Computational Quantum Physics
Exercise 3

PHYS-463
Prof. G. Carleo

Problem 3.1 Exact diagonalization of the transverse field Ising Hamiltonian

The most accurate method for solving a quantum many-body problem is exact diagonalization of the Hamiltonian matrix. In order to make the most out of the computational resources available, it is very helpful to make use of all available tools for simplifying the problem.

In this exercise, we consider the transverse field Ising model in 1D, given by the Hamiltonian

$$\hat{H}_{\text{Ising}} = J \sum_{i=1}^{N} \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z - \Gamma \sum_{i=1}^{N} \hat{\sigma}_i^x, \tag{1}$$

where we used periodic boundary conditions, $\sigma_{N+1} = \sigma_1$. This model shows (for $N \to \infty$) a quantum phase transition at $\Gamma/J = 1$. To see the onset of this transition, we need to find the ground state while tuning Γ (keeping $J = 1$ as the unit of energy).

The goal of this exercise is to represent H_{Ising} as a sparse matrix and diagonalize it with (a variant of) the Lanczos algorithm provided in scipy as well as the power method.

In the following we have $\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

a) When we write $\hat{\sigma}_i^x$ in Eq. (1) we mean that we act with $\hat{\sigma}^x$ on the i-th site and with identities everywhere else: $\hat{\sigma}_i^x = \mathbb{I} \otimes ... \otimes \sigma^x \otimes ... \otimes \mathbb{I}$. This operator corresponds to a $2^N \times 2^N$ matrix. We can use `scipy.sparse.kron` successively to construct these matrices. Write a method that returns (given the number of spins N) a list where the i-th entry is the sparse matrix representation (`scipy.sparse.csr_array`) of $\hat{\sigma}_i^x$. Do the same with all the $\hat{\sigma}_i^z$.
Hint: Notice that the identity matrix itself should be stored consistently as a sparse matrix. You can declare a sparse identity matrix of size L using `scipy.sparse.csr_array(scipy.sparse.identity(L))`.

b) Write a function that constructs the Ising Hamiltonian given the list of all $\hat{\sigma}_i^x$ and $\hat{\sigma}_i^z$ generated in a).
Hint: For two matrices A and B in the `csr_array` format, matrix multiplication can be done simply by `A @ B`. The symbol `@` denotes matrix-matrix multiplication in numpy and scipy.
To check your results, construct the Hamiltonian for $N = 2$ spins with $J = 1$ and $\Gamma = 0.1$. You should obtain:

$$\begin{pmatrix} 2 & -0.1 & -0.1 & 0 \\ -0.1 & -2 & 0 & -0.1 \\ -0.1 & 0 & -2 & -0.1 \\ 0 & -0.1 & -0.1 & 2 \end{pmatrix}.$$

c) First use the eigensolver `scipy.sparse.linalg.eigsh` (an advanced version of Lanczos algorithm) to diagonalize the Hamiltonian and find the first few energy eigenvalues. Then write a function that implements the power method from the lecture (4.2.1) to find the ground state and its energy. Compare your results. Plot the energy obtained from the power method for each iteration step to observe the convergence behaviour of the method.

d) For system sizes $N \in \{6, 8, 10, 12\}$ compute the ground state for ≈ 20 different values of $\Gamma \in [0, 2]$ $(J = 1)$. Compute the largest-distance spin-spin correlator $C = \langle\psi_0| \hat{\sigma}_0^z \hat{\sigma}_{N/2}^z |\psi_0\rangle$ for each of the obtained ground states and plot it against the different values for Γ.

e) For the same set of system sizes and magnetic fields Γ and again using `scipy.sparse.linalg.eigsh`, find the first excited energy level and plot it along with the ground state energy against the magnetic field Γ. Extrapolate the energy gap at $\Gamma/J = 1$ to infinite system size, i.e., $1/N \to 0$. What can you observe? Explain your results from this exercise and exercise d) in light of the mentioned phase transition at $\Gamma/J = 1$.

Problem 3.2 Time evolution of the transverse field Ising chain

The goal of this exercise is to perform time evolution on a quantum state using sparse matrix-vector multiplications. We study the transverse field Ising chain with open boundary conditions, which is defined by the following Hamiltonian

$$\hat{H}_{\text{Ising}} = \hat{H}_{\text{ZZ}} + \hat{H}_{\text{X}} = J \sum_{i=1}^{N-1} \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z - \Gamma \sum_{i=1}^{N} \hat{\sigma}_i^x. \tag{2}$$

As a first step we split the time evolution operator into 2 non-commuting terms, the diagonal part given by H_{ZZ} and the non-diagonal part given by H_{X}. The error involved in doing this can be kept small by choosing a small time step Δ_t (see section 4.4):

$$\hat{U} = \exp\Big(-i\Delta_t \hat{H}\Big) \approx \exp\Big(-i\frac{\Delta_t}{2}\hat{H}_{\text{ZZ}}\Big) \exp\Big(-i\Delta_t \hat{H}_{\text{X}}\Big) \exp\Big(-i\frac{\Delta_t}{2}\hat{H}_{\text{ZZ}}\Big) + O(\Delta_t^3). \tag{3}$$

The diagonal part $\exp\Big(-i\frac{\Delta_t}{2}\hat{H}_{\text{ZZ}}\Big)$ multiplies each basis state with a phase factor, and can be simplified into a product of two-site operators

$$\exp\Big(-i\frac{\Delta_t}{2}\hat{H}_{\text{ZZ}}\Big) = \prod_{i=1}^{N-1} \exp\Big(-i\frac{\Delta_t}{2} J \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z\Big). \tag{4}$$

The non-diagonal part can also be simplified into a product of single-site operators

$$\exp\Big(-i\Delta_t \hat{H}_{\text{X}}\Big) = \prod_{i=1}^{N} \exp(i\Delta_t \Gamma \hat{\sigma}_i^x). \tag{5}$$

a) Write each time evolution operator for the transverse field part $\exp(i\Delta_t\Gamma\hat{\sigma}_i^x)$ as a sparse matrix.

b) Write each time evolution operators for the diagonal part $\exp\big(-i\Delta_t J\hat{\sigma}_i^z\hat{\sigma}_{i+1}^z\big)$ and $\exp\big(-i\frac{\Delta_t}{2}J\hat{\sigma}_i^z\hat{\sigma}_{i+1}^z\big)$ as sparse matrices.

c) Implement the code that performs the time evolution for total time t on the transverse field Ising chain with open boundary conditions in space.

d) For system size $N = 10$, use your code to compute the time evolution of a state. Start with an all-down configuration using only a transverse field Γ and without Ising coupling (thus take $J = 0$). Measure the magnetisation of each site at ≈ 100 different time points in $[0, 10]$.
Hint 1: The initial state can be constructed from a tensor product of single-site states $|\Psi\rangle = |{\downarrow}\rangle \otimes |{\downarrow}\rangle \otimes \ldots \otimes |{\downarrow}\rangle$. In the z-basis, the single-site state $|{\downarrow}\rangle$ is represented by a two-component vector $(0, 1)$. Note that we don't need to store $|\Psi\rangle$ as a sparse vector, and we can use `numpy.kron` to construct it, because in general it's dense during the time evolution.
Hint 2: You can measure the magnetisation of a site i by computing $\langle\Psi|\hat{\sigma}_i^z|\Psi\rangle$. In numpy, you need to take the real part of the result, and the imaginary part should be zero.

e) Make a 2D color plot to show the magnetisation as a function of site and time. You should see oscillations. Can you relate the period of the oscillation to the magnitude of Γ?
Hint: You can use `matplotlib.pyplot.pcolormesh` to make the 2D color plot.

f) Now set the Ising coupling $J = 1$ and use a transverse field of $\Gamma = 0.4$. The starting configuration is again all-down except of a single spin flipped in the middle. What do you observe? How does the behaviour change with Γ?