Quantum Ising Model

Abstract

In this assignment a quantum system composed by N particles of spin 1/2 in a one-dimensional lattice is considered. This system is described by an Ising Hamiltonian. We are asked to write the matrix representation of the given Hamiltonian and then to diagonalize it, varying N. At the end the spectrum including the first 5 energy levels (k=5) is analyzed.

Theory

The system composed by N spin-1/2 particles is described by the following Hamiltonian:

$$\hat{H} = \lambda \sum_{i=1}^{N} \sigma_z^i + \sum_{i=1}^{N-1} \sigma_x^i \sigma_x^{i+1} \tag{1}$$

where the σs are the Pauli matrices and λ is the interaction strength, considered $\lambda \in [0,3]$.
The Pauli matrices (in the z-basis) are:

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2}$$

To compute the matrix representation of this Hamiltonian, we have to define the Hilbert space $\mathcal{H} = \bigotimes_{i=1}^{N} \mathcal{H}_i$ where $\mathcal{H}_i$ is the Hilbert space of the i-th site. The dimension of this $\mathcal{H}$ is 2^N. On the contrary the terms that composed our Hamiltonian are not defined on a space of dimension 2^N, since each Pauli matrix acts on a Hilbert space of dimension 2. Moreover each term acts on its respective subsystem.
The given Hamiltonian can be splitted in

- Diagonal part $\lambda \sum_{i=1}^{N} \sigma_z^i$
  This term is diagonal in the z-basis and the i-th term in the sum can be written as

$$\bigotimes_{j=1}^{i} \mathbb{1}_2 \otimes \sigma_z^i \otimes \bigotimes_{j=i+1}^{N} \mathbb{1}_2 \tag{3}$$

- Interaction term $\sum_{i=1}^{N-1} \sigma_x^i \sigma_x^{i+1}$
  Each term of the summation is a tensor product of two matrices

$$\sigma_x \otimes \sigma_x = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

The system is not separable. One can use *Mean-Field* approximation to estimate the energies.
It holds:

$$E[\psi_{MF}] = \langle \psi_{MF} | \hat{H} | \psi_{MF} \rangle = \lambda \sum_{j=1}^{N} \langle \psi_{MF} | \sigma_z^j | \psi_{MF} \rangle + \sum_{j=1}^{N-1} \langle \psi_{MF} | \sigma_x^j | \psi_{MF} \rangle^2 \tag{5}$$

When $N \to \infty$

$$e[\psi] = \frac{E[\psi]}{N} \xrightarrow{N \to \infty} \lambda \langle \psi_{MF} | \sigma_z | \psi_{MF} \rangle + \langle \psi_{MF} | \sigma_x | \psi_{MF} \rangle^2 \tag{6}$$

This quantity has to be minimized to find the ground state of the system. The result is the following:

$$e = \begin{cases} -1 - \frac{\lambda^2}{4} & \lambda \in [-2, 2] \\ -|\lambda| & \lambda \notin [-2, 2] \end{cases} \tag{7}$$

Code development

Firstly two MODULEs are included

- The usual MODULE 'DEBUGMOD', used also in previous exercises, to debug a generic code
- The 'OPER' MODULE, that contains a function to do the tensor product of the two matrices given as input.

```
FUNCTION TENSOR_PROD(m_1, m_2)
  ! This function takes as input two matrices
  ! and does the tensor product
  IMPLICIT NONE

  COMPLEX*16, DIMENSION(:,:) :: m_1, m_2
  COMPLEX*16, DIMENSION(SIZE(m_1,1)*SIZE(m_2,1),SIZE(m_1,2)*SIZE(m_2,2)) ::
  TENSOR_PROD
  INTEGER*8 :: ii, jj, left_r, left_c, right_r, right_c

  left_r = SIZE(m_1,1)
  left_c = SIZE(m_1,2)

  right_r = SIZE(m_2,1)
  right_c= SIZE(m_2,2)

  DO ii=0, left_r-1
     DO jj=0, left_c-1
        TENSOR_PROD(ii*right_r+1:(ii+1)*right_r,jj*right_c+1:(jj+1)*right_c) =
  m_1(ii+1,jj+1)*m_2(:,:)
     END DO
  END DO

  RETURN
END FUNCTION TENSOR_PROD
```

Listing 1: Function to perform the tensor product

Moreover, at each run the program asks the user the number of particles N and the dimension d of the subsystems. The aforementioned MODULE 'DEBUGMOD', activated if the logical flag 'debug_flag' is set to TRUE, is used to check if the inserted quantities are positive, otherwise a warning message is printed.

```
CALL DEBUG(debug_flag,d,d < 0 ,'Insert a positive dimension')
CALL DEBUG(debug_flag,N,N < 0 ,'Insert a positive number of particle')
```

Listing 2: DEBUG check

In the main PROGRAM, called Ex9, the Hamiltonian is evaluated considering separately the two parts, the diagonal part, called H_nonint in the code, and the interaction term , called H_int. Both are initialized as matrices of dimension $d^N x d^N$. The first part, since it includes a diagonal matrix, is simply evaluated as a vector, then shaped in matrix form. The interesting part concerns the computation of the interaction term, in which we have to evaluate N-1 tensor products. Recall that the first one is easy to evaluate, it is the one in the **Theory** section. As the listing 3 shows, a 'container matrix' , allocated with the same dimension of H_int is used to perform the computation, that includes the previous defined function TENSOR_PROD.

```
! interaction terms (that are N-1)
DO ii=1,N-1

   container = 0

   DO jj=1,N

      IF (jj.EQ.1) THEN

         IF (((N+1-jj).EQ.ii).OR.((N+1-jj).EQ.(ii+1))) THEN
            container(1:d**(jj),1:d**(jj)) = sigma_x(:,:)

         ELSE
            container(1:d**(jj),1:d**(jj)) = id_2(:,:)

         END IF
      ELSE

```

```
19          IF (((N+1-jj).EQ.ii).OR.((N+1-jj).EQ.(ii+1))) THEN
20              container(1:d**(jj),1:d**(jj)) = TENSOR_PROD(sigma_x, container(1:d**(
    jj-1),1:d**(jj-1) ))
21
22          ELSE
23              container(1:d**(jj),1:d**(jj)) = TENSOR_PROD(id_2, container(1:d**(jj
    -1),1:d**(jj-1) ))
24
25          END IF
26       END IF
27    END DO
28
29    H_int(:,:) = H_int(:,:) + container(:,:) ! Update the interaction term
```

Listing 3: Evaluation of the interaction term

I have chosen to perform 15 iterations (15 values of λ), with λ calculated according to the following rule

$$\lambda = \frac{3}{14} * i \tag{8}$$

in a loop with $i = 0, 14$. Recall that $\lambda \in [0,3]$.
The whole Hamiltonian is then diagonalized, using `ZHEEV` subroutine provided by `LAPACK`. The documentation can be found at the following link [1]. At the end a matrix of dimension 15x6 is printed on a file, called `lambda_eigenvalues.txt`. In the first column it contains the values of λ and in the other 5 columns the corresponding eigenvalues, so we have the first 5 energy levels. The results are plotted through the `GNUPLOT` script `'plot.gnu'`, for different values of N. The plots are shown in the following section.

Results

The maximum value for the number of particles N_{max} that can be reached is N=12. For higher numbers the terminal gives a memory error, the system is not able to allocate the objects. This is linked to the RAM available in my laptop.
In the following, the graphs of the first 5 energy levels (k=5) vs λ are shown. The dimension of the subsystems, d, is fixed to 2 while N varies. In the plots also the theoretical values, obtained from equation (7) are shown. To compare the solutions given by the program with the theoretical ones, the eigenvalues are divided by N, so in the y axis there is E/N. However for $\lambda = 0$ would be more correct normalize with respect to $N - 1$, since there are only $N - 1$ tensor products. This detail can be neglected if we consider that the theoretical result is obtained in the thermodynamic limit, $N \to \infty$. For $N = 2$ the 5th energy level is zero as expected, since we have only two particles.

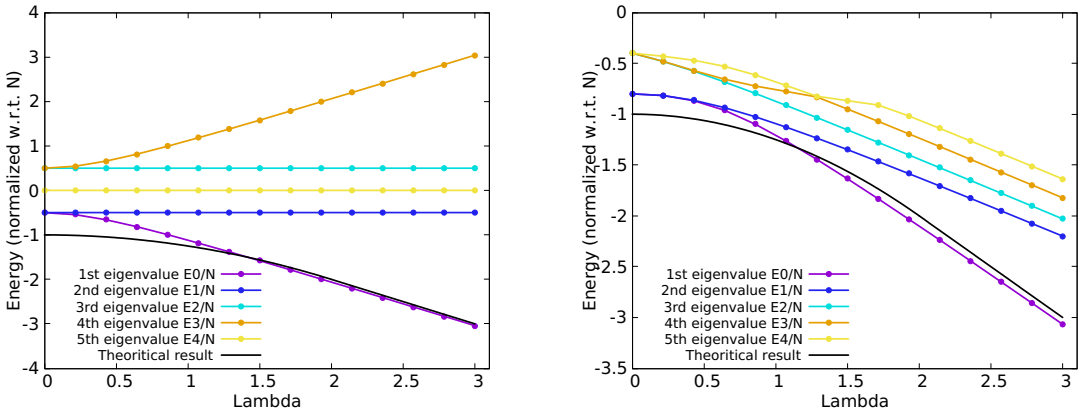

Figure 1: Plot of the first 5 energy levels vs λ with the theoretical result, for N=2 (left) and N=5 (right).

[1] http://www.netlib.org/lapack/explore-html/df/d9a/group__complex16_h_eigen_gaf23fb5b3ae38072ef4890ba43d5cfea2.html#gaf23fb5b3ae38072ef4890ba43d5cfea2

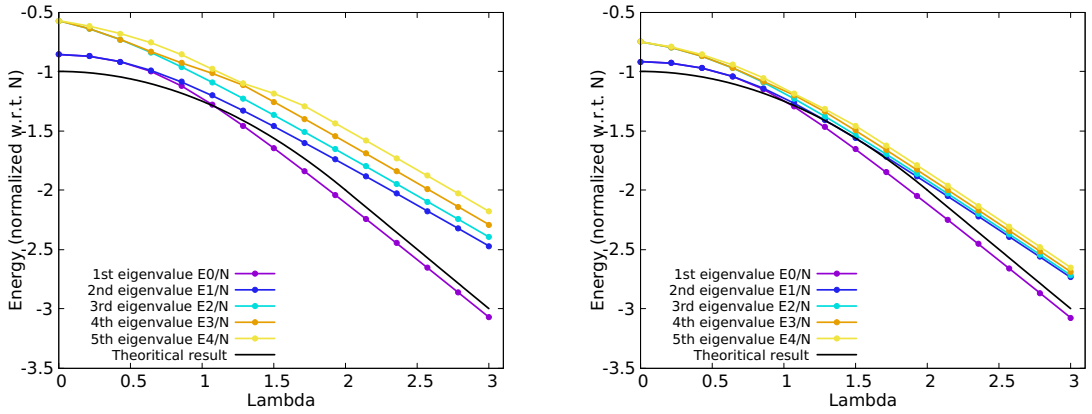

Figure 2: Plot of the first 5 energy levels vs λ with the theoretical result, for N=7 (left) and $N_{max} = 12$

One can note that the eigenvalues have a negative sign. This fact has no physical meaning since energy is defined up to a constant. At the ground state, corresponding to $\lambda = 0$, all the figures show a double degeneration. Physically λ represent the external field and if it is absent along the z-axis, the spins are aligned according to the x-axis. Moreover, at a certain point, close to $\lambda = 1$, the second eigenvalue (blue curve) tends to move away from the first one (purple curve) and cluster with the others . Indeed, for values of $\lambda > 1$ the theory states there is a quantum phase transition, while the *Mean Field* result predicts that this phase transition happens when $\lambda = 2$. Ideally, as N increases, the spectrum tends to become continuous.

Self-Evaluation

In this exercise I learned how to treat a composite quantum system described by an Ising Hamiltonian. Moreover I made a sense on how Hamiltonian's features reflect on the spectrum. In particular that when we look at the spectrum in function of λ a quantum phase transition happens. The work could be improved, maybe using a subroutine or something *ad hoc* ('built in') to perform the tensor product.