Quantum information and computing: Exercises report, week 10. Quantum Ising Model and Real Space Renormalization Group

Alessandro Lambertini
(Dated: January 9, 2021)

Through the exercise of this week, we study from a numerical perspective the 1-D quantum Ising model in a transverse field, but this time by means of the RSRG algorithm. We implement the algorithm and we study the ground state of the system for different values of the interaction strenght λ.

I. THEORY

The Hamiltonian that describes properly our system is the same of last week assignment:

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

where σ_z and σ_x are Pauli's matrices and the two terms under the summation signs are expanded as follows:

$$\sigma_z \otimes \mathbb{1}_2 \otimes \cdots \otimes \mathbb{1}_N + \mathbb{1}_1 \otimes \sigma_z \otimes \mathbb{1}_3 \otimes \cdots \otimes \mathbb{1}_N + \cdots + \mathbb{1}_1 \otimes \mathbb{1}_2 \otimes \cdots \otimes \sigma_z \tag{2}$$

$$\sigma_x \otimes \sigma_x \mathbb{1}_3 \otimes \cdots \otimes \mathbb{1}_N + \mathbb{1}_1 \otimes \sigma_x \otimes \sigma_x \mathbb{1}_4 \otimes \cdots \otimes \mathbb{1}_N + \cdots + \mathbb{1}_1 \otimes \mathbb{1}_2 \otimes \cdots \otimes \sigma_x \otimes \sigma_x \tag{3}$$

We push our system to the thermodynamic limit using the Real Space Renormalization Group algorithm. RSRG is an iterative algorithm that, at each iteration, allows to double the size of the system while keeping the dimension of the hamiltonian the same through all the iterative process. It works as follows:

- Write the hamiltonian of our original system: $H_N : \mathbb{C}^{2N} \to \mathbb{C}^{2N}$, where N is the number of particles in our system.
- Built up the doubled system H_{2N}:

$$H_{2N} = H_A + H_B + H_{AB} \tag{4}$$

  where:

$$H_A = H_N \otimes \mathbb{1} \tag{5}$$
$$H_B = \mathbb{1} \otimes H_N \tag{6}$$
$$H_{AB} = H_{left} \otimes H_{right} \tag{7}$$
$$\tag{8}$$

  and

$$H_{left} = \mathbb{1}_1 \otimes \cdots \otimes \mathbb{1}_{N-1} \otimes \sigma_x \tag{9}$$
$$H_{right} = \sigma_x \otimes \mathbb{1}_1 \otimes \cdots \otimes \mathbb{1}_{N-1} \tag{10}$$

- Diagonalize the new hamiltonian.
- construct the projection operator P by taking the first 2^N eigenvectors found after the diagonalization
- project H_{2N} into the lower dimensional space defined by P: $H'_N = P^T H_{2N} P$
- update also H_{left} and H_{right}:

$$H'_{left} = P^T (H_{left} \otimes \mathbb{1}) P \tag{11}$$
$$H'_{right} = P^T (H_{right} \otimes \mathbb{1}) P \tag{12}$$

- repeat the last five point of the list.

We will compare, qualitatively, our solutions with the one analitically obtained under the mean field assumptions, that is:

$$\begin{cases} E_0 = -1 - \frac{\lambda^2}{4} & \lambda \in [-2:2] \\ E_0 = -|\lambda| & \lambda \notin [-2:2] \end{cases} \tag{13}$$

II. CODE DEVELOPMENT

In order to accomplish the tasks this time I had all the instruments ready to be used. In fact, i wrote all the subroutines needed during the previous weeks exercises. I exploited them from the *ex10_module* written in the file *ex10_module.f90*. The main program is located in the file *ex10.f90* and it is reported below:

```
!!!!!!!!!!!!!!!!! tensor product of two complex matrices !!!!!!!!!!!!!!!!!
program ex10
  use ex10module
  implicit none

!declare the variable
complex(kind=8), dimension(:,:), allocatable :: identity, initial_system, &
H_a,h_b,H_l,H_r,H_ab,H_2N,H_2N_copy,P, H_l_prov,H_r_prov
real(kind=8), dimension(:), allocatable :: eigv,lambda_i,ground_states
real(kind=8) :: step,lambda_max
real(kind=8) :: norm_term
integer :: N, ii,jj,M,iter

!Create the lambda array
M = 100
lambda_max = 3
allocate(lambda_i(M),ground_states(M))

step = lambda_max/real(M-1,8)

do ii=1,M
  lambda_i(ii) = 0+(ii-1)*step
end do

!set parameters
N = 2
iter = 20
norm_term =  real(2,8)**iter

!initialize identities
allocate(identity(2**N,2**N))
do ii=1,2**N
  identity(ii,ii) = (1,0)
end do

!cycle over lambdas
do ii=1,M
!initialize the hamiltonian of the two subsystems
call H_ising_np_tr(N,lambda_i(ii),initial_system)

call sigma_x_i(N,N,H_l)
call sigma_x_i(N,1,H_r)
!Real Space Renormalization Group
do jj=1,iter

  call tensor_product(initial_system,identity,H_a)
  call tensor_product(identity,initial_system,H_b)
  call tensor_product(H_l,H_r,H_ab)

  H_2N = H_a + H_b + H_ab

  H_2N_copy = H_2N
```

```
  call diag_hermit_matrix(H_2N,eigv)

  P = H_2N(:,1:2**N)

  initial_system = MATMUL(TRANSPOSE(CONJG(P)),MATMUL(H_2N_copy,P))

  call tensor_product(H_l,identity,H_l_prov)
  call tensor_product(identity,H_r,H_r_prov)

  H_l = MATMUL( TRANSPOSE(CONJG(P)),MATMUL(H_l_prov,P))
  H_r = MATMUL( TRANSPOSE(CONJG(P)),MATMUL(H_r_prov,P))

  deallocate(H_a,H_b,H_ab,eigv,P,H_2N,H_l_prov,H_r_prov,H_2N_copy)

end do

ground_states(ii) = eigv(1)/norm_term/N

deallocate(initial_system)

end do

call write_arrays_on_file('ground_states.txt',lambda_i,ground_states)

end program ex10
```

III. RESULTS

Below i report the result obtained running the above program. The ground states for different lambdas are obtained starting from a system with 2 spins and doubled 50 times.

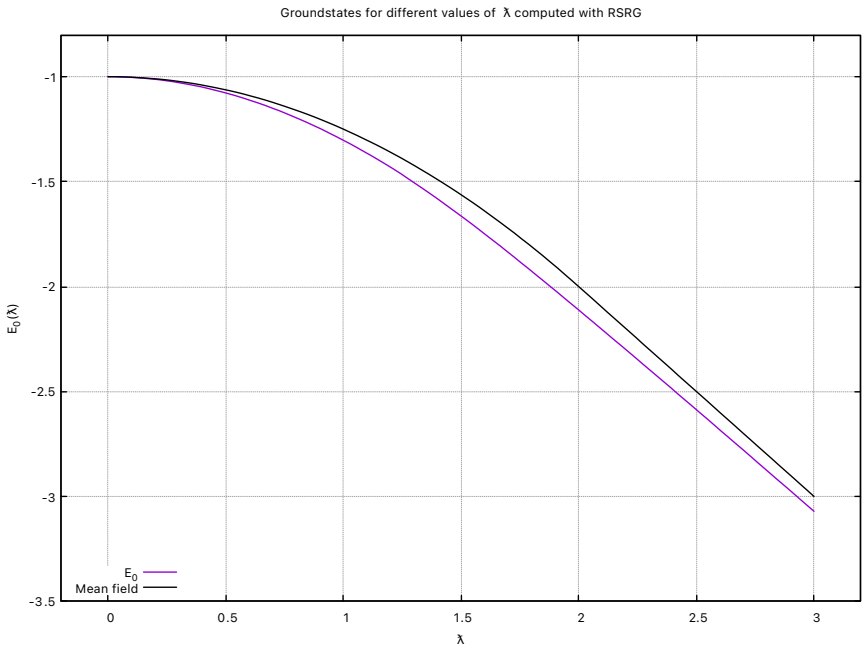

FIG. 1. The ground states of a system of spins at thermodynamical equilibrium as a function of λ computed by means of the RSRG algorithm

IV. SELF-EVALUATION

I think the main objectives of the exercise are reached. I have learnt how to implement the RSRG algorithm to analyse the ground state of chains of spins with different number of particles.