Week 9: Quantum Ising Model

Alice Pagano

(Dated: December 14, 2020)

In this Report, we initilize and diagonalize the Hamiltonian of an Ising model with N spin-1/2 particles in a one-dimensional lattice. In particular, we study the energy spectrum as a function of the interaction strength λ.

I. THEORY

The **quantum Ising model** represents one of the simplest nontrivial many-body quantum system. Let us consider a linear chain of N interacting spins 1/2 in presence of an external field of *intensity* λ. The Hamiltonian of the model reads:

$$H = \sum_{i=1}^{N-1} H_{i,i+1} = \lambda \sum_{i=1}^{N} \sigma_i^z + \sum_{i=1}^{N-1} \sigma_i^x \sigma_{i+1}^x \tag{1}$$

where σs are the Pauli matrices and the coefficient λ determines the relative strength of the external field compared to the nearest neighbour interaction.

More particularly, this system features a lattice with nearest neighbour interactions determined by the alignment or anti-alignment of spin projections along the x-axis, as well as an external magnetic field perpendicular to the x-axis (without loss of generality, along the z-axis) which creates an energetic bias for one z-axis spin direction over the other. An important feature of this setup is that, in a quantum sense, the spin projection along the z-axis and the spin projection along the x-axis are not commuting observable quantities.

To solve with a numerical simulation the Ising model for N particles, let us remind that the Pauli matrices can be rewritten in an explicit form as:

$$\sigma_i^z = \mathbb{1}_1 \otimes \mathbb{1}_2 \otimes \cdots \otimes \sigma_i^z \otimes \cdots \otimes \mathbb{1}_N \tag{2a}$$

$$\sigma_i^x \sigma_{i+1}^x = \mathbb{1}_1 \otimes \mathbb{1}_2 \otimes \cdots \otimes \sigma_i^x \otimes \sigma_{i+1}^x \otimes \cdots \otimes \mathbb{1}_N \tag{2b}$$

II. CODE DEVELOPMENT

In order to write the hamiltonian of the Ising model for a system with N particles, we develop a program inside the file "ising_hamilt.f90".

First of all, a user-defined `FUNCTION` **`tensor_product`**`(Mat1,Mat2)` is coded for performing the tensor product between two matrices (i.e. between operators) `Mat1` and `Mat2`.

```
! Compute tensor product between two generic matrices
function tensor_product(Mat1,Mat2) result(Tens)
    ...
    N1 = shape(Mat1)
    N2 = shape(Mat2)
    N(1) = N1(1)*N2(1)
    N(2) = N1(2)*N2(2)

    allocate(Tens(N(1),N(2)))

    do ii=1,N1(1),1
        do jj=1,N1(2),1
            do kk=1,N2(1),1
                do mm=1,N2(2),1
                    Tens( (ii-1)*N2(1)+kk, (jj-1)*N2(2)+mm ) = Mat1(ii,jj)*Mat2(kk,mm)
                end do
            end do
        end do
    end do

end function tensor_product
```

Then, the main steps of the program are in order:

1. the total number of subsystems `N` and the strength λ are given as input;
2. the matrix σ_x and σ_z are defined;
3. the total hamiltonian `H` is computed as the sum between a non-interacting term `H_non_int` and an interacting one `H_int` as:

```
H = lambda*H_non_int(N_part,sigmaz) + H_int(N_part,sigmax)
```

In particular, the two terms are computed by calling:

- the `FUNCTION` **`H_non_int(N,sigmaz)`**, which takes in input the number of particles and σ_z. It performs the tensor product between σ_z and the identity matrix by exploiting the relationship given by Eq. (2a).

```
! Compute non-interacting term of Ising Hamiltonian
function H_non_int(N, sigmaz) result(H0)

    integer :: N, dim
    integer :: ii, jj, kk
    complex(8), dimension(:,:) :: sigmaz
    complex(8), dimension(:,:), allocatable :: H0
    complex(8), dimension(:,:), allocatable :: tmpMat

    dim = size(sigmaz,1)**N
    allocate(H0(dim,dim))
    allocate(tmpMat(dim,dim))

    H0 = cmplx(0.0,0.0)

    do ii=1,N,1
        tmpMat = tensor_product ( tensor_product( identity(ii-1), sigmaz) , identity(N-ii) )
        H0 = H0 + tmpMat
    end do

end function H_non_int
```

- the `FUNCTION` **`H_int(N,sigmax)`**, which takes in input the number of particles and σ_x. It performs the tensor product between σ_x and the identity matrix by exploiting the relationship given by Eq. (2b).

```
! Compute interacting term of Ising Hamiltonian
function H_int(N,sigmax) result(H1)

    integer :: N, dim
    integer :: ii, jj, kk
    complex(8), dimension(:,:) :: sigmax
    complex(8), dimension(:,:), allocatable :: H1
    complex(8), dimension(:,:), allocatable :: tmpMat

    dim = size(sigmax,1)**N
    allocate(H1(dim,dim))
    allocate(tmpMat(dim,dim))

    H1 = cmplx(0.0,0.0)

    do ii=1,N-1,1
        tmpMat = tensor_product( tensor_product ( tensor_product( identity(ii-1), sigmax),
                                 sigmax ), identity(N-ii-1) )
        H1 = H1 + tmpMat
    end do

end function H_int
```

4. then, the Ising hamiltonian is diagonalized by calling the Lapack's routine **`zheev`**;

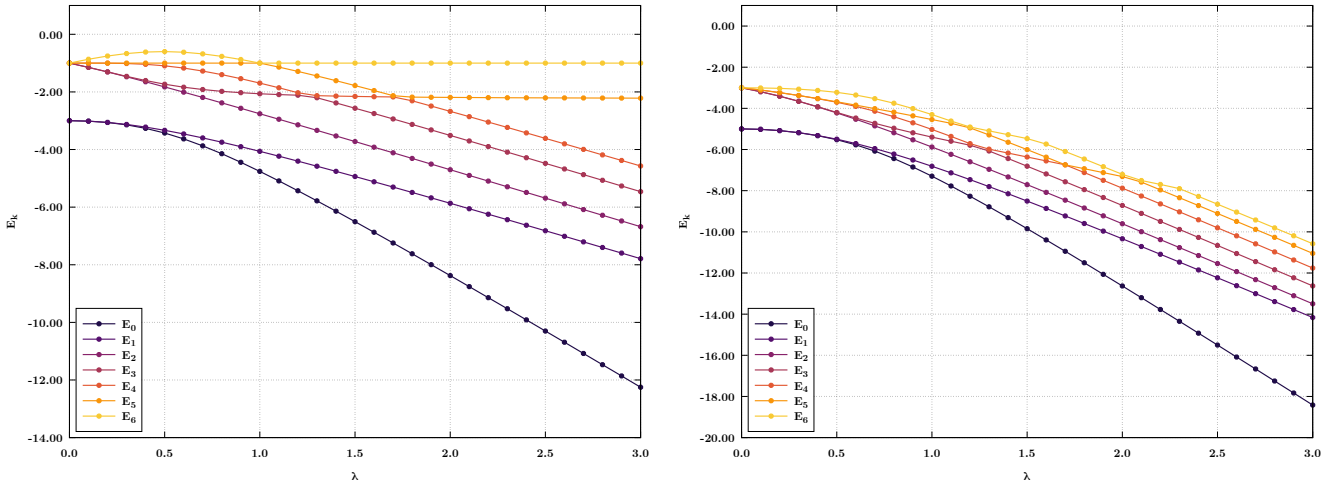
5. the first `klevels` eigenvalues are stored in order to study the energy spectrum for a fixed number of particles `N` and strength `lambda`.

Finally, with a python script "script.py", we execute the program for different `N` and different `lambda`. At the end, the first `klevels` of the energy spectrum for different particles are plotted as a function of `lambda`.

III. RESULTS

We execute the program for $N \in [3:10]$ and $\lambda \in [0:3]$. In Fig. 1 the first seven eigenstates as a function of λ for different N are shown.

Let us focus on the ground state E_0 and on the first excited state of the hamiltonian E_1. We can note that for a nill interaction strength they are degenerate, independently on the number of particles used in the simulation. However, for $\lambda \neq 0$, the ground state and the first excited state start to differ. It is interesting to notice that the difference $E_1 - E_0$ decrease with higher N. This trend is more visible in Fig. 2, where the diffeence between the two eigenstates is plotted as a function of λ. It seems that the difference for $N \to \infty$ approaches a zero value for $\lambda < 1$, denoting the presence of a quantum phase transition at $\lambda = 1$.



(a) 4 particles

(b) 6 particles

(c) 8 particles

(d) 10 particles

FIG. 1 Plots of the energy spectrum for the first 7 levels as a function of strength λ for a different number of particles N.

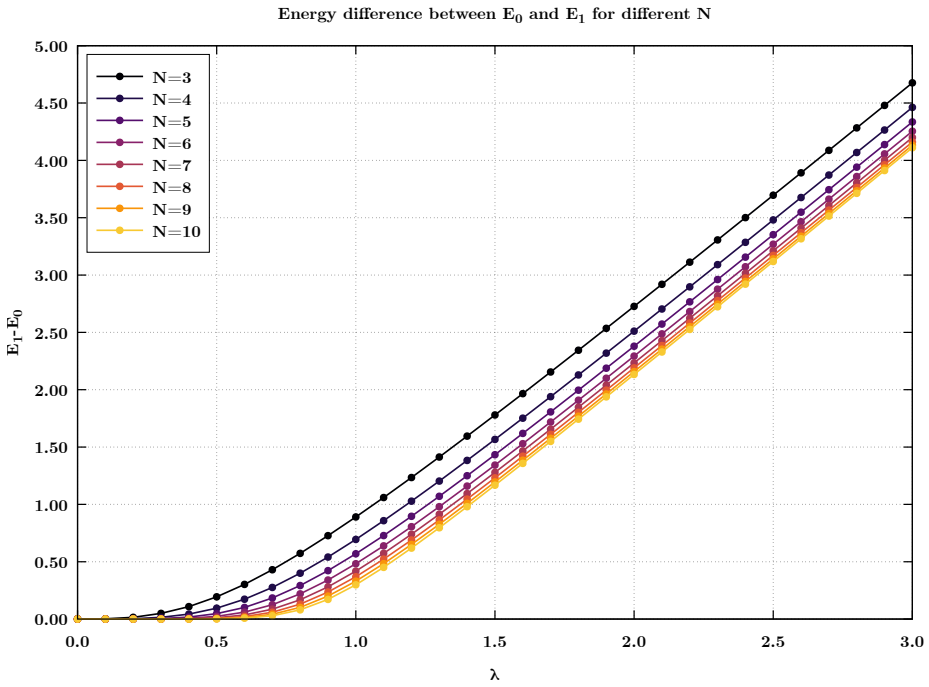


FIG. 2 Difference between first excited state E_1 and ground state E_0 as a function of the interaction length λ for different number of particles N.

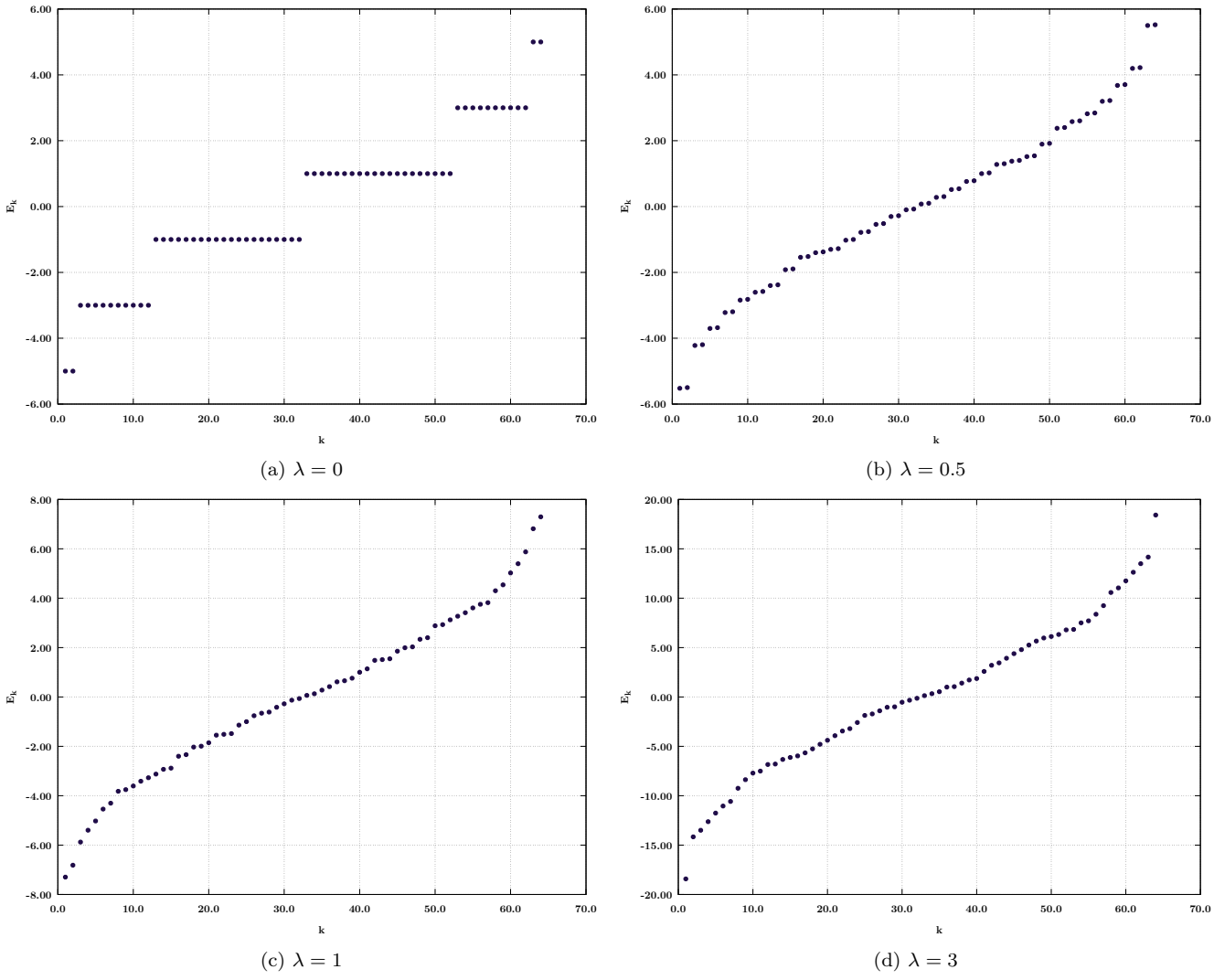
Now, let us focus on the case of $N = 6$ particles. In Fig. 3, we can see the energy spectrum for fixed values of the interaction strength.

As already mentioned, in Fig. 3a, we notice that for $\lambda = 0$ the ground state is two-fold degenerate. This corresponds to the case when all the spin are aligned down. In particular, the ground state energy is equal to $N - 1 = 5$. Furthermore, the first excited state turns out to be $2(N - 1) = 10$ degenerate.

Then, for $\lambda = 0.5$ the degeneracy disappear as we can see in Fig. 3b. However, in this case we note a trend of the energy states to be near in pairs. This trend completely fate away when $\lambda = 1$ and expecially for $\lambda = 3$. Moreover, we note that by increasing the strength coefficient, the ground state level become more distant from all the other states.

IV. SELF-EVALUATION

The code implemented works well and returns consistent results. In particular, we learn how to simulate a quantum Ising model and we study its spectrum for different values of the strength parameters λ. In a further implementation of the code, it could be useful to optimize the function for the tensor product in order to be able to simulate systems with higher values of N.



(a) $\lambda = 0$

(b) $\lambda = 0.5$

(c) $\lambda = 1$

(d) $\lambda = 3$

FIG. 3 Plot of energy levels for an Ising model with 6 particles.