Quantum Ising Model

Abstract

In this work we study a system composed of N spin-1/2 particles on a one-dimensional lattice with Ising-like Hamiltonian H. First we compute the matrix representation of H, then we diagonalize it and we study the first k energy levels.

1 Theory

Consider N spin-1/2 particles on a one-dimensional lattice, described by the Hamiltonian

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

where

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

are the Pauli matrices and λ is the interaction strength. Such Hamiltonian describes anti-ferromagnetic couplings.
In order to compute the matrix representation of H we define the Hilbert space $\mathcal{H} = \bigotimes_i^N \mathcal{H}_i$, which in our case has dimension 2^N. Given a local basis $\mathcal{B}_i = \{|0\rangle, |1\rangle\}$ for each local Hilbert space $\mathcal{H}_i$, we define the global basis

$$\mathcal{B} = \bigotimes_i^N \mathcal{B}_i. \tag{2}$$

Using this global basis we can represent the Hamiltonian's matrix elements as

$$H_{qp} = \langle q | H | p \rangle . \tag{3}$$

where $|p\rangle$ is the p-th global basis vector. However, notice that the Hamiltonian's terms do not act on the full Hilbert space, but on one of dimension 2. That is, they act only on one subsystem, leaving the others untouched. Let us first consider the diagonal term $\lambda \sum_i^N \sigma_z^i$. In our global basis we have

$$\sigma_z^i = \mathbb{1}^1 \otimes \cdots \otimes \mathbb{1}^{i-1} \otimes \sigma_z^i \otimes \mathbb{1}^{i+1} \otimes \cdots \otimes \mathbb{1}^N = \begin{pmatrix} \mathbb{1} & \mathbb{0} \\ \mathbb{0} & -\mathbb{1} \end{pmatrix}. \tag{4}$$

Moving on to the interaction part $\sum_i^{N-1} \sigma_x^i \sigma_x^{i+1}$, we have

$$\sigma_x^i \otimes \sigma_x^{i+1} = \mathbb{1}^1 \otimes \cdots \otimes \mathbb{1}^{i-1} \otimes \sigma_x^i \otimes \sigma_x^{i+1} \otimes \mathbb{1}^{i+2} \otimes \cdots \otimes \mathbb{1}^N. \tag{5}$$

Fortunately, we do not need to compute all these tensor products. We said that the Hamiltonian terms act only on a Hilbert space of dimension 2, so we evaluate their effect on the local basis $\mathcal{B}_i$. For the diagonal part we have

$$\begin{aligned} \langle q | \lambda \sum_i^N \sigma_z^i | p \rangle &= \lambda \sum_i^N \langle q | \sigma_z^i | p \rangle = \lambda \sum_i^N \left\langle \beta^i \right| \sigma_z^i \left| \beta'^i \right\rangle \prod_{j \neq i} \left\langle \beta^j \middle| \beta'^j \right\rangle \\ &= \lambda \delta_{\beta^j, \beta'^j} \cdot \sum_i^N \begin{cases} 1 & \text{if } \beta^i = \beta'^i = 0 \\ -1 & \text{if } \beta^i = \beta'^i = 1 \end{cases}, \end{aligned} \tag{6}$$

whereas for the interaction part we have

$$\langle q| \sum_i^{N-1} \sigma_x^i \sigma_x^{i+1} |p\rangle = \sum_i^{N-1} \langle q| \sigma_x^i \sigma_x^{i+1} |p\rangle = \sum_i^{N-1} \left\langle \beta^i \beta^{i+1} \right| \sigma_x^i \sigma_x^{i+1} \left| \beta'^i \beta'^{i+1} \right\rangle \prod_{j \neq i} \left\langle \beta^j \beta^{j+1} \middle| \beta'^j \beta'^{j+1} \right\rangle . \tag{7}$$

Notice that $\left\langle \beta^i \beta^{i+1} \right| \sigma_x^i \sigma_x^{i+1} \left| \beta'^i \beta'^{i+1} \right\rangle$ is 1 if $\beta^i \neq \beta'^i$ and $\beta^{i+1} \neq \beta'^{i+1}$, 0 otherwise. So, it acts like a XOR operator.

1.1 Mean field theory

Under the mean field assumption we can write

$$\left|\psi^{\mathrm{MF}}\right\rangle = \bigotimes_{i=1}^{N} \left|\psi^1\right\rangle , \tag{8}$$

and the energy of the system is given by

$$E^{\mathrm{MF}} = \left\langle \psi^{\mathrm{MF}} \right| H \left| \psi^{\mathrm{MF}} \right\rangle = \sum_{i=1}^{N-1} \left(\left\langle \psi^1 \right| \sigma_x^i \left| \psi^1 \right\rangle \right)^2 + \lambda \sum_{i=1}^{N} \left\langle \psi^1 \right| \sigma_z^i \left| \psi^1 \right\rangle . \tag{9}$$

The energy is extensive, so in the thermodynamic limit it diverges. Thus, we consider the energy density $e = E/N$, which is intensive and can be computed and minimized. The result is the following:

$$e^{\mathrm{MF}} = \begin{cases} -1 - \frac{\lambda^2}{4} & \lambda \in [-2, 2] \\ -|\lambda| & \text{otherwise.} \end{cases} \tag{10}$$

This expression is not exact, in fact it predicts a quantum phase transition to happen at $|\lambda| = 2$, whereas we know that it takes place for $|\lambda| = 1$[1].

2 Code development

The main program is structured as follows. We use two nested `for` loops: the first running on different values of N and the second on different values of λ. Inside the inner loop we initialize the Hamiltonian, we diagonalize it and we save the first k eigenvalues. Then, for each N, we write the results in different output files which are named according to the value of N.
The initialization of the Hamiltonian is carried out by means of the subroutine `IsingHamiltonian`, which constructs H by first computing the diagonal terms and then the iteraction ones. Then, the two terms are summed.

```
subroutine IsingHamiltonian(H, N, lambda)
    complex*16, dimension(:,:) :: H
    complex*16, dimension(:), allocatable :: diag
    integer :: N, ii, jj, info
    real*8 :: lambda
    allocate(diag(2**N), stat=info)
    if (info .ne. 0) then
        print *, "Allocation error!"
```

[1] https://paramekanti.weebly.com/uploads/1/1/2/8/11287579/atan_paper.pdf

```
        stop
    end if
    ! first fill the diagonal terms
    diag = N
    do ii = 0, 2**N-1
        do jj = 0, N-1
            diag(ii+1) = diag(ii+1) - 2*mod(ii/(2**jj), 2)
        end do
        diag(ii+1) = diag(ii+1) * lambda
    end do
    ! then compute the interaction terms
    do ii = 0, 2**N-1
        do jj = 0, 2**N-1
            H(jj+1, ii+1) = ComputeInter(jj, ii, N)
        end do
    end do
    ! add the two terms to find the complete Hamiltonian
    do ii = 1, 2**N
        H(ii, ii) = diag(ii) + H(ii, ii)
    end do
    deallocate(diag)
    return
end subroutine IsingHamiltonian
```

In order to compute the interaction term, we define the function `ComputeInter`, which returns 1 if q and p have only two different neighbour bits, i.e. if $\text{XOR}(q,p) = 2^i + 2^{i-1}$ for $i = 1, \dots, N-1$.

```
function ComputeInter(q, p, N) result(res)
    integer :: q, p, N, ii, res
    res = 0
    do ii = 1, N-1
        if ((2**(ii-1)+2**ii) == xor(q,p)) then
            res = res + 1
        end if
    end do
    return
end function ComputeInter
```

The diagonalization is done using the `Diagonalize` subroutine, which is a wrapper for the LAPACK `zheev` subroutine.

```
subroutine Diagonalize(matr, eig, info)
    complex*16, dimension(:, :), intent(in) :: matr
    real*8, dimension(size(matr, 1)) :: eig
    real*8, dimension(:), allocatable :: rwork
    complex*16, dimension(:), allocatable :: work
    character(1) :: jobz, uplo
    integer :: n, lda, lwork, info
    n = size(matr, 1)
    lda = size(matr, 1)
    jobz = "V"
    uplo = "U"
    allocate(rwork(max(1, 3*size(matr, 1)-2)))
    ! ------------ find the optimal lwork -----------------------------
    lwork = -1
    allocate(work(1))
    call zheev(jobz, uplo, n, matr, lda, eig, work, lwork, rwork, info)
    lwork = int(work(1))
    deallocate(work)
    ! -----------------------------------------------------------------
    ! allocate work using the optimal lwork
    allocate(work(lwork))
    ! perform the diagonalization
```

```
23      call zheev(jobz, uplo, n, matr, lda, eig, work, lwork, rwork, info)
24      deallocate(rwork, work)
25      return
26  end subroutine Diagonalize
```

Notice that inside `Diagonalize` we call `zheev` two times: the first to find the optimal parameter `lwork`, the second to perform the actual computation.

3 Results

The first test we do concerns the maximum value of N that we can use. In order to find that, we fix the value of λ (e.g. $\lambda = 1$) and we loop only over N calling the subroutine `cpu_time` first before allocating H and then after the diagonalization. The plot of the CPU time as a function of N is shown in Figure 1. The maximum value we can get is $N = 14$, for which the computation takes approximately six hours and a half.

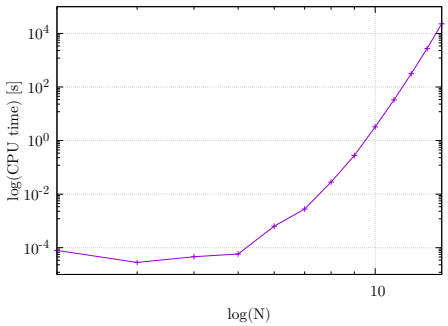


Figure 1: CPU time to allocate, initialize and diagonalize H for different values of N and fixed λ.

First, we study the ground state energy density $e = \frac{E}{N-1}$ as a function of λ for different values of N. We divide the energy by $N - 1$ because we are not in the thermodynamic limit, so we must consider a finite size correction that takes into account the fact that the spins at the boundaries "loose 1/2 interaction" each one. The results are shown in Figure 2, in which we also plot the mean field energy density. We see that, as we expect, as N increases the results get closer to the mean field solution.

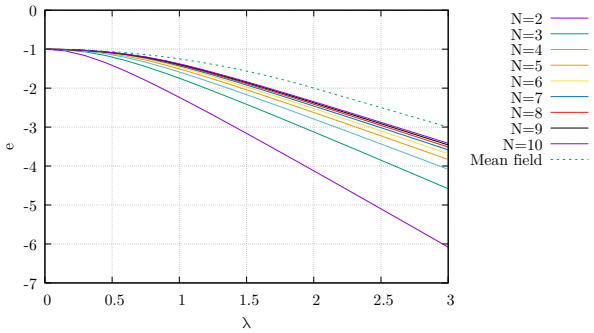
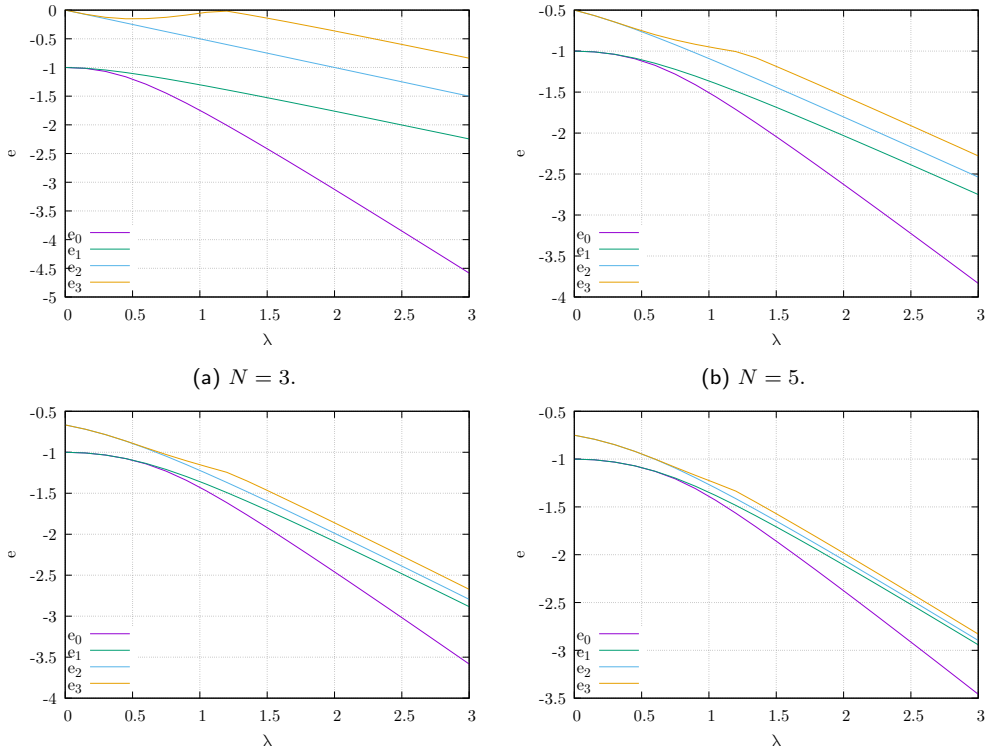


Figure 2: Ground state energy density as function of λ for different values of N.

Then, we consider the value of the energy density for the first k levels as a function of λ, for

different values of N. The results for $k = 4$ and $N = 3, 5, 7, 9$ are shown in Figure 3. Notice that for $\lambda = 0$ the ground state is degenerate. In fact, if the external field is everywhere zero the energy is given only by the interaction term, which has $\mathbb{Z}^2$ symmetry: this means that it is invariant under switching the value of the spin in all the lattice sites. In this case, the energy has two equivalent minima (the x components of the spins are alternated and the spin chain can begin either with $+1$ or -1). A non-zero field breaks this symmetry, in fact by increasing λ we see that the degeneracy disappears. When this happens, we have a quantum phase transition. By looking at the plot we see that, as expected, by increasing N (i.e. getting "closer" to the thermodynamic limit) this phase transition occurs for a value of λ closer and closer to the theoretical one, $\lambda = 1$.



(a) $N = 3$.

(b) $N = 5$.

(c) $N = 7$.

(d) $N = 9$.

Figure 3: First $k = 4$ eigenvalues for different values of $N = 3, 5, 7, 9$.

4 Self evaluation

This exercise was very instructive since it made me learn how to simulate a quantum Ising model, also helping me to understand the theory.
One possible improvement could be to better manage the outputs.