Spin Computations

1. Show that for appropriate dimensional rectangular matrices we have

   (a) $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

   (b) $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{pmatrix}$

   What are the conditions on the dimensions?

2. Show that the most general spin-1/2 state can be written as $\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle$.

3. The Hamiltonian of a spin-1/2 system is $H = \vec{B}.\vec{S}$, where $\vec{B} = B_x \hat{i} + B_y \hat{j} + B_z \hat{k}$ is an external magnetic field and S are related to Pauli matrices.

   (a) Calculate the three components of magnetization in the ground state for random external magnetic field.

   (b) Plot the three components of magnetization in the ground state as a function of B_z, keeping B_x and B_y fixed to some random values.

   (c) Let the initial state (at $t = 0$) be a random complex vector $|\psi(0)\rangle$. Find the state at a later time t.

   (d) Write a program to calculate the time evolution for a random initial state. Plot the three components of magnetization with time.

   Follow up reading: section 4.3 from Townsend.

4. Repeat the above exercise for a spin-1 system.

5. Write two python functions to solve such problems in general for system of any size and also work for both spin-1/2 and spin-1. First one will take the Hamiltonian and a list of observables as input and return their expectation values in the ground state. The second one takes the Hamiltonian, the initial state, a list of time points for observation and a list of observables as input and return their expectation values as a function of time.

6. An interacting spin-1/2 and spin-1 particle in external magnetic field is represented by the following Hamiltonian

$$H = \vec{S}_A.\vec{S}_B + hS^z_A + S^z_B.$$

   (a) Let particle A be in a random spin-1/2 state and particle B be in a random spin-1 state. Calculate the expectation value of the z component of magnetization in this state for each particle using two methods - first using the matrices and vectors of individual particles and second using the matrices and vectors of the full system. [Required matrices: $S^z_{A1} = 2 \times 2$, $S^z_{A2} = 6 \times 6$, $S^z_{B1} = 3 \times 3$, $S^z_{B2} = 6 \times 6$, $\psi_1 = 2 \times 1$, $\psi_2 = 3 \times 1$, $\psi = \psi_1 \otimes \psi_2 = 6 \times 1$]

   (b) Write a program to find the eigenvalues and eigenvectors of the system. Plot the expectation value of the z component of magnetization in the ground state as a function of h for each particle.

Follow up reading: section 5.1 from Townsend.

7. Consider a system of three spin-1/2 particles (A, B, C) given by the Hamiltonian

$$H = \vec{S}_A.\vec{S}_B + \vec{S}_B.\vec{S}_C.$$

Calculate the expectation values of the nine spin operators in the ground state.

8. What are the matrices $S_0^x, S_1^x, \cdots, S_{L-1}^x$ for L spin-1/2 systems? Write a python program to generate them for an arbitrary L.

9. The 1d spin-1/2 XXZ system in random external magnetic field is given by

$$H = \sum_i \left(S_i^x S_{i+1}^x + S_i^y S_{i+1}^y + \Delta S_i^z S_{i+1}^z \right) + \sum_i h_i S_i^z .$$

   Write a program to construct this Hamiltonian for a given value of system size L for both periodic and open boundary condition.

10. Write a program to perform time evolution using above Hamiltonian and the initial state $|\uparrow\downarrow\uparrow\downarrow\cdots\rangle$ and calculate z-magnetization of the first site as a function of time.

11. The above Hamiltonian commutes with the total spin operator $\sum_i S_i^z$. Show this using spin-1/2 algebra.

12. What state do we get if we apply the XXZ Hamiltonian to $|\uparrow\downarrow\uparrow\downarrow\cdots\rangle$?

13. How many entries are nonzero in the matrix S_i^x? How about the XXZ Hamiltonian?

14. Find the ground state of the XXZ Hamiltonian using the sparse matrix algorithm `eigsh`.

Projection Operator

If we represent spin up and down in z-basis by $|0\rangle$ and $|1\rangle$ respectively, then the tensor product of many spin-1/2 states will give us the computational basis in dictionary order, e.g. $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $\cdots, |111\rangle$ for three systems. The total z-magnetization in these states are not contiguous, in the above example they are $0, 1, 1, 2, 1, 2, 2, 3$.

1. These values can be made contiguous by permuting the basis states. Find the matrix $\mathcal{P}_1$ that performs such a permutation. In the above example multiplication by this matrix should interchange the middle two states and leave others unaffected, i.e.

   $$\mathcal{P}_1|011\rangle = |100\rangle, \quad \mathcal{P}_1|100\rangle = |011\rangle, \quad \mathcal{P}_1|i\rangle = |i\rangle \text{ for others}$$

   Write a python function that generates the matrix $\mathcal{P}_1$ for any L.

2. Visualize the matrix $\mathcal{P}_1 H \mathcal{P}_1^\dagger$.

3. We can project to a subspace with given total z-magnetization using a suitable matrix, say $\mathcal{P}_2$. In our example if we want to project to total z-magnetization 1 subspace, $\mathcal{P}_2$ should be such that

   $$\mathcal{P}_2|001\rangle = |001\rangle, \quad \mathcal{P}_2|010\rangle = |010\rangle, \quad \mathcal{P}_2|100\rangle = |100\rangle,$$
   $$\mathcal{P}_2|i\rangle = 0 \text{ for others}$$

   Write a python function that generates the matrix $\mathcal{P}_2$ for any L.

4. Visualize the matrix $\mathcal{P}_1 \mathcal{P}_2 H \mathcal{P}_2^\dagger \mathcal{P}_1^\dagger$.

5. We can select a square contiguous block of a square matrix A using an appropriate rectangular matrix $\mathcal{X}$ via $\mathcal{X} A \mathcal{X}^\dagger$. What is $\mathcal{X}$ if we want to select indices $1, 2, 3$ (out of 0 to 7) of a 8×8 matrix?

6. Visualize the matrix $\mathcal{P}H\mathcal{P}^\dagger$, where $\mathcal{P} = \mathcal{X}\mathcal{P}_1\mathcal{P}_2$.
7. Carefully analyze the matrix $\mathcal{P}$ and come up with a direct algorithm to generate it.

`numpy` functions required:
`array, eye, zeros, kron, dot, diag, reshape,`
`linalg.eigh, linalg.svd`
`matplotlib` function to visualize the matrices: `pcolormesh`