Renormalization Group

Saverio Monaco

Quantum Information and Computing

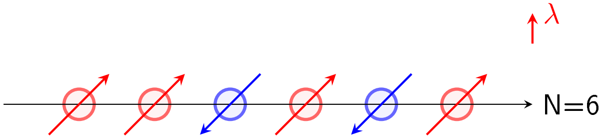
Dipartimento di Fisica e Astronomia "Galileo Galilei"

Università degli Studi di Padova

December 17, 2021

Theory

1-D Ising Model:



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

where

$$\sigma_z^i = \underbrace{\mathbb{I} \otimes \mathbb{I} \otimes \ldots \otimes \mathbb{I}}_{i-1} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \underbrace{\mathbb{I} \otimes \ldots \otimes \mathbb{I}}_{N-i}$$

$$\sigma_x^i = \underbrace{\mathbb{I} \otimes \mathbb{I} \otimes \ldots \otimes \mathbb{I}}_{i-1} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \underbrace{\mathbb{I} \otimes \ldots \otimes \mathbb{I}}_{N-i}$$

Algorithm:

1. Initialize Ising's Hamiltonian for a given N: H_N
2. Double the system size:

$$H_{2N} = H_N \otimes \bigotimes_{i=1}^{N} \mathbb{I} + \bigotimes_{i=1}^{N} \mathbb{I} \otimes H_N + \left[\bigotimes_{j=1}^{N-1} \mathbb{I} \otimes \sigma^x \right] \otimes \left[\sigma^x \otimes \bigotimes_{j=1}^{N-1} \mathbb{I} \right]$$

3. Diagonalize H_{2N} and build the projector P using the first 2^N eigenvalues.
4. Reduce the 2N-Hamiltonian:

$$\tilde{H}_{2N} = P^\dagger H_{2N} P \qquad dim[\tilde{H}_{2N}] = 2^N$$

5. Iterate:

$$H_{2N}^{(1)} = \tilde{H}_{2N} \otimes [\mathbb{I}]_N + [\mathbb{I}]_N \otimes \tilde{H}_{2N} + \tilde{H}_{int}$$

$$\tilde{H}_{int} = P^\dagger \left[\bigotimes_{j=1}^{N} \mathbb{I} \otimes H_{int}^{L} \right] P \otimes P^\dagger \left[H_{int}^{R} \otimes \bigotimes_{j=1}^{N} \mathbb{I} \right]$$

Code development

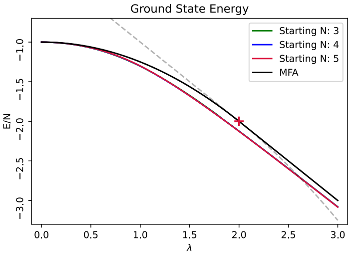
Before looping:

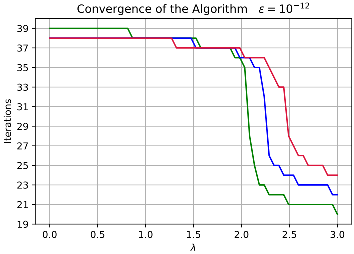
```
! to initialize the first hamiltonian (system size = N)
function ising_init_H(N,lambda) result(H)

! to initialize the 2 matrices for the interactions between
! 2 subsystems of system size = N
call init_interaction_H(N,HL,HR)
```

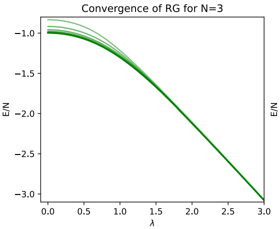
At each iteration:

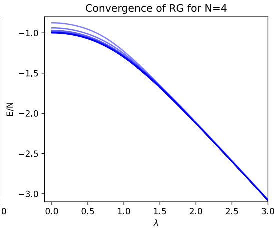
```
H2N = mat_tensor_I(HN) + I_tensor_mat(HN) + tens_prod(HL,HR)

call diagonalize_H(H2N, evls, 2**N, P)

HLred = tens_prod(HL, identity(2**N))
HRred = tens_prod(identity(2**N), HR)
call project(P, H2N, HN)
call project(P, HLred, HL)
call project(P, HRred, HR)
```

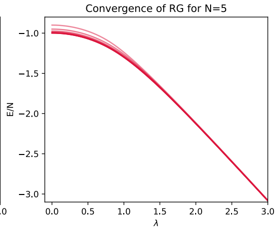
Results


Ground State Energy
Starting N: 3
Starting N: 4
Starting N: 5
MFA
E/N
λ


Convergence of the Algorithm $\varepsilon = 10^{-12}$
Iterations
λ


Convergence of RG for N=3
E/N
λ


Convergence of RG for N=4
E/N
λ


Convergence of RG for N=5
E/N
λ

Self evaluation

Correctness: The algorithm seems to give the proper values of the Ground-State energy for each λ according to last exercise.

Stability: The algorithm gives a `-Infinity` result for the energies at around 80 iterations due to the overflow for the integer `sizeofspace`.

Efficiency: `double precision` arrays were used to use as little memory as possible, for the diagonalization it was used `DSYEVR` function from the Lapack library.