Report: Week 8

Quantum Information and Computing (2021/22)

Prof. Simone Montangero

Samuele Piccinelli

Università degli Studi di Padova

11 January 2022

Transverse-field Ising model (TFI)

We consider a linear chain of N interacting spin-1/2 particles in presence of an external field of intensity λ.
The problem is given in an Hamiltonian $\hat{H}_N : \mathbb{C}^{2^N} \to \mathbb{C}^{2^N}$,

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

The notation simplifies the one of a **tensor product** that can be written as

$$\sigma_i^z = \mathbb{1}_1 \otimes \cdots \mathbb{1}_{i-1} \otimes \sigma_i^z \otimes \mathbb{1}_{i+1} \otimes \cdots \mathbb{1}_N$$
$$\sigma_i^x \sigma_{i+1}^x = \mathbb{1}_1 \otimes \cdots \mathbb{1}_{i-1} \otimes \sigma_i^x \otimes \sigma_{i+1}^x \otimes \mathbb{1}_{i+2} \otimes \cdots \mathbb{1}_N ;$$

λ is the **interaction strength parameter** and the σs are the **Pauli matrices**.

In order to get the ground state (GS) of $\hat{H}_N$ we exploit the **real-space renormalization group** (RSRG) algorithm.

The latter is based on the hypothesis that the GS of a system is composed of the low-energy states of its non-interacting bipartitions.

RSRA algorithm

The algorithm consists of the following steps:

1. We consider a system of N sites with a $\left(2^N \times 2^N\right)$ Hamiltonian $\hat{H}_N$.

2. We **double** the system size to consider a system of $2N$ particles and build its Hamiltonian as

$$\hat{H}_{2N} = \hat{H}_N \otimes \mathbb{1}_N + \mathbb{1}_N \otimes \hat{H}_N + \hat{H}_{\text{int}}$$
$$\hat{H}_{\text{int}} = \underbrace{H^L}_{\left(\bigotimes_{j=1}^{N-1} \mathbb{1} \otimes \sigma^x\right)} \otimes \underbrace{H^R}_{\left(\sigma^x \otimes \bigotimes_{j=1}^{N-1} \mathbb{1}\right)} .$$

3. We **diagonalize** $\hat{H}_{2N}$: we build P as a matrix whose columns are the N lowest eigenvectors of $\hat{H}_{2N}$.

4. We **project** the Hamiltonian, the left and right interaction matrices,

$$H_N = P^\dagger H_{2N} P \qquad\qquad H^{L/R} = P^\dagger H^{L/R} P.$$

5. We **iterate** steps 1 to 4 for n_{iter} times until $\mathcal{E}_0^{n+1} - \mathcal{E}_0^n < \varepsilon$ with ε fixed threshold. The **energy density per site** is then given by

$$\frac{\mathcal{E}_0^{n_{\text{iter}}}}{N \cdot 2^{n_{\text{iter}}}}.$$

Listing 1: Implementation of the iteration step of the RSRG algorithm.

```
! Initialize double space Hamiltonian
ham_2N = kronecker_product_c(ham, identity_c(N)) + &
         kronecker_product_c(identity_c(N), ham) + &
         kronecker_product_c(ham_L, ham_R)
! Diagonalize double space Hamiltonian
call diag_mat(ham_2N, ham_2N_eigvec, ham_2N_eigval)
gs = ham_2N_eigval(1) / N
! Build P and adj(P) matrices
P     = ham_2N_eigvec(:,:2**N)
P_adj = TRANSPOSE(CONJG(P))
! Project Hamiltonian
ham = MATMUL(MATMUL(P_adj, ham_2N), P)
! Project H_L and H_R
ham_L = MATMUL(MATMUL(P_adj, kronecker_product_c(ham_L, identity_c(N))), P)
ham_R = MATMUL(MATMUL(P_adj, kronecker_product_c(identity_c(N), ham_R)), P)
```

Listing 2: Main program loop.

```
do while(abs(e_next - e_prec) > eps)
    e_prec = e_next

    ham   = ham   / 2.0d0
    ham_L = ham_L / SQRT(2.0d0)
    ham_R = ham_R / SQRT(2.0d0)

    call rsrg_step(N, ham, ham_L, ham_R, e_next)
    Niter  = Niter + 1
end do
```

The program is executed for $N = 2,\ 3,\ 4$, $\lambda \in [0, 3]$ and $\varepsilon = 10^{-10}$. A **Python script** is used to run it for different sets of input parameters.

We observe that the number of iterations n_{iter} decreases for increasing values of λ.

The **ground state energy density** as a function of λ is shown in Fig. 1: the results found are compared with the **mean-field** (MF) theory predictions,

$$\mathcal{E}_0^{\text{MF}}/N = \begin{cases} -1 - \lambda^2/4 & \lambda \in [-2, 2] \\ -|\lambda| & \text{otherwise.} \end{cases} \tag{2}$$

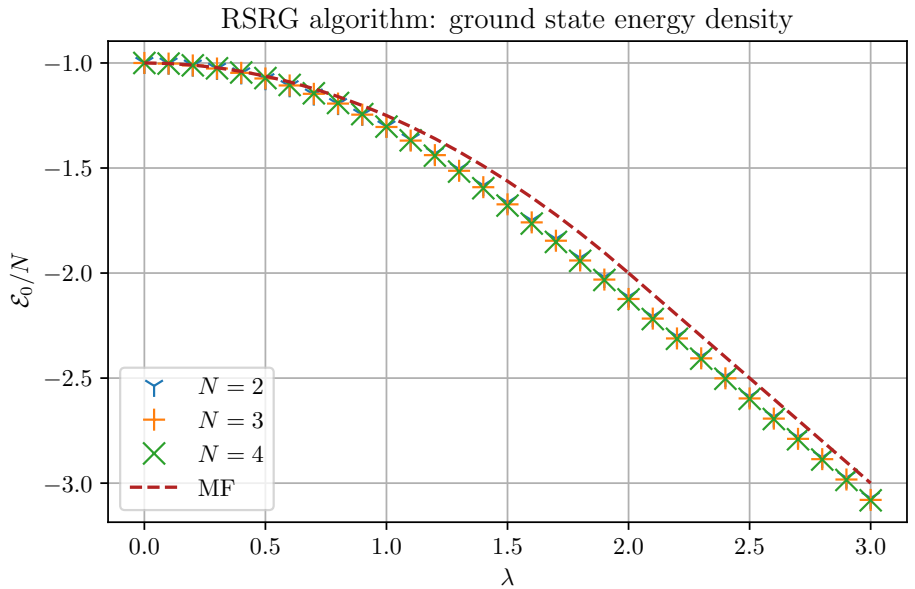

Figure 1: Ground state energy density per site as a function of λ and MF solution.