Real Space Renormalization Group

Abstract

In this assignment, a quantum Ising model of N spin-$\frac{1}{2}$ particles is considered, and it is treated using the *Real Space Renormalization Group (RSRG)* technique.

Theory

The subject of the analysis is a system of N spin-$\frac{1}{2}$ particles, with hamiltonian

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

where σ are Pauli matrices, and λ is the interaction strength. For this hamiltonian, the coupling is anti-ferromagnetic. Indeed, a minus sign can be absorbed, for the first term, in λ; the interaction part, however, remains positive. Among the general properties of this hamiltonian, the most notable are its translational invariance, and the invariance with respect to spin flip (such a flip is carried out by applying σ^x). This system will be studied by means of *Real Space Renormalization Group (RSRG)*.

In RSRG, a 1-D system has its size N augmented by a factor 2: the N sites are replicated, and the two copies are placed one beside the other. For a system of size N, the hamiltonian is $\hat{H}_N : \mathbb{C}^d \to \mathbb{C}^d$, while for a system of size $2N$, it is $\hat{H}_{2N} : \mathbb{C}^{d^2} \to \mathbb{C}^{d^2}$. Instead of writing the hamiltonian matrix of the increased system as one single block, two matrices are created, one for each of the two halves of the total system; each of them is written as the exact hamiltonian of the respective spin block, disregarding the other one completely. An additional term is then added, to take the interaction between the two parts of the system into account. Indeed, denoting the whole system of $2N$ spins as LR, one can write

$$\hat{H}_{2N}^{LR} = \hat{H}_N^L + \hat{H}_N^R + \hat{H}_{int}^{LR} \tag{2}$$

where $\hat{H}_N^i$ are the hamiltonian matrices for the two halves, and $\hat{H}_{int}^{LR}$ is the interaction term, that connects the two parts. Of course, this notation hides the fact that an appropriate number of identity operators is to be taken for each of the three terms, as each of the hamiltonians over one half of the whole system must act on the other half without any effect, and the interaction term must only act on the concerned sites.

In the case at hand, the interaction term connects the rightmost spin of the left subsystem with the leftmost spin of the right system, and it is a simple tensor product of two σ_x operators. As for the hamiltonians over the two halves, they are the exact hamiltonians for an Ising chain of N spins.

Once the hamiltonian has been written in this way, one can diagonalize it as

$$\hat{H}_{2N}^{LR} = \sum_i E_i \left|E_i\right\rangle \left\langle E_i\right| \tag{3}$$

The resulting set of eigenvalues and eigenvectors represent the energies of the system and the corresponding states respectively. Taking advantage of a physical intuition, if one

wants to study the ground state of a given system, only the lowest eigenvalues need to be taken into account. As a consequence, limiting the sum above to the first d values will only discard terms which have high energy, and which do not play a significant role in defining the ground state.

The hamiltonian can be effectively truncated using this idea: defining the projector $\hat{P}$ as the truncated change of basis (derived from the diagonalization of $\hat{H}_{2N}$ itself), the truncated hamiltonian is

$$\hat{H}^{tr}_{2N} : \mathbb{C}^d \to \mathbb{C}^d, \quad \hat{H}^{tr}_{2N} = \hat{P}^\dagger \hat{H}_{2N} \hat{P} \tag{4}$$

One can then employ the truncated version of the hamiltonian in further computations. Indeed, the whole process can be repeated multiple times, as the dimension of the truncated hamiltonian never exceeds d.

For the sake of clarity, the whole process for the 1D quantum Ising model is reported here:

1. Write the hamiltonian for a system of N spins, $\hat{H}_N$; this will be used for both the left ($\hat{H}^L_N$) and right ($\hat{H}^R_N$) parts;
2. Write the hamiltonian for a $2N$ system as
$$\hat{H}_{2N} = \hat{H}^L_N \otimes \left(\bigotimes_{i=1}^{N} \mathbf{I}_2 \right) + \left(\bigotimes_{i=1}^{N} \mathbf{I}_2 \right) \otimes \hat{H}^R_N + \hat{L}_{int} \otimes \hat{R}_{int} \tag{5}$$
where $\hat{L}_{int} = \left(\bigotimes_{i=1}^{N-1} \mathbf{I}_2 \right) \otimes \sigma_x$ and $\hat{R}_{int} = \sigma_x \otimes \left(\bigotimes_{i=1}^{N-1} \mathbf{I}_2 \right)$ are the left and right components of the interaction term;
3. Diagonalize $\hat{H}_{2N}$;
4. Using the eigenvectors associated to the lowest N eigenvalues, compute the projector $\hat{P}$ as the $\left(2^N \times N\right)$ shaped matrix the columns of which are the selected eigenvectors;
5. Use $\hat{P}$ to project the hamiltonian:
$$\hat{H}^{tr}_{2N} = \hat{P}^\dagger \hat{H}_{2N} \hat{P} \tag{6}$$
6. Compute the upscaled left and right interaction terms:
$$\hat{L}'_{int} = \hat{P}^\dagger \left(\bigotimes_{i=1}^{N} \mathbf{I}_2 \otimes \hat{L}_{int} \right) \hat{P}, \quad \hat{R}'_{int} = \hat{P}^\dagger \left(\hat{R}_{int} \otimes \bigotimes_{i=1}^{N} \mathbf{I}_2 \right) \hat{P} \tag{7}$$
7. Repeat the procedure from step 2, substituting the $\hat{H}^L$ and $\hat{H}^R$ hamiltonians with the truncated $\hat{H}^{tr}$ from step 5, and L_{int}, R_{int} with L'_{int}, R'_{int}. Reiterate for as many times as needed. Since the size increases exponentially, few steps are enough to produce meaningful results.

This method allows to expand the numerical study of the ground state well beyond the naive technique. Indeed, the results of this method can be compared with the predictions of the mean field solution (provided that the energy is scaled by the total number of sites in the last iteration of the algorithm).

Code development

The code is organized as follows:

- `utils.f90`: a library of functions and subroutines that is extensively used in the actual source codes. For instance, functions for projecting matrices or computing Kronecker products.
- `debug_module.f90`: a debugging module.
- `ex10_RG.f90`: a program in which the RSRG method is implemented. It prints out the ground state energy for a given value of λ.

Most of the code from the previous assignment was reused in this one, as the creation of the hamiltonian matrix follows the exact same principles. The only noteworthy addition comes in the form of the projection and the Kronecker multiplication of identity matrices of various sizes. Since many tensor products between identity operators are present in the algorithm, and since the Kronecker product of an arbitrary number of identity matrices is still an identity matrix (of appropriate size), a function that returns an identity matrix of arbitrary size was implemented. In addition to the avoidance of expensive Kronecker products, this also makes for much more readable code.

Results

The algorithm was run for $n = 100$ iterations, and the starting number of sites was set to 2, 3, and 4. For all values of N, a matrix of size $2N$ has to be diagonalized at each step (as part of the doubling step in the RSRG algorithm). This limits the possible values (as 7 is the largest possible N such that the hamiltonian can be stored in the RAM), but also the feasible ones. Indeed, for N larger than 4, the diagonalization step takes a lot of time.

The results for various N are shown, together with the mean field prediction. As in the previous homework, λ goes from 0 to 3, this time in 100 steps. The results for negative λ are identical, due to the symmetry of the problem.

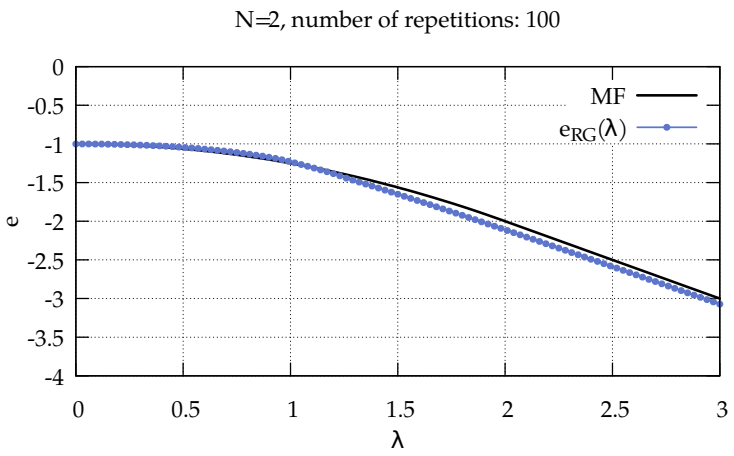

Figure 1: $N = 2$, $n = 100$.

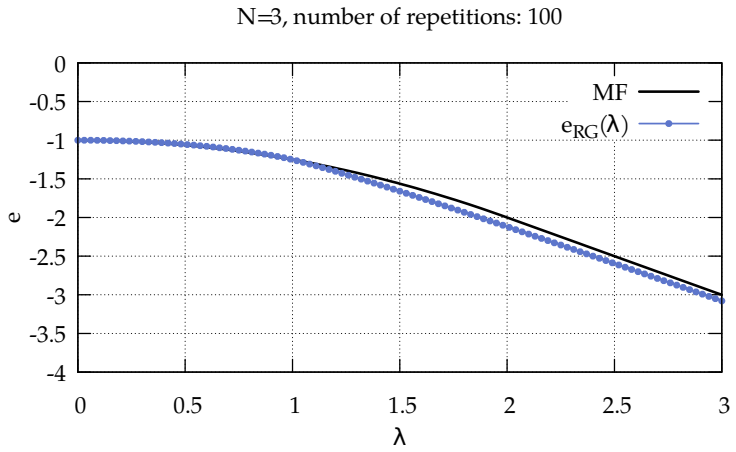

Figure 2: $N = 3$, $n = 100$.

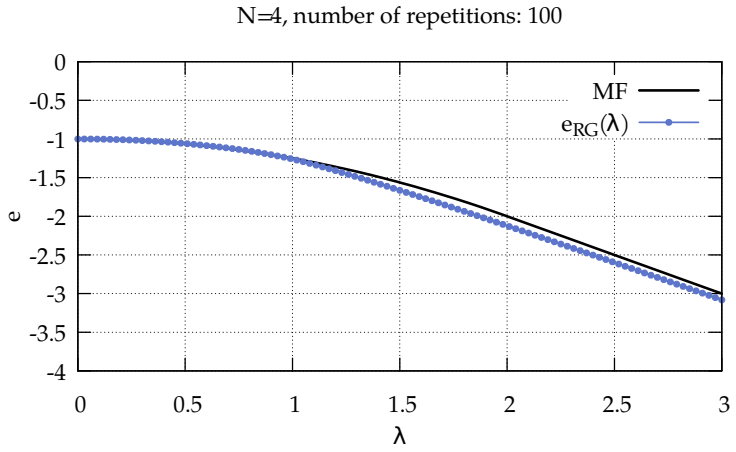

Figure 3: $N = 4$, $n = 100$.

The results for the different N are similar enough that they are not easily discernible just by comparison. Indeed, in Fig. 4, all the curves are shown together, so to make their similarity more evident.

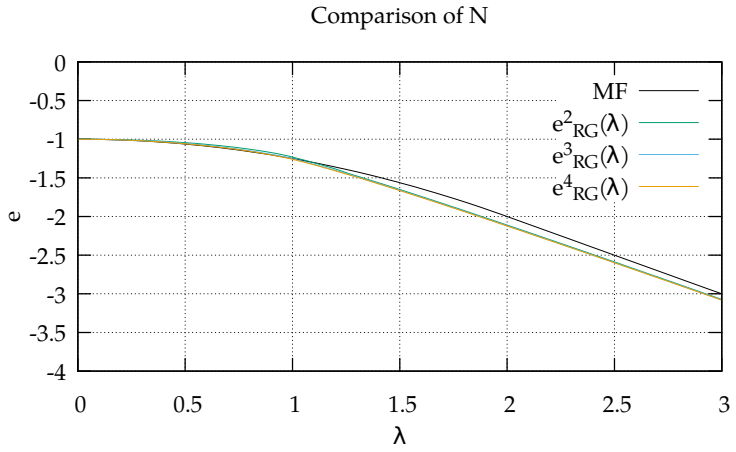

Figure 4: The ground state, computed via RSRG for $N = 2, 3, 4$ and $n = 100$. The mean field prediction is also shown.

Referring to the previous homework, it seems that also in this case increasing the

number of overall sites results in the computed ground state to get closer to the mean field prediction in the region for which $\lambda \sim 0$.

The differences of the mean field prediction and the RSRG solutions are also shown in Fig. 5.

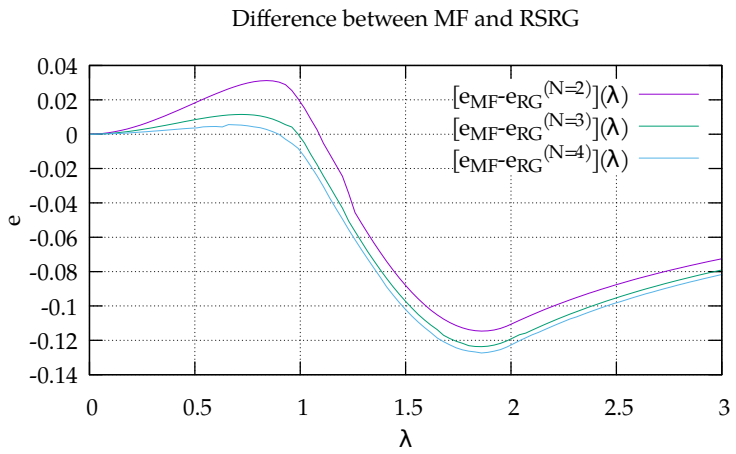

Figure 5: Plot of the differences between the RSRG predictions and the mean field ones.

This plot highlights the fact that, for increasing N, the RSRG results are more and more in line with those predicted by the mean field when $\lambda \sim 0$, while the opposite is true for larger values of λ. Resorting to classical reasoning (without claiming rigor), this may be due to the fact that the mean field assumption only holds when the external field is sufficiently small, so much so that the approximation of uncorrelated sites becomes more and more valid.

A final check was performed in order to check whether these slight differences are intrinsically due to the starting number N, or if they can be explained by the fact that the final number of spins for the various N is vastly different. Indeed, after n iterations of the algorithm, if the starting number of sites is N, the final spin chain will be composed of $N2^{(n+1)}$, so there should only be at most a factor 2 difference between the results for $N = 2$ and $N = 4$.

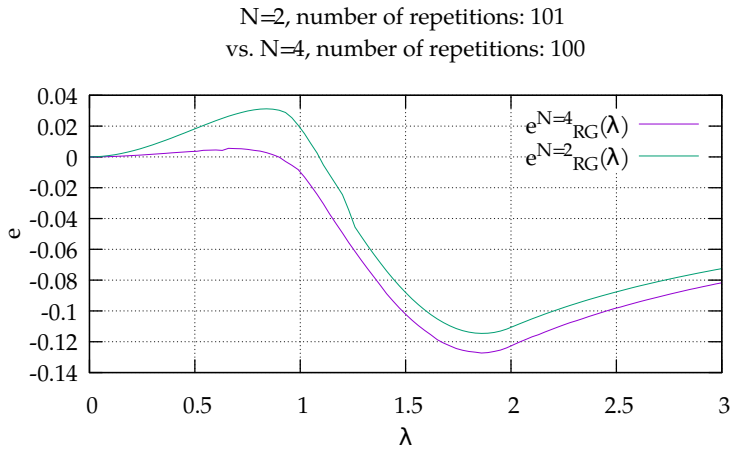

Figure 6: A comparison of the difference between the mean field prediction and the result from the RSRG algorithm. Adding one iteration to the $N = 2$ version does not actually make it equal to the $N = 4$ version.

The result shown in Fig. 6 seems to suggest that the best course of action is to perform the RSRG algorithm by using the largest feasible value of N.

An additional comment is devoted to the failed attempt at implementing the *infinite density matrix renormalization group (DMRG)*. The source code is provided, but despite the fact that it compiles (an executable is also provided), the results that it produces are far from correct. In particular, the resulting curve is not at all smooth, and significantly diverges from the mean field one (which, despite it not being correct, should nevertheless provide at least a general outline of the solution).

Comments and self evaluation

The final result of this assignment is consistent with the predictions. The trend that was observed in the previous assignments for increasing values of N appears once again. Furthermore, some inner intricacies of the RSRG method are shown, such as its dependence on the initial value of N.

As a personal comment, this assignment was very interesting and satisfying. Indeed, defying the hardware limits by means of a software trick such as this is a great computational achievement. A negative experience instead comes from the failure in implementing the DMRG method, which eluded all my attempts.