1D Quantum Ising model

Abstract

In this assignment, a system of N spin-$\frac{1}{2}$ particles is considered, with an Ising hamiltonian. The matrix representation of said hamiltonian is built, and the eigenvalues are computed for different values of N.

Theory

The subject of the analysis is a system of N spin-$\frac{1}{2}$ particles, with 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 σ are Pauli matrices, and λ is the interaction strength. For this hamiltonian, the coupling is anti-ferromagnetic. Indeed, a minus sign can be absorbed, for the first term, in λ; the interaction part, however, remains positive. Among the general properties of this hamiltonian, the most notable are its translational invariance, and the invariance with respect to spin flip (such a flip is carried out by applying σ^x).

In order to compute the matrix representation of this hamiltonian, one has to define the Hilbert space; in this case, it is $\mathcal{H} = \bigotimes_{i=1}^{N} \mathcal{H}_i$, where $\mathcal{H}_i$ is each site's Hilbert space. The resulting dimension of $\mathcal{H}$ is 2^N. Note that the hamiltonian is written as a sum of terms which, when taken on their own, are not defined on a space of dimension 2^N. Indeed, each matrix σ only acts on a Hilbert space of dimension 2; however, each term must be understood as only acting on the respective subsystem, leaving all the others untouched. A formally correct (but much heavier) notation would require each term to be recast as a tensor product, with a proper number of identity matrices for each of the subsystems on which the explicitly written terms act.

Once the Hilbert space is defined, a basis must be chosen. For each site i, one can associate a local basis $\mathcal{B}_i = \{|0\rangle, |1\rangle\}$ to $\mathcal{H}_i$. These local bases can then be composed into a single, global basis

$$\mathcal{B} = \left\{ |v\rangle = \bigotimes_{i=1}^{N} |\alpha_i\rangle_{\alpha=0,1} \right\} \tag{2}$$

As a consequence, the elements of $\mathcal{B}$ represent all possible combinations of all sites' states.

Once this basis has been chosen, $\hat{H}$ can be represented using it; indeed,

$$H_{mn} = \langle m|\hat{H}|n\rangle \tag{3}$$

At this point, the structure of the hamiltonian has to be taken into account in order to ease the computations. It can be decomposed into two parts: a diagonal term and an interaction one.

- **Diagonal part:** $\lambda \sum_{i=1}^{N} \sigma_z^i$

  As the name suggests, this term is diagonal (in the z basis, which is its eigenbasis):

$$\sigma_z^i = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{4}$$

  Indeed, only diagonal terms are preserved. When written more formally, the i-th term in the sum is actually

$$\left(\bigotimes_{j=1}^{i} \mathbb{1}_2 \right) \otimes \sigma_z^i \otimes \left(\bigotimes_{j=i+1}^{N} \mathbb{1}_2 \right); \tag{5}$$

  the tensor products between the identities and σ do not alter their diagonality.

- **Interaction term:** $\sum_{i=1}^{N-1} \sigma_x^i \sigma_x^{i+1}$

  Each of the terms in the summation is a (tensor) product of two matrices σ_x:

$$\sigma_x \otimes \sigma_x = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \tag{6}$$

  Similarly to the diagonal term, an appropriate 'padding' has to be performed, taking the tensor products with identity matrices.

The actual computation of the hamiltonian matrix can be carried out in two ways, which are however perfectly equivalent.

One can simply perform all the needed tensor products, starting from 2 by 2 matrices and building up to the 2^N hamiltonian. This requires a large number of intermediate steps, since every term in both sums has to be computed as a series of $\mathcal{O}(N)$ tensor products.

Alternatively, one can compute the effect of the hamiltonian operator on two elements of the basis $\mathcal{B}$, and then find a general rule for setting the (m, n) element of the matrix representation of $\hat{H}$. For the diagonal part:

$$\langle m|\lambda \sum_{i=1}^{N} \sigma_z^i |n\rangle = \lambda \sum_{i=1}^{N} \langle m|\sigma_z^i|n\rangle = \lambda \sum_{i=1}^{N} \langle \alpha_i|\sigma_z^i|\alpha_i'\rangle \prod_{j\neq i} \langle \alpha_j|\alpha_j'\rangle \tag{7}$$

where, in the last passage, the local bases have been used in place of the global one. The product is non-null only when $\alpha_j = \alpha_j'$ for all j. This is equivalent to only taking diagonal terms; if this is assumed, then

$$\langle \alpha_i|\sigma_z^i|\alpha_i'\rangle = \begin{cases} +1 & \text{if } \alpha_i = \alpha_i' = 0 \\ -1 & \text{if } \alpha_i = \alpha_i' = 1 \end{cases} \tag{8}$$

Since the values of α are linked to the base-2 encoding of m, this gives a formula for computing the value of the diagonal terms.

A similar procedure can be carried out for the interaction term. Indeed,

$$\langle m|\sum_{i=1}^{N-1}\sigma_x^i\sigma_x^{i+1}|n\rangle = \sum_{i=1}^{N-1}\langle m|\sigma_x^i\sigma_x^{i+1}|n\rangle = \sum_{i=1}^{N-1}\langle\alpha_i\alpha_{i+1}|\sigma_x^i\sigma_x^{i+1}|\alpha_i'\alpha_{i+1}'\rangle\prod_{j\neq i}\langle\alpha_j\alpha_{j+1}|\alpha_j'\alpha_{j+1}'\rangle \tag{9}$$

Unlike the diagonal case, here the matrix $\sigma_x^i\sigma_x^{i+1}$ does not select the pairings $\alpha_i = \alpha_i'$; instead, it acts as a logical `XOR` operation. Indeed,

$$\langle\alpha_i\alpha_{i+1}|\sigma_x^i\sigma_x^{i+1}|\alpha_i'\alpha_{i+1}'\rangle = \begin{cases}+1 & \text{if } \alpha_i \neq \alpha_i' \text{ and } \alpha_{i+1} \neq \alpha_{i+1}' \\ 0 & \text{otherwise}\end{cases} \tag{10}$$

Once again, this gives a formula, based on the base-2 encodings of the indexes (m, n).

The latter method has been preferred in the development of the code.

Code development

The code is organized as follows:

- `utils.f90`: a library of functions and subroutines that is extensively used in the actual source codes.
- `debug_module.f90`: a debugging module.
- `ex9.f90`: a program in which the matrix representation of the hamiltonian for a particular value of λ is computed and diagonalized, and the first k eigenvalues printed.
- `state_comp.f90`: a program that computes the k most likely configurations and their probabilities for varying values of λ.

The most interesting part of the exercise consists in computing the matrix representation of the hamiltonian. As was briefly outlined above, at least two strategies can be pursued. If one decides to just apply the definition, the hamiltonian can be computed as a sum of the interaction terms. These can be computed by preforming the Kronecker product between matrices, which is the matrix representation of the tensor product. Then, starting from the matrices

$$\sigma_z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}, \quad \sigma_x = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \quad \mathbb{1}_2 = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix},$$

each summand (in both the interaction and the diagonal part) can be created by iteratively performing the Kronecker product, starting from the right matrices up to the leftmost one. Then, after each of these matrices has been produced, it must be summed to the hamiltonian. This approach, while correct, is not efficient: indeed, for each of the $2N-1$ terms, one should allocate a matrix of size 2^N. The time needed to perform the Kronecker products and the memory requirements [1] make this method rather inefficient, especially for larger values of N.

[1] Indeed, keeping the $2^N \times 2^N$ hamiltonian matrix as an accumulator, while also having a temporary $2^N \times 2^N$ matrix where to store the next summand would effectively double the needed memory.

The method that was employed is instead based on the action of the hamiltonian itself on a generic pair of states, indexed by m and n. As specified before, this approach leads to formulas that can be used to quickly compute the matrix, by filling it element by element. The function that performs the operation is reported below.

```
  FUNCTION XOR(n,m,NN)
!!$  ------------------------------------------------------------
!!$    This function computes the matrix element of the quantum
!!$    1D Ising hamiltonian.
!!$    ARGUMENTS
!!$    - n    : an INTEGER*4, the first index
!!$    - m    : an INTEGER*4, the second index
!!$    - NN   : an INTEGER*4, the number of subsystems
!!$    RETURNS
!!$    - xor  : an INTEGER*4, the (m,n) matrix entry
!!$  ------------------------------------------------------------
    INTEGER*4 :: m,n,NN,kk,xor

    xor=0
    DO kk=1,NN-1
       IF(IEOR(m,n).EQ.(2**(kk-1)+2**kk))THEN
          xor=xor+1
       END IF
    END DO
    RETURN
  END FUNCTION XOR
```

The function is named `XOR` because of the way the matrix $\sigma_x \otimes \sigma_x$ acts. Indeed, whenever the binary representation of m and n differs only for two neighboring bits, the element (m, n) is to be added 1. As for the diagonal, no function was implemented, as the operation was short; it is reported here.

```
  ! CONSTRUCTING THE DIAG
!!$  ------------------------------------------------------------
  !creating the diagonal part of the hamiltonian;
  !These diagonal terms, which come from the non-interacting part
  !of the hamiltonian, are computed one at a time. The formula is
  !derived from the matrix element expression.
  diag=NN
  DO ii=0,2**NN-1
     DO jj=0,NN-1
        diag(ii+1)=diag(ii+1)-2*MOD(ii/(2**jj),2)
     END DO
  END DO
!!$  ------------------------------------------------------------
```

Results

The largest N_{MAX} that the used machine could manage was 14: at this value, the operative system issued a memory error before any computation could be carried out.

However, due to the lacking performance of the machine, the highest value that was actually used was $N = 12$; higher values required times larger than 1 full day.

The chosen number of eigenvalues was set to $k = 5$, as higher number of states were deemed to clutter the exposition. In the following, the (normalized with respect to N) eigenvalues for three different values of N are shown.

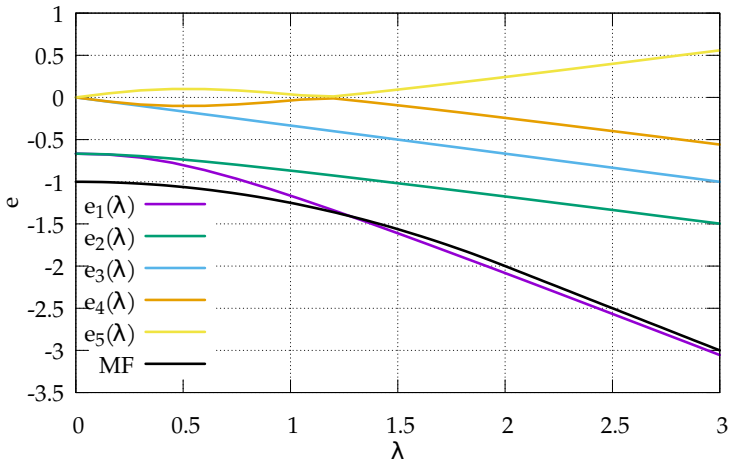

Figure 1: Spectrum for the first $k = 5$ eigenvalues. $N = 3$, λ ranging from 0 to 3 in 21 steps. An avoided level crossing is evident between e_4 and e_5.

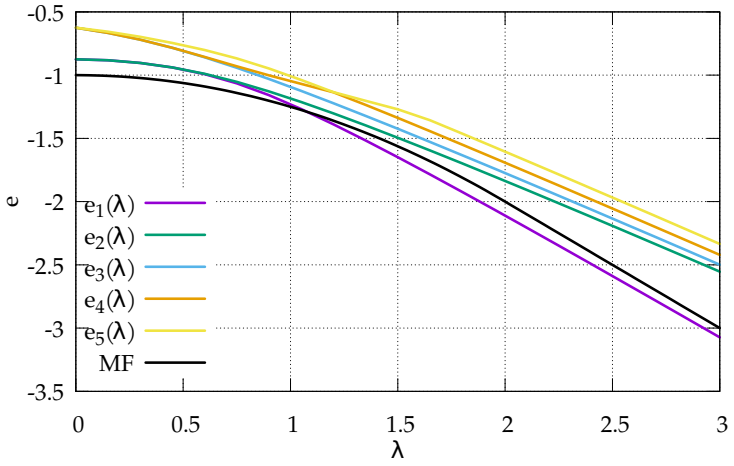

Figure 2: Spectrum for the first $k = 5$ eigenvalues. $N = 8$, λ ranging from 0 to 3 in 21 steps.

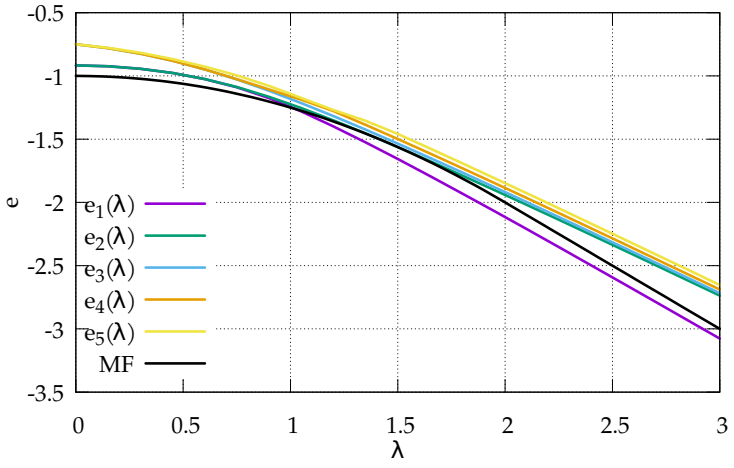

Figure 3: Spectrum for the first $k = 5$ eigenvalues. $N = 12$, λ ranging from 0 to 3 in 21 steps.

The most interesting feature of these plots lies in the degeneracy of the energy levels for $\lambda = 0$: indeed, in these cases, the only contribution to the energy is given by the interaction part, which has two equivalent minima (due to $\mathbb{Z}^2$ symmetry in the choice of the spins). The intervals between overlapping energies at $\lambda = 0$ clearly shows the discrete distribution of energies in a finite chain. Ideally, as $N \to \infty$, the spectrum would become continuous.

Another interesting phenomenon is visible in 1: an avoided energy level crossing. This happens close to $\lambda = 1$; for this value, two distinct energies approach the same value. A classical intuition would suggest that, in that point, due to $\mathbb{Z}^2$ symmetry, two distinct families of configurations (that is, sets of configurations with the same energy; this may be due, for instance, to translational invariance) have almost overlapping energies.

As λ increases, the states begin to split: the $\mathbb{Z}^2$ symmetry is broken due to the 'external field' component of the hamiltonian in λ, and, progressively, this component is overwhelming in dictating the energies. Indeed, for values larger than 1, the theory dictates that there is a quantum phase transition (while for the MF, the transition happens at $\lambda = 2$).

Furthermore, increasing the number of sites makes the ground state closer and closer to the mean field prediction. This is clearer in the following plot.

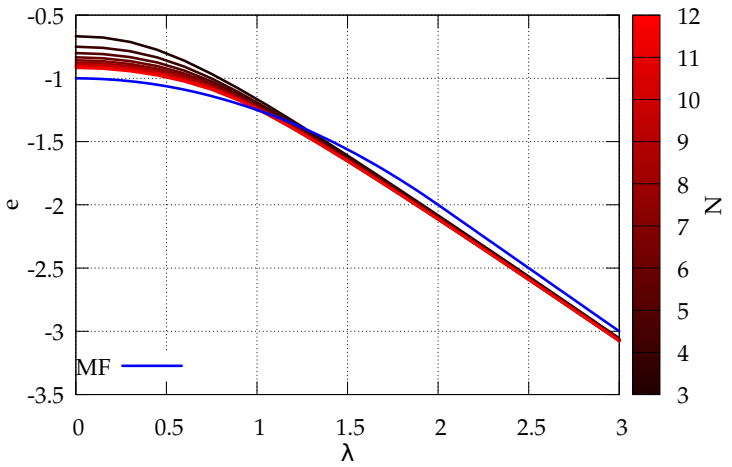

Figure 4: The ground state energies for N ranging from 3 to 12. λ ranging from 0 to 3 in 21 steps.

The 5 most probable configurations of $N = 8$ spins for varying values of λ have been computed, and are reported below. Results for additional values of λ can be found in the file `states.dat`.

Config.	Probability
11111001	7.8125e-03
00100010	7.8125e-03
01111110	7.8125e-03
11011000	7.8125e-03
01110100	7.8125e-03

Config.	Probability
11111111	6.0414e-01
11100111	3.9247e-02
11001111	3.9060e-02
11110011	3.9060e-02
10011111	3.8196e-02

Config.	Probability
11111111	9.5138e-01
11100111	6.7016e-03
11110011	6.7015e-03
11001111	6.7015e-03
10011111	6.6989e-03

Table 1: From left to right, $\lambda = 0$, $\lambda = 1.05$ and $\lambda = 3$.

For low values of λ, the interaction part dominates, and no one particular configuration seems to be favored. However, as λ increases, the homogeneous configuration becomes dominant, so much that its probability is two orders of magnitude greater than the next most probable. Even these, however, are mostly just translations of a common configuration, where pairs of spins are anti-aligned with the rest.

Comments and self evaluation

The final result of this assignment is consistent with the predictions. In particular, some interesting phenomena (avoided crossings, splitting) arise in the spectra of the energies, while the probabilities of the ground state are consistent with the intuition when λ increases.

Additional analysis may be carried out in order to study the properties of the density matrix of the ground state.

As a personal comment, this assignment proved indeed interesting, as most of the coding was simply organizing the functions implemented in the previous assignments.