Week 10: Real-space Renormalization Group Algorithm

Alice Pagano

(Dated: January 9, 2021)

In this Report, given the quantum Ising Hamiltonian in transverse field on a one-dimensional lattice with nearest neighbour interaction, we compute the ground state energy as a function of the transverse field λ by means of the real-space RG algorithm.

I. THEORY

A. Quantum Ising Model

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:

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

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

In order to solve with numerical simulation this model, let us remind that the Pauli matrices can be rewritten in an explicit form as:

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

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

B. Real-space Renormalization Group Algorithm

Real-space renormalization group (RG) method is an approximation method based on a very powerful physical intuition: the hypothesis that the ground state of a system is composed of low-energy states of the system's (non-interacting) bipartitions. It is an iterative process to build a system, or a description of a system, of size $2N$ from a truncated description of system of size N. The algorithm, for the Hamiltonian in Eq. (1), proceed as follows:

1. Consider a system composed of N sites and build the Hamiltonian $\hat{H}_N : \mathbb{C}^{d^N} \to \mathbb{C}^{d^N}$.
2. Construct the Hamiltonian of a system of size $2N$ using the Hamiltonian $\hat{H}_N$ for each bipartition and the interaction among them, as:
$$\hat{H}_{2N} = \hat{H}_N \otimes \mathbb{1}_N + \mathbb{1}_N \otimes \hat{H}_N + \hat{H}_{int} \tag{3}$$
where $\hat{H}_{int}$ can be obtained as $\hat{H}_{int} = \hat{A}_N \otimes \hat{B}_N$ where $\hat{A} = \mathbb{1}_2 \otimes \sigma^x$ and $\hat{B} = \sigma^x \otimes \mathbb{1}_2$ are the operator acting on each system bipartition.
3. Diagonalize $\hat{H}_{2N}$, finding its eigenvalues and eigenvectors. Compose a matrix P whose columns are the 2^N lowest eigenvectors of $\hat{H}_{2N}$.
4. Compute the projected Hamiltonian $\hat{H}_N^{trunc} = P^\dagger \hat{H}_{2N} P$ as well as any other needed operator representation in the projected space. In particular, the projected operator acting on each system bipartition are $\hat{A}' = P^\dagger A P$ and $\hat{B}' = P^\dagger B P$.
5. Repeat the steps 1–4 with the following substitutions
$$\hat{H}_N \to \hat{H}_N^{trunc}, \quad A' \to \hat{A}', \quad B' \to \hat{B}'$$
until the desired system size is reached or convergence to the renormalization group fixed point is achieved. Notice that, at each step of the algorithm, the dimension of the Hamiltonian representation is kept constant to N.

II. CODE DEVELOPMENT

In order to write the hamiltonian of the Ising model for a system with N particles and compute its ground state by means of the RG algorithm, we develop a Fortran program.

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`. Then, the main steps of the program are:

1. The total number of subsystems `N` and the number of iterations of the RG algorithm `iter` are given as input. Subsequently, we also create an array of λ values in the range $[-5; 5]$ and we define the matrices σ_x and σ_z.

2. Then, we loop over the values of λ. For each λ, we initilize the system hamiltonian as the sum between a non-interacting term `H_non_int` and an interacting one `H_int`. 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).
   - 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).

   Moreover, we initilize also the A and B operators as the tensor product between the identity matrix and σ_x.

```
! Initialization of system Hamiltonian and of operators
H = lambda(ii)*H_non_int(N_part,sigmaz) + H_int(N_part,sigmax)
A = tensor_product(identity(N_part-1),sigmax)
B = tensor_product(sigmax,identity(N_part-1))
```

3. For each λ and for `iter` times, we call `SUBROUTINE` **`real_space_RG`** `(HN, N, A, B)` to perform the RG algorithm. First of all, the Hamiltonian of size $2N$ is initialized as in Eq. (3). Then, we diagonalize $\hat{H}_{2N}$ and we construct the projected matrix P and its adjoint $P^\dagger$. At least, the Hamiltonian $\hat{H}_{2N}$ and the operators A and B are projected.

```
subroutine real_space_RG(HN, N, A, B)
    ...
    ! Set dimension
    dim = 2**N

    allocate( H2N(dim**2,dim**2)     )
    allocate( A_int(dim**2 ,dim**2)  )
    allocate( B_int(dim**2 ,dim**2)  )
    allocate( P(dim**2,dim)          )
    allocate( Pdag(dim,dim**2)       )

    ! Initialize H2N Hamiltonian
    H2N = tensor_product(HN, identity(N)) +tensor_product(identity(N), HN) + tensor_product(A,B)

    CALL diag_hamiltonian(H2N, eig_val, eig_vec)

    ! Initialize P and Pdag (adjoint)
    do ii = 1, dim, 1
        P(:,ii) = eig_vec(:,ii)
    end do
    Pdag = transpose(conjg(P))

    ! Update H2N hamiltonian
    HN = matmul(matmul(Pdag, H2N),P)

    ! Update A_int and B_int
    A_int = tensor_product(A,identity(N))
    B_int = tensor_product(identity(N),B)
    A = matmul(matmul(Pdag,A_int),P)
    B = matmul(matmul(Pdag,B_int),P)

end subroutine real_space_RG
```

Moreover, before each iteration, we divide by a 2 factor the Hamiltonian H_N and by $\sqrt{2}$ the A and B operators in order to keep the numbers low.

```
! Perform RG algorithm
do jj=1,iter,1
    ! Divide per 2 each cycle to keep the numbers low
    H = H * 0.5
    A = 1/sqrt(2.) * A
    B = 1/sqrt(2.) * B
    ! Call RG algorithm
    CALL real_space_RG(H, N_part, A, B)
end do
```

4. Finally, for each λ, we diagonalize the final Hamiltonian $\hat{H}_N$ and we find the ground state energy density $G_0 = E_0/N$.

The ground state energy $G_0(\lambda)$ is plotted as a function of λ thanks to a gnuplot script.

III. RESULTS

The program is executed for simulating a system of $N = 2$ particles with the interaction strength varing in the range $\lambda \in [-5; 5]$. More specifically, the RG algorithm is executed for $N_{iter} = 100$.

The results of the simulation are illustrated in Fig. 1. In this plot, the RG method outcomes are compared with the one of the Mean Field solution:

$$G_0^{MF}(\lambda) = \begin{cases} -1 - \lambda^2/4 & \lambda \in [-2:2] \\ -|\lambda| & \lambda \notin [-2:2] \end{cases} \tag{4}$$

We note that there is a good agreement between the two solutions.

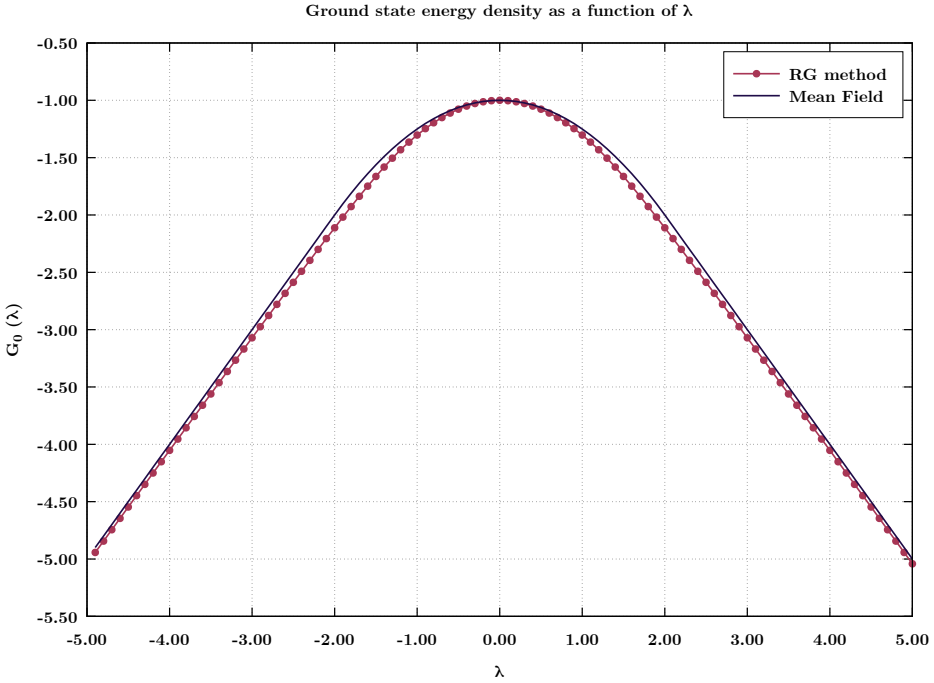

FIG. 1 Ground state energy $G_0(\lambda)$ of a quantum Ising model as a function of λ for $N = 2$ particles. Both the Mean Field solution and the results of RG algorithm are reported.

IV. SELF-EVALUATION

The code implemented works well and returns consistent results. In particular, we learn how to compute the ground state energy of a quantum Ising model by using the RG algorithm and we study its spectrum for different values of the strength parameters λ. In a further development of the code, it could be useful to compute the error associated to each RG iteration in order to find the optimal iteration number. Moreover, the tensor product function should be optimized to simulate also more complex systems.