = \sum_i (-1)^{i+1} n_i$ (which starts at its maximum value of $L/2$ for $t = 0$) for $\Delta = 0$ and $\Delta = \pm 2$. What do you observe? Do you have any explanation?