



EXERCISE 6

RENORMALIZATION GROUP

QUANTUM INFORMATION AND COMPUTING COURSE 2021/2022

ALESSANDRO MARCOMINI (2024286)

PROF. SIMONE MONTANGER

01/11/22

EXERCISE GOALS

Given the quantum Ising Hamiltonian in transverse field on a one-dimensional lattice with nearest neighbor interaction:

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

Compute the ground state energy as a function of the transverse field λ by means of the real-space RG algorithm.

RG IDEA

- Consider a N -size one-dim system described by a given Hamiltonian (TFI)
- Consider the $2N$ -size system described by the concatenation of the previous ones
- Diagonalize such Hamiltonian and approximate it as the projection onto the N -size subspace of minimal energies, then iterate (cheap simulation of larger systems)

IN THEORY...

For spin $1/2$ particles in the chain, the TFI Hamiltonian scales as $n = 2^N$ (exact diagonalization of 10+ particles system is already computational demanding...)

Real-field RG allows to simulate chains of millions of particles by keeping only the most relevant information of the Hamiltonian low-energy states. In practice, the simulation size N is defined and kept constant while at each iteration the full-system diagonal Hamiltonian is approximated by the projection onto the N -dimensional subspace of lower energy. This procedure allows to double the size of the simulated chain at each iteration.

Theoretically speaking, we encounter the thermodynamical limit for $N \rightarrow \infty$. In this scenario the energy per spin is constant. We see that RG allows to display this behaviour.

CODE DEVELOPMENT

- Create first four components:

$$H_{TFI}^L = H_{half} \otimes 1_n, H_{INT}^L = 1_{n-1} \otimes \sigma^X$$

and same for R

- Compose them to create 2N-system Hamiltonian
- Diagonalize it, store first n eigenvectors in M matrix
- Perform matrix-matrix multiplication to project into low-energy eigenspace

```
do while (.NOT. thermod_limit)

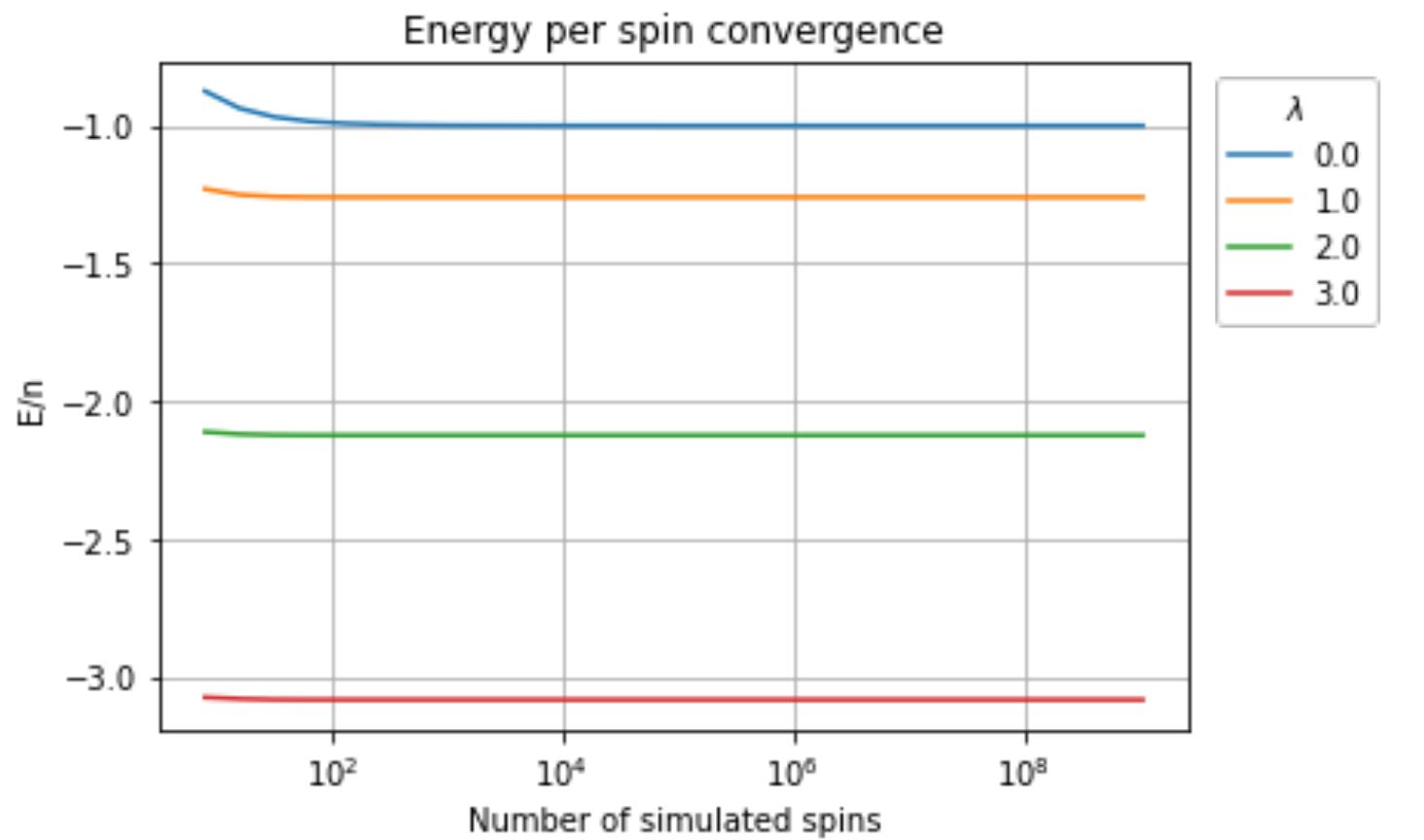
    ! Construct full system Hamiltonian
    H_int = TensorProduct(H_int_L,H_int_R)
    H_tot = TensorProduct(H_half,Identity(space_dim))
    + TensorProduct(Identity(space_dim),H_half) + H_int
    A = H_tot

    ! Diagonalize
    call zheev( 'Vectors', 'Lower', space_dim**2,
    |     |     A, LDA, W, WORK, LWORK, RWORK, INFO)
    if (INFO/=0) then...
    end if

    ! Check Energy
    new_E = W(1)/(2**iii*N)
    thermod_limit = abs((new_E - old_E)/old_E) < eps

    ! Update subsystem parts
    M = A(:,1:space_dim)
    H_half = matmul(adjoint(M), matmul(H_tot,M))
    H_int_L = matmul(adjoint(M),
    |     |     matmul(TensorProduct(Identity(space_dim),H_int_L),M))
    H_int_R = matmul(adjoint(M),
    |     |     matmul(TensorProduct(H_int_R,Identity(space_dim)),M))
    H_int = TensorProduct(H_int_L,H_int_R)
    old_E = new_E
```

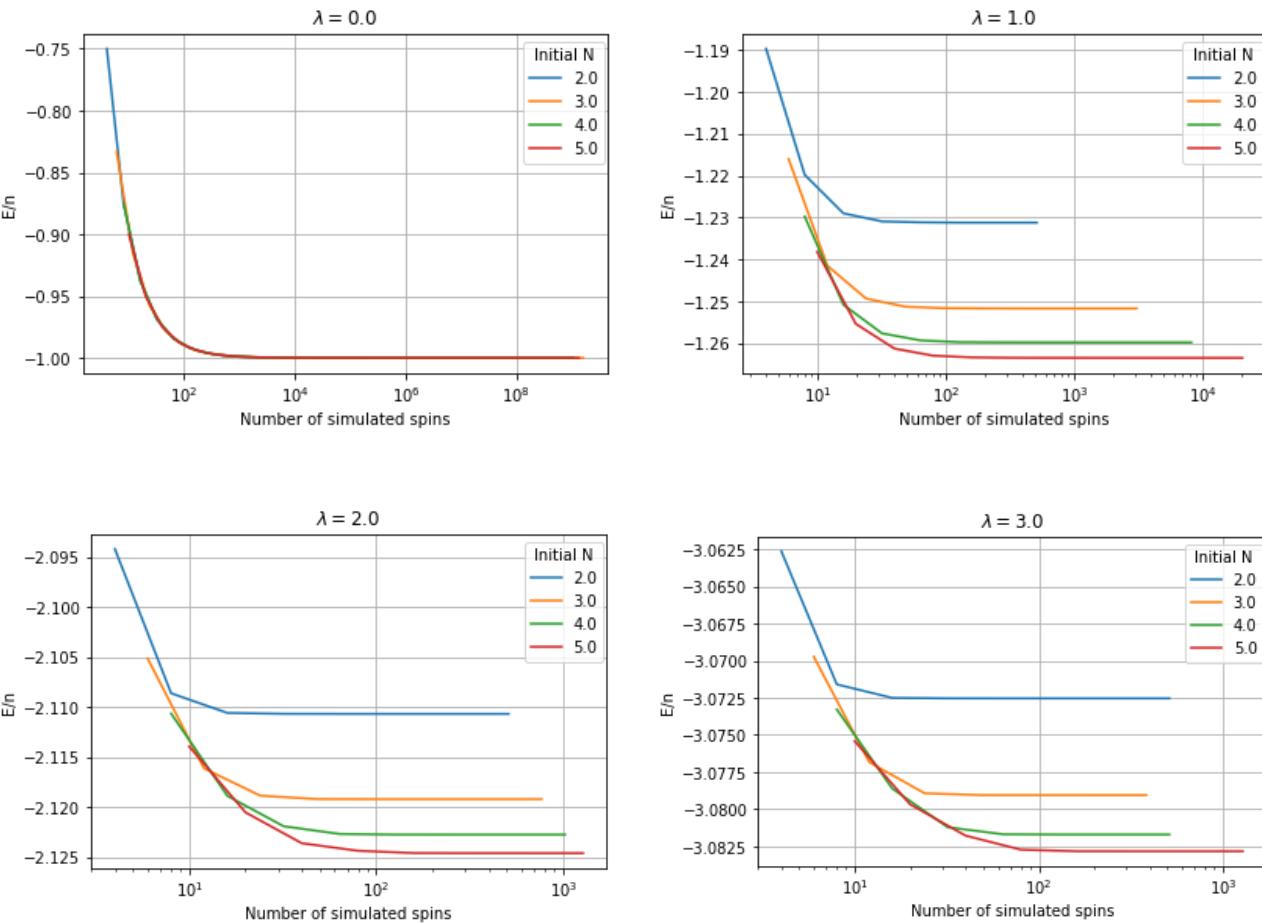
TREND OF ENERGY PER SPIN ENLARGING THE SYSTEM



*initial N=4

DEPENDENCY ON INITIAL SUBSYSTEMS SIZES (N)

- Perfect alignment for $\lambda = 0$
(as expected: hamiltonian composed of single-particle operators)
- Different values of convergence by changing N for larger λ
(variations of $O(0.1\%)$)
- Number of iterations required before convergence* is independent on N



*Convergence: relative change in energy per spin $< 10^{-9}$