Quantum information and computing: Report 10

Nicola Lonigro
University of Padua
(Dated: January 10, 2021)

The Transverse filed Ising model was solved using the Real-space Renormalization Group and the ground state energy was computed as a function of the coupling strength λ

I. THEORY

A. Transverse field Ising model

The transverse field Ising model is a quantum version of the classical Ising model. It features a lattice with nearest neighbour interactions determined by the alignment or anti-alignment of spin projections along the z axis, as well as an external magnetic field perpendicular to the z axis which creates an energetic bias for one x-axis spin direction over the other.

An important feature of this setup is that, in a quantum sense, the spin projection along the x axis and the spin projection along the z axis are not commuting observable quantities. That is, they cannot both be observed simultaneously. This means classical statistical mechanics cannot describe this model, and a quantum treatment is needed.

In particular we can describe the 1D model with the Hamiltonian

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

Where λ is the coupling strength and $\sigma^{(i)}$'s are the Pauli matrices referring to the i-th spin in the chain. In particular this will be given by the application of the operator only on the relevant spin, while leaving the others unchanged and it is equivalent to the tensor product of the respective Pauli operator with identity operators for all the other spins.

B. Renormalization Group

In theoretical physics, the term renormalization group (RG) refers to a formal apparatus that allows systematic investigation of the changes of a physical system as viewed at different scales. In this case the algorithm consists in:

- Replicating a starting system of size N to obtain a system of size 2N with the respective Hamiltonian
- Diagonalize the new Hamiltonian
- Projecting the new system in the space spanned by the eigenvectors corresponding to the N lowest eigenvalues

By iterating this procedure enough times, it is possible to effectively estimating the ground state of an infinite chain of spins, under the assumption that the information relating to this configuration can be propagated through the algorithm by keeping only the lowest eigenvalues

II. CODE DEVELOPMENT

The code uses the same structure of previous exercises. In this case there are 5 inputs: the number of spins, the coupling strength for the first and second term, the number of excited levels to write to file and the number of steps to iterate the RN algorithm for. In practice the constant of the second term has always been left equal to one during the exercise but it is present to have a more general code.
The Hamiltonian of the starting system is initialized using the same function as the previous exercise, then a new function has been implemented to loop the RN algorithm

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!Fill Hermitian matrix
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  CALL Def_Hamiltonian(N,H,couple_1,couple_2)
  CALL CHECKPOINT(D_FLAG,"After Initialization",
    N,H,couple_1,couple_2,egvals)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Renormalization group loop
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  CALL RG_iter(N,H,couple_2,N_steps)
  CALL CHECKPOINT(D_FLAG,"After Initialization",
    N,H,couple_1,couple_2,egvals)
```

The new function will take the Hamiltonian of the starting system,the number of iterations and the coupling constant as an input. Before starting the loop it will initialize two helper tensors that are later used in the definition of the interaction Hamiltonian for the first replica of the system. In particular the two matrices correspond to the tensor products of identity matrices for all the spins but the right-most one and the left-most one respectively, for which the Pauli matrix σ_x is used instead.

```
  !Define Initial interaction terms
    DO i = 1,2**N
        IF (i > 2**(N-1) ) THEN
            int_1(i,i-2**(N-1)) = int_1(i,i-2**(
    N-1)) + 1
        ELSE
```

```
            int_1(i,i +2**(N-1)) = int_1(i,i
+2**(N-1)) + 1
        END IF
    END DO

    DO m=1,2**(N-1)
      DO i = 1,2
        IF (i > 1 ) THEN
          int_2(i+(m-1)*2,i -1 +(m-1)*2) = int_2
(i+(m-1)*2,i-1+(m-1)*2) + 1
        ELSE
          int_2(i+(m-1)*2,i +1 +(m-1)*2) = int_2
(i+(m-1)*2,i+1+(m-1)*2) + 1
        END IF
      END DO
    END DO
```

As done for the definition of the initial Hamiltonian, the matrices are not created using the full tensor product expression but exploiting the properties of the identity matrix to initialize directly the matrix in the correct dimension. The Hamiltonian for the new system is then initialized and the three terms corresponding to the Hamiltonian of the first system, the replica system and the interaction term are computed separately and summed

```
    !RG iteration loop
    DO step = 1,N_steps
      H_tot = 0
      int_1_tot = 0
      int_2_tot = 0

      DO i = 1,2**N
        DO j = 1,2**N
          DO k = 1,2**N
            H_tot((i-1)*2**N+j,(i-1)*2**N+k) =
H_tot((i-1)*2**N+j,(i-1)*2**N+k) + H(j,k)  !
 Tensor Product of H and ID
            int_1_tot((i-1)*2**N+j,(i-1)*2**N+k)
 = int_1_tot((i-1)*2**N+j,(i-1)*2**N+k) +
int_1(j,k)
          END DO
        END DO
      END DO

      DO i = 1,2**N
        DO j = 1,2**N
          DO k = 1,2**N
            H_tot((i-1)*2**N+k, (j-1)*2**N+k) =
H_tot((i-1)*2**N+k, (j-1)*2**N+k) + H(i,j)
!  Tensor Product of ID and H
            int_2_tot((i-1)*2**N+j,(i-1)*2**N+k)
 = int_2_tot((i-1)*2**N+j,(i-1)*2**N+k) +
int_2(j,k)
          END DO
        END DO
      END DO


      DO i=1, 2**N
        DO j=1, 2**N
          DO k=1, 2**N
            DO m=1, 2**N
              H_tot( (i-1)*2**N+k,(j-1)*2**N+m)
=   H_tot((i-1)*2**N+k,(j-1)*2**N+m) +
coupling_2*int_1(i,j)*int_2(k,m) !
Interaction term
            END DO
          END DO
        END DO
      END DO
```

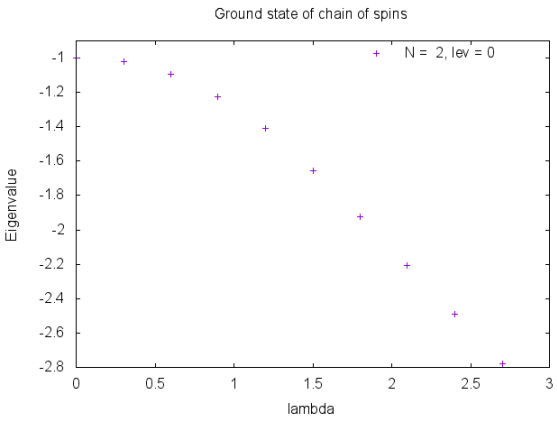

FIG. 1. Ground state energy density for infinite chain of spins in transverse field Ising model

```
      END DO
```

The new Hamiltonian is then diagonalized, the first N eigenvectors are taken and the starting Hamiltonian is overwritten with the projection of the new Hamiltonian in the subspace of dimension N via matrix multiplication. The loop is then repeated for the set amount of time. Notice that it is required to create a copy of the new Hamiltonian before diagonalizing it as the zheev function will overwrite the matrix is given as an input. Notice also that the Hamiltonian is divided by 2 at each iteration to end up with a size-independent energy density.

```
      H_copy = H_tot
      CALL zheev('V','U',dim,H_tot,dim,egvals,
    WORK,LWORK,RWORK,INFO)
      CALL FATAL(INFO /= 0, "main", "Eigenvalues
     not found")

      P = H_tot(:,1:2**(2*N-1))
      P_dag = TRANSPOSE(CONJG(P))
      coupling_2 = coupling_2*0.5
      H = 0.5*MATMUL(MATMUL(P_dag,H_copy),P)
      int_1 = MATMUL(P_dag,MATMUL(int_1_tot,P))
      int_2 = MATMUL(P_dag,MATMUL(int_2_tot,P))
```

After the selected number of iterations, the final Hamiltonian is diagonalized once again and the corresponding eigenvalues are written to file. The script used also in previous versions of the code was modified to scan over λ.

III. RESULTS

The code has been run scanning values of λ in the interval [0:3]. The resulting ground state energy density is reported in Fig.1. The energy density has been used instead of the ground state energy to have a result indepenent on the number of iterations of the RG algorithm.

IV. SELF-EVALUATION

An implementation of the RG algorithm has been added to the code. An even more generalizable code could be implemented to work with any operator acting on only one spin, however this would not exploit the favorable properties of the Pauli matrices such as the diagonal form of the σ_z. An even faster code could be achieved by looking for a direct implementation of the interaction Hamiltonian without passing from the full definition of the tensor product which requires 4 loops, however this does not seem to be relevant as the code can be run in just a few seconds and so the requirement for a fast computation is less stringent then when using the brute-force method to diagnonalize a large Hamitonian.