Numerical simulation of the Quantum Ising Model in transverse-field

Salvador Alberto
Università degli Studi di Padova
Course: Quantum Information and Computing
Assignment 7

January 7, 2025

Contents

1 Introduction

2 Theoretical framework

2.1 Quantum Ising Model in transverse- field

The quantum Ising model in transverse-field describes a many-body system made by N interacting particles on a 1D lattice, subjected to an external magnetic field. The particles are assumed to be spin 1/2. Each particle interacts only with its nearest neighbors. This interaction results in the alignment or anti-alignment along a specific axis. In this study, I consider this to be the x-axis. The magnetic field is *transverse* in the sense that it acts on a perpendicular direction to x. I consider it to be parallel to z. I consider *open* border conditions, meaning that the first and last particles of the lattice do not interact with each other. The Hamiltonian describing such system is:

$$\hat{H} = -J \sum_{i=1}^{N-1} \hat{\sigma}_i^x \hat{\sigma}_{i+1}^x - \lambda \sum_{i=1}^{N} \hat{\sigma}_z. \tag{1}$$

It is an Hermitian observable, as it is a linear combination of Hermitian operators, namely the Pauli matrices. The first term is called *interaction term* and characterizes the particle-particle interaction. The second term defines the interaction between particles and the magnetic field. Both terms are weighted by a parameter of the model. Depending on their values, we can have different types of interactions. λ determines the strength of the magnetic field. We assume it to be negative, in the range $[-3, 0]$: the particles tend to be aligned in the *same* direction of the field. J can be either positive or negative: $J > 0$ characterizes *ferromagnetic* interactions, with each pair of spins that tends to be aligned in the same direction to minimize the energy; $J < 0$ defines *anti-ferromagnetic* interactions between spin pairs, with anti-alignment favoured. In the following, I consider $|J| = 1$ and, unless stated otherwise, $J = -1$.

2.2 Quantum phase transition in the Ising model

The behavior induced by (1) strongly depends on the value of λ. Consider the $\lambda = 0$ case. The Hamiltonian is then composed only by the interaction term. In the $J = -1$ anti-ferromagnetic case, the particles tend to be anti-aligned along the x-axis. The ground state of this system is the one with maximal anti-alignment, with every spin aligned oppositely to the following one, as in $|{\rightarrow}{\leftarrow}{\rightarrow}\cdots{\leftarrow}{\rightarrow}\rangle$. We say the system is in an *ordered phase*. As λ increases, the external magnetic field starts characterizing the dynamics. In the $\lambda >> J$ case, the prevalent interaction is the one between the particles and the field. This leads the particles to be aligned the direction of the field, as in $|{\uparrow}{\uparrow}\cdots{\uparrow}\rangle$. We say the system is in a *disordered phase*. The reason for the name is that each particle acts independently of all the others and all the quantum correlations between them are lost. In between these two phases, there is an intermediate regime that we refer to as *phase transition*. This is characterized by values of λ comparable to J. Both terms in the Hamiltonian (1) contribute to the dynamics.

Since $\hat{\sigma}_z$ and $\hat{\sigma}_x$ do not commute, this ultimately results in *quantum fluctuations* that lead the spins the spins to be in a state with a mix of alignment between z and x. The stronger the field, the more powerful these fluctuations are. There exists a critical value of the field strength at which the fluctuations are such to determine an abrupt change in the system's property, the so-called phase transition.

2.2.1 Phase transition and magnetization

One way to observe the quantum phase transition is to look at the system's *magnetization*. Consider the magnetization along the z-axis. We compute its average value over all spins, defined as:

$$\hat{M}_z = \sum_{i=1}^{N} \frac{\hat{\sigma}_z}{N}. \tag{2}$$

This measures the degree of z-alignment of the spins. It takes values between -1, when none of the spins are aligned with the magnetic field, and 1 in the opposite case, when all the spins are aligned to the field. In the $\lambda << J$ "ordered" phase we expect a zero-valued $\hat{M}_z$: all spins are aligned along the x-axis. In contrast, in the $\lambda >> J$ "disordered" phase, it should be equal to $+1$, since all spins tend to align with the field to minimize the energy of the system.

3 Numerical Methods

3.1 Building the Hamiltonian matrix

The first necessary thing to simulate the quantum Ising model is to obtain the matrix representation of the Hamiltonian (1). To carry it out, firstly I define a `pauli_matrix` function which allows me to easily retrieve one among the Pauli matrices as numpy arrays. Then, I need another function, named `hamiltonian_ising`. This returns the complete Hamiltonian given as input the system size N and field strength λ. Inside of it, I call two sub-routines, one for building each of the two terms composing the Hamiltonian. For the field term, I have the `gen_hamiltonian_field` function. For each of the N particles, it computes energy of interaction with the field:

$$H_{field,i} = \mathbb{1}_1 \otimes \cdots \otimes \mathbb{1}_{i-1} \otimes \hat{\sigma}_i^z \otimes \mathbb{1}_{i+1} \otimes \cdots \otimes \mathbb{1}_N. \tag{3}$$

Exploiting the associativity property of the tensor product, I separate this product into three sub-terms (two in the case the i-th particle is in one of the extremes). I compute each subterm separately, put the results in a list, and finally take the product between the elements of the list. To compute the tensor product I use the `numpy.kron` function. I do this for each particle and store their single contributions in another list. I finally sum all these terms to obtain the field term of the Hamiltonian. To compute the interaction term I follow an analogous approach, this time computing:

$$H_{field,i} = \mathbb{1}_1 \otimes \cdots \otimes \mathbb{1}_{i-1} \otimes \hat{\sigma}_i^x \otimes \hat{\sigma}_{i+1}^x \otimes \mathbb{1}_{i+2} \otimes \cdots \otimes \mathbb{1}_N. \tag{4}$$

In the first version of my implementation, I used standard numpy arrays for storing the matrices discussed above. To gain better performance, I consequently modified the code to use sparse matrices from the `scipy` library. The first version is contained in the `ising_module.py` file, while the updated version is in `ising_module_sparse.py`. In Sec.4 I discuss the difference in performance for the diagonalization of the Hamiltonian between these two versions. I use the non-sparse module only for this comparison, considering as main code the one using sparse matrices.

3.2 Building the magnetization operator

Inside the `ising_module_sparse.py`, I define a `magnetization_z` function, able to compute the expectation value of the z-component of the magnetization operator. Inside it, I construct the matrix version of the magnetization operator (2). This is easily done by exploiting its similarity with the field term of the Hamiltonian. I simply call `gen_hamiltonian_field` dividing by the number of particles. The same function computes the expectation value of $\hat{M}_z$ for the system in a state $|\psi\rangle$, represented in the wave-function formalism using the computational basis, passed as an input argument. By doing so, I obtain:

$$m_z = \langle\psi|\hat{M}_z|\psi\rangle = \sum_{i,j} \psi_j^* M_{i,j} \psi_i. \tag{5}$$

To achieve this I use the `numpy.dot` and the `numpy.conj` functions. I choose not to use the analogous sparse methods here because the computational cost of a matrix-vector product scales as $\mathcal{O}(N^2)$, which is significantly lower than the $\approx \mathcal{O}(N^3)$ scaling of matrix diagonalization. For the system sizes accessible using sparse diagonalization methods, products between dense matrix and vectors remain computationally feasible.

4 Results

In the following, I present the results I obtain from my simulation of the quantum Ising model in transverse field discussed above. All results are computed using sparse matrices methods, implemented inside the `ising_module_sparse.py` script. The decision to use sparse matrices was motivated by their ability to handle larger systems: while using the diagonalization method from NumPy for dense matrices the maximum system size N my laptop could simulate is 12, using the sparse methods instead I am able to reach $N = 20$.

4.1 Diagonalization of different Ising systems

Exploiting the functions described in Sec.3.1, I obtain the Hamiltonian for a system with a given size N and field strength λ. One useful thing to do dealing with quantum systems is to diagonalize the Hamiltonian, to obtain the energy spectrum. In the following, I want to study how the choice of the parameters N and λ affects the diagonalization operation. In addition, it is interesting to look at how the eigenvalues change varying these parameters. Inside of the module `diag_ising_benchmark.py` I write a code that builds the Hamiltonian

through the `hamiltonian_ising` function discussed in Sec.3.1. Then, I proceed to diagonalize it. Since it is a Hermitian matrix, I use the related diagonalization methods from the SciPy library, namely the function `sparse.linalg.eigsh`. This allows me to retrieve the first k eigenvalues, from the smallest algebraic one. I set $k = 11$ as the very first energy levels are enough to understand the general behavior. I normalize all the eigenvalues dividing them by the number of sites of the system. This allows for an easy comparison between the results obtained for different systems. I store the results in a pandas DataFrame that I consequently save in a CSV file. The code is structured in a way to test multiple paired values of N and λ iteratively. For N I choose it to be in the range from 6 to 20, with steps of 1. For λ I set the following values $\{-3, -2, -1, 0\}$.
In Fig.1 I display some of the results obtained, in particular showing how the energy of the ground and first excited states vary with the field strength, for different system sizes. The plot reveals that as the strength increases, the system size has a greater impact on the energy of the first excited state, showing increased dispersion. In contrast, the ground state's energy remains almost the same for every N. The stability of the ground state energy could be explained considering that it represents the most stable configuration, and this is not altered by the addition of more spins to the system. Instead, with increasing system sizes, the number of different configurations leading to the first excited states becomes grows significantly. This ultimately leads to noticeable differences in the energy of the first excited state across system sizes.

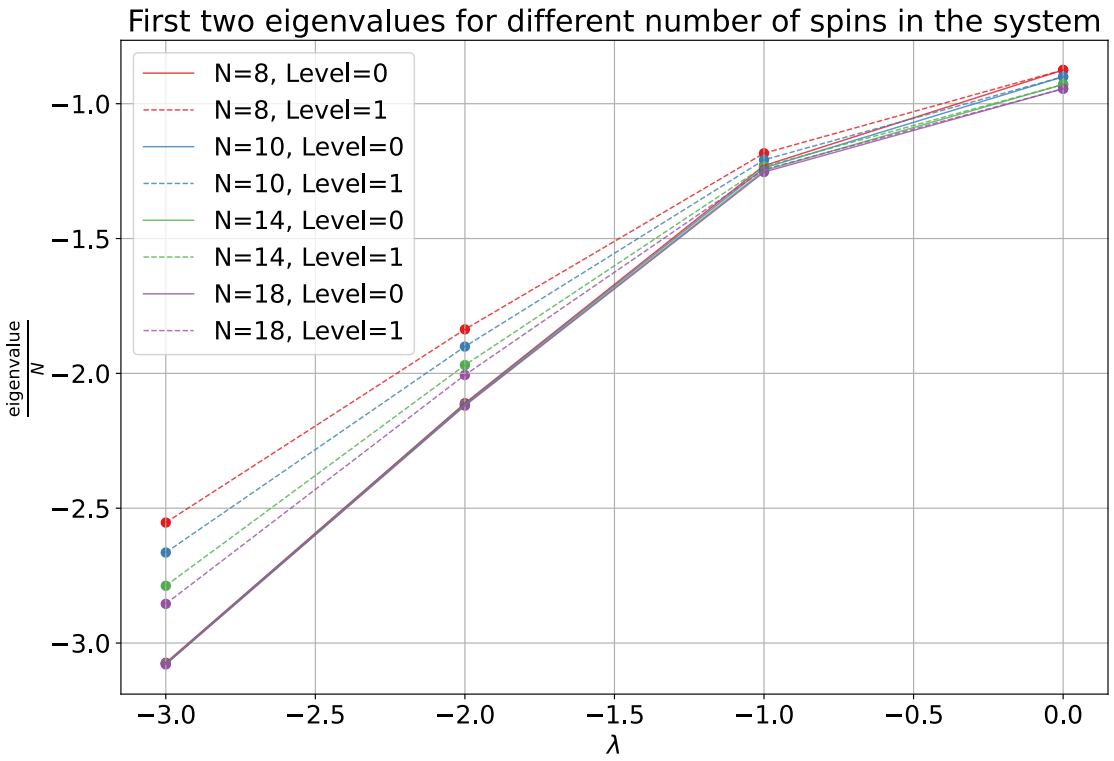

Figure 1: First two eigenvalues (ground state and first excited state) of the transverse-field Ising model as a function of the transverse field parameter λ, for different system sizes ($N = 8, 10, 14, 18$). The ground state energies are linked by a continuous line while the first excited one by a dashed line

4.2 Observing the phase transition through $\hat{M}_z$

One of the most interesting features of the quantum Ising model is the phase transition that happens by varying the field strength λ, as discussed in Sec.2.2.1. This transition can be observed by looking at the z-component of the magnetization operator, i.e., the one parallel with the magnetic field. I investigate this phenomenon by computing the expectation value of $\hat{M}_z$ with the system in the ground state. I do so for different values of λ and different system sizes N, to check whether the phase transition occurs every time and with the same characteristics. The code I use is the one in the `magnetization_ising.py` script. Inside of it, I set a specific set of values for N and λ. For each value of N, I build a set of systems with different values of λ. In particular, I choose to use the *numpy.logspace* function to obtain 30 points between 10^-3 and 10^2, evenly spaced in a logarithmic scale. For each system, I build the Hamiltonian matrix and diagonalize it following the strategy discussed in Sec.4.1. I set the number of eigenvalues and eigenvectors to be computed to one to retrieve only the ground state. I compute the expectation value of $\hat{M}_z$ for the system in the ground state calling the `magnetization_z` function presented in Sec.3.2. I iterate this operation for every value of λ and for every N. I store all the results in a pandas DataFrame that I consequently save in a CSV file. Some of the results I obtain are shown in Fig.2.
From Fig.2 we see how the system for weak field values has almost zero transverse magneti-

zation. This is in accordance with the theoretical framework discussed in Sec.2.2.1: all the spins are aligned along the x-axis, and so the expectation value of $\hat{M}_z$ is zero. In contrast, when the field becomes more powerful, the interaction that mostly affects the system behavior is the one with the field. The spins tend to be aligned with it and the magnetization goes to one, as we observe from the plot. The phase transition occurs at $\lambda_C \approx -1$ for every system considered, which is exactly the value of the coupling constant J I have set for all the systems. This confirms our intuition: the transition occurs when the two contributions to the Hamiltonian (1) are of the same order of magnitude. The system size does not affect the occurrence of the transition.

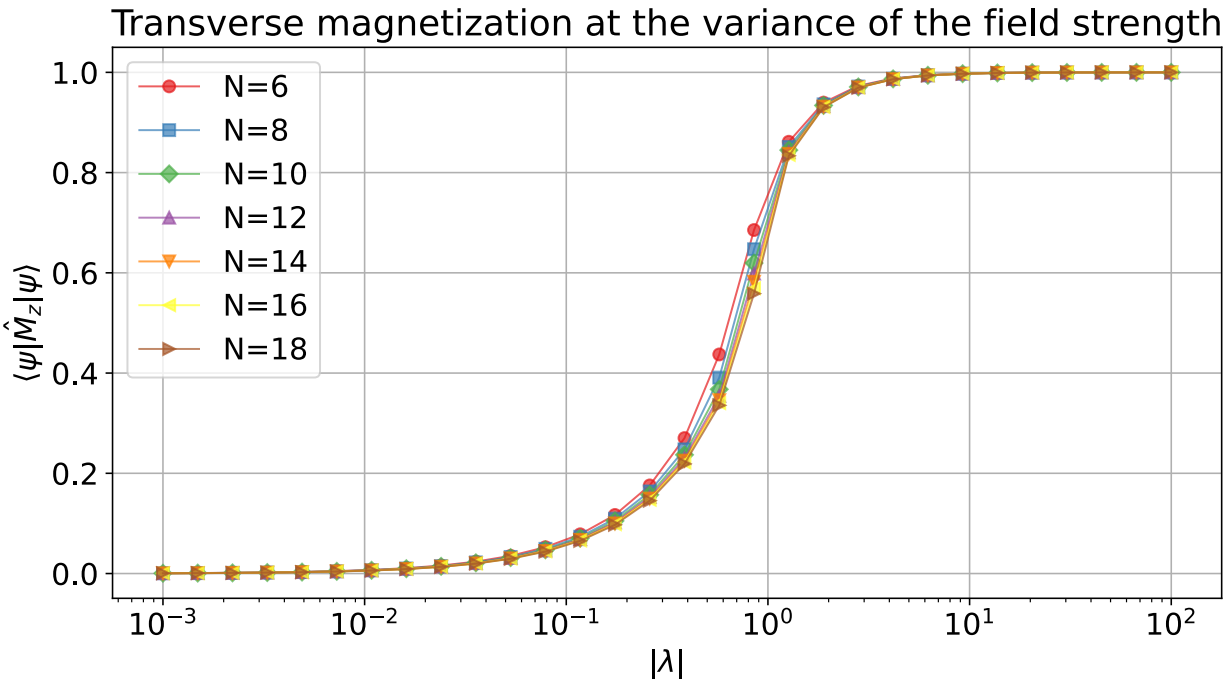

Figure 2: Transverse magnetization expectation value as a function of the field strength λ for various system sizes ($N = 6, 8, 10, 12, 14, 16, 18$). The field strength is reported referring to its absolute value. The magnetization transitions from 0, in the weak-field regime, to 1, in the in the strong-field regime.

5 Conclusions

In this study I have correctly implemented a numerical simulation of the quantum Ising model in transverse field.
The use of sparse matrices for the description of its Hamiltonian resulted to be very beneficial. The maximum number of spins composing the system that a generic laptop could effectively simulate is about 20. Instead, with the use of dense matrices this is limited to ≈ 12. With both methods the diagonalization takes at most one minute, and the computation fails when the maximum system size is reached.
The impact of the field on the eigenvalues can be summarized in the two following facts:

- The energy of the system decreases with increased strength of the field.
- Depending on the level considered, the eigenvalues for different system sizes become more spread around an average value for increased values of the field strength. This can be explained considering the number of the spin configurations that describe the specific level.

Finally, the phase transition is observed for every system considered. It occurs around $\lambda_c \approx -J = -1$ as expected from the theory. The system size does not affect the critical value λ_c.

References

[1] D. J. Griffiths, *Introduction to Quantum Mechanics*, 3rd Edition, Pearson, 2018.

[2] S. Montangero, *Introduction to Tensor Network Methods*, Springer, 2018.