Exercise 5.1: Lanczos algorithm - ground state and dynamical properties

The file `lanczos.py` contains an implementation of the lanczos algorithm and functions to generate a sparse matrix representing the (by now familiar) Hamiltonian of the transverse field Ising model, where we choose ferromagnetic $J = 1$ for now. For simplicity, we use the full basis and do not exploit symmetries to block-diagonalize H.

a) Generate the Hamiltonian for $L = 14$ and $g = 1.5$ using the functions provided in `lanczos.py`. Moreover, generate a random state (in the full basis), which we can use as a starting vector for the Lanczos iteration.

b) Call the function `lanczos()` with parameters `N=200, stabilize=False`. The function returns the tridiagonal matrix T and orthonormal basis of the Krylov space generated during the Lanczos iteration. Determine the 10 smallest eigenvalues of T using `np.linalg.eigvalsh`. Do you find a ground state degeneracy? Do you expect a degeneracy for these parameters?

   Call `lanczos()` again with `stabilize=True`. What does this option do? Do you get the expected degeneracy now? Can you explain this? Confirm the results by comparing with the (quasi-exact) energies returned by `scipy.sparse.linalg.eigsh`.

c) Find the ground state $|u_0\rangle$ of T and use it to find the ground state $|\psi_0\rangle$ of H in the full basis. Check that the state you obtain is normalized, has the correct energy $E_0 = \langle\psi_0|H|\psi_0\rangle$. Calculate the variance $\langle\psi_0|H^2|\psi_0\rangle - \langle\psi_0|H|\psi_0\rangle^2$ to see if $|\psi_0\rangle$ is an eigenstate of H.

   Hint: Let V denote the matrix containing the vectors returned by `lanczos()` as columns, then $V^\dagger H V = T$ Hence, if $|u\rangle$ is an eigenvector of T, $|\psi_0\rangle := V\,|u_0\rangle$ is an (approximate) eigenvector of H.

To determine dynamical properties, we want to calculate

$$I(\hat{O}, z) = -\frac{1}{\pi}\mathrm{Im}\,\underbrace{\langle\psi_0|\hat{O}^\dagger \frac{1}{z-H}\hat{O}|\psi_0\rangle}_{\equiv x_0} \qquad \text{with } z = \omega + E_0 + \mathrm{i}\epsilon \tag{1}$$

Recall from class that this can be done by starting a Lanczos iteration from $|\phi_0\rangle = \hat{O}\,|\psi_0\rangle$, which gives us the tridiagonal matrix T such that

$$z - T = \begin{pmatrix} z-\alpha_0 & -\beta_1 & & \\ -\beta_1 & z-\alpha_1 & -\beta_2 & \\ & -\beta_2 & z-\alpha_1 & \ddots \\ & & \ddots & \ddots \end{pmatrix} \tag{2}$$

Using Cramer's Rule, we can evaluate $x_0 = \frac{\det A}{\det(z-T)}$, where A is $z - T$ with the first column replaced by the unit vector $(1, 0, \dots)^T$. This leads to the continued fraction

$$\begin{aligned} x_0 &= \frac{1}{z - \alpha_0 - \beta_1^2 \frac{\det D_2}{\det D_1}}, \\ \frac{\det D_2}{\det D_1} &= \frac{1}{z - \alpha_1 - \beta_2^2 \frac{\det D_3}{\det D_2}}, \\ \frac{\det D_3}{\det D_2} &= \frac{1}{z - \alpha_2 - \beta_3^2 \frac{\det D_4}{\det D_3}}, \dots \end{aligned} \tag{3}$$

where D_n is the matrix $z - T$ with the first n rows and columns ommited. For a $N \times N$ matrix T, the last ratio evaluates as

$$\frac{\det D_{N-1}}{\det D_{N-2}} = \frac{1}{z - \alpha_{N-2} - \frac{\beta_{N-1}^2}{z - \alpha_{N-1}}} \tag{4}$$

d) Construct the state $|\phi_0\rangle = S_0^+ |\psi_0\rangle$ and get the tridiagonal matrix T of a Lanczos iteration starting from this initial state. Here, $S_j^+ = \frac{1}{2}(\sigma_j^x + \mathrm{i}\sigma_j^y)$ labels the spin-raising operator on site j.

   Hint: To construct S_0^+, take a look at `lanczos.py` again.

e) Write a function which (given z and T) evaluates the continued fractions of eq. (3) to calculate I.

   Hint: You can extract α and β from T using `np.diag`. Take care that we labeled α_n starting from $n = 0$, while for β_n we start from $n = 1$.

f) Plot $I(S_0^+, \omega)$ versus ω. Choose $z = \omega + E_0 + \mathrm{i}\epsilon$ for $\omega \in [-1, 10]$ and ϵ in the order of $0.001 \lesssim \epsilon \lesssim 0.1$. What is the influence of ϵ?

   Hint: There's a good chance that the function calculating I from e) works with z being a numpy array with different ω values; this will lead to a faster evaluation than calling it for each ω separately.

g) To get more physical insight, we can choose a momentum-dependent operator for $\hat{O}$. Calculate $I(S_k^+, \omega)$ for the k values compatible with the chosen $L = 14$, where S_k^+ is defined as

$$S_k^+ = \frac{1}{\sqrt{L}} \sum_{j=0}^{L-1} e^{\mathrm{i}jk} S_j^+. \tag{5}$$

   Create a 2D colorplot of $I(S_k^+, \omega)$ with k on the x-axis and ω on the y-axis.

   Hint: To create the colorplot, you can use the function provided in `lanczos.py`.

h) Regenerate similar colorplots for other values of g in both phases and at the critical point, both for ferromagnetic $J = 1$ and antiferromagnetic $J = -1$.

Bonus Use for $\hat{O}$ the fermionic creation ($c_k^\dagger$) and annihilation (c_k) from the Jordan-Wigner transformation, defined by

$$c_k^\dagger = \frac{1}{\sqrt{L}} \sum_j e^{-\mathrm{i}jk} c_j^\dagger, \qquad c_j^\dagger = \sigma^z \otimes \dots \otimes \sigma^z \otimes S_j^- \otimes \mathbb{1} \otimes \dots \otimes \mathbb{1}. \tag{6}$$