February 5-9, 2025: Quantum Computing, Quantum Machine Learning and Quantum Information Theories

Morten Hjorth-Jensen[1,2]

[1]Department of Physics, University of Oslo
[2]Department of Physics and Astronomy and Facility for Rare Isotope Beams, Michigan State University, USA

February 7

Plans for the week of February 5-9

1. Reminder from last week
2. Entanglement and Schmidt decomposition
3. Entropy as a measurement of entanglement
4. Simple Hamiltonian systems and how to use the density matrix to estimate degrees of entanglement
5. Introduction to gates and calculations

Density matrices and traces

In order to study entanglement and why it is so important for quantum computing, we need to introduce some basic measures and useful quantities. For these endeavors, we will use our two-qubit system from the second lecture in order to introduce, through examples, density matrices and entropy. These two quantities, together with technicalities like the Schmidt decomposition define important quantities in analyzing quantum computing examples.

Reminder on density matrices

We have the spectral decomposition of a given operator $\boldsymbol{A}$ given by

$$\boldsymbol{A} = \sum_i \lambda_i |i\rangle\langle i|,$$

with the ONB $|i\rangle$ being eigenvectors of $\boldsymbol{A}$ and λ_i being the eigenvectors. Similarly, a operator which is a function of $\boldsymbol{A}$ is given by

$$f(\boldsymbol{A}) = \sum_i f(\boldsymbol{A})|i\rangle\langle i|.$$

The trace of a product of matrices is cyclic, that is

$$\mathrm{tr}[\boldsymbol{ABC}]) = \mathrm{tr}[\boldsymbol{BCA}]) = \mathrm{tr}[\boldsymbol{CBA}]),$$

and we have also

$$\mathrm{tr}[\boldsymbol{A}|\psi\rangle\langle\psi|]) = \langle\psi|\boldsymbol{A}|\psi\rangle.$$

Using the spectral decomposition we defined also the density matrix as

$$\rho = \sum_i p_i|i\rangle\langle i|,$$

where the probability p_i are the eigenvalues of the density linked with the ONB $|i\rangle$.

The trace of the density matrix $\mathrm{tr}\rho = 1$ and is invariant under unitary transformations $|\psi_i'\rangle = \boldsymbol{U}|\psi_i\rangle$. The unitary transformation of the density matrix gives, with $\boldsymbol{U}^\dagger\boldsymbol{U} = \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$,

$$\boldsymbol{U}\rho\boldsymbol{U}^\dagger = \sum_i pi_i\boldsymbol{U}|\psi_i\rangle\langle\psi_i|\boldsymbol{U}^\dagger,$$

and with the unitary transformation it is easy to show that thet trace of the transformaed density matrix is equal to one,

$$\mathrm{tr}\left[\boldsymbol{U}\rho\boldsymbol{U}^\dagger\right] = \mathrm{tr}\left[\boldsymbol{U}\boldsymbol{U}^\dagger\rho\right] = 1.$$

Two-qubit system and definition of density matrices

This system can be thought of as composed of two subsystems A and B. Each subsystem has computational basis states

$$|0\rangle_{\mathrm{A,B}} = \begin{bmatrix}1 & 0\end{bmatrix}^T \qquad |1\rangle_{\mathrm{A,B}} = \begin{bmatrix}0 & 1\end{bmatrix}^T.$$

The subsystems could represent single particles or composite many-particle systems of a given symmetry. This leads to the many-body computational basis states

$$|00\rangle = |0\rangle_{\mathrm{A}} \otimes |0\rangle_{\mathrm{B}} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T,$$

and

$$|01\rangle = |0\rangle_{\mathrm{A}} \otimes |1\rangle_{\mathrm{B}} = \begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}^T,$$

and

$$|10\rangle = |1\rangle_{\mathrm{A}} \otimes |0\rangle_{\mathrm{B}} = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}^T,$$

and finally

$$|11\rangle = |1\rangle_{\mathrm{A}} \otimes |1\rangle_{\mathrm{B}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T.$$

These computational basis states define also the eigenstates of the non-interacting Hamiltonian

$$\begin{aligned} H_0|00\rangle &= \epsilon_{00}|00\rangle, \\ H_0|10\rangle &= \epsilon_{10}|10\rangle, \\ H_0|01\rangle &= \epsilon_{01}|01\rangle, \end{aligned}$$

and

$$H_0|11\rangle = \epsilon_{11}|11\rangle.$$

The interacting part of the Hamiltonian H_{I} is given by the tensor product of two σ_x and σ_z matrices, respectively, that is

$$H_{\mathrm{I}} = H_x \sigma_x \otimes \sigma_x + H_z \sigma_z \otimes \sigma_z,$$

where H_x and H_z are interaction strength parameters. Our final Hamiltonian matrix is given by

$$\boldsymbol{H} = \begin{bmatrix} \epsilon_{00} + H_z & 0 & 0 & H_x \\ 0 & \epsilon_{10} - H_z & H_x & 0 \\ 0 & H_x & \epsilon_{01} - H_z & 0 \\ H_x & 0 & 0 & \epsilon_{11} + H_z \end{bmatrix}.$$

The four eigenstates of the above Hamiltonian matrix can in turn be used to define density matrices. As an example, the density matrix of the first eigenstate (lowest energy E_0) Ψ_0 is

$$\rho_0 = \left(\alpha_{00}|00\rangle\langle 00| + \alpha_{10}|10\rangle\langle 10| + \alpha_{01}|01\rangle\langle 01| + \alpha_{11}|11\rangle\langle 11|\right),$$

where the coefficients α_{ij} are the eigenvector coefficients resulting from the solution of the above eigenvalue problem.

We can then in turn define the density matrix for the subsets A or B as

$$\rho_A = \mathrm{Tr}_B(\rho_0) = \langle 0|\rho_0|0\rangle_B + \langle 1|\rho_0|1\rangle_B,$$

or

$$\rho_B = \mathrm{Tr}_A(\rho_0) = \langle 0|\rho_0|0\rangle_A + \langle 1|\rho_0|1\rangle_A.$$

The density matrices for these subsets can be used to compute the so-called von Neumann entropy, which is one of the possible measures of entanglement. A pure state has entropy equal zero while entangled state have an entropy larger than zero. The von-Neumann entropy is defined as

$$S(A,B) = -\mathrm{Tr}\left(\rho_{A,B} \log_2(\rho_{A,B})\right).$$

The example here shows the above von Neumann entropy based on the density matrix for the lowest many-body state. We see clearly a jump in the entropy around the point where we have a level crossing. At interaction strenght $\lambda = 0$ we have many-body states purely defined by their computational basis states. As we switch on the interaction strength, we obtain an increased degree of mixing and the entropy increases till we reach the level crossing point where we see an additional and sudden increase in entropy. Similar behaviors are observed for the other states. The most important result from this example is that entanglement is driven by the Hamiltonian itself and the strength of the interaction matrix elements and the non-interacting energies.

```python
%matplotlib inline
from  matplotlib import pyplot as plt
import numpy as np
from scipy.linalg import logm, expm
def log2M(a): # base 2 matrix logarithm
    return logm(a)/np.log(2.0)

dim = 4
Hamiltonian = np.zeros((dim,dim))
#number of lambda values
n = 40
lmbd = np.linspace(0.0,1.0,n)
Hx = 2.0
Hz = 3.0
# Non-diagonal part as sigma_x tensor product with sigma_x
sx = np.matrix([[0,1],[1,0]])
sx2 = Hx*np.kron(sx, sx)
# Diagonal part as sigma_z tensor product with sigma_z
sz = np.matrix([[1,0],[0,-1]])
sz2 = Hz*np.kron(sz, sz)
noninteracting = [0.0, 2.5, 6.5, 7.0]
D = np.diag(noninteracting)
Eigenvalue = np.zeros((dim,n))
Entropy = np.zeros(n)

for i in range(n):
    Hamiltonian = lmbd[i]*(sx2+sz2)+D
    # diagonalize and obtain eigenvalues, not necessarily sorted
    EigValues, EigVectors = np.linalg.eig(Hamiltonian)
    # sort eigenvectors and eigenvalues
    permute = EigValues.argsort()
    EigValues = EigValues[permute]
    EigVectors = EigVectors[:,permute]
    # Compute density matrix for selected system state, here ground state
    DensityMatrix = np.zeros((dim,dim))
    DensityMatrix = np.outer(EigVectors[:,0],EigVectors[:,0])
    # Project down on substates and find density matrix for subsystem
    d = np.matrix([[1,0],[0,1]])
    v1 = [1.0,0.0]
    proj1 = np.kron(v1,d)
    x1 = proj1 @ DensityMatrix @ proj1.T
    v2 = [0.0,1.0]
    proj2 = np.kron(v2,d)
    x2 = proj2 @ DensityMatrix @ proj2.T
    # Total density matrix for subsystem
    total = x1+x2
    # von Neumann Entropy for subsystem
    Entropy[i] = -np.matrix.trace(total @ log2M(total))
    # Plotting eigenvalues and entropy as functions of interaction strengths
    Eigenvalue[0,i] = EigValues[0]
```

```
        Eigenvalue[1,i] = EigValues[1]
        Eigenvalue[2,i] = EigValues[2]
        Eigenvalue[3,i] = EigValues[3]
    plt.plot(lmbd, Eigenvalue[0,:] ,'b-',lmbd, Eigenvalue[1,:],'g-',)
    plt.plot(lmbd, Eigenvalue[2,:] ,'r-',lmbd, Eigenvalue[3,:],'y-',)
    plt.xlabel('$\lambda$')
    plt.ylabel('Eigenvalues')
    plt.show()
    plt.plot(lmbd, Entropy)
    plt.xlabel('$\lambda$')
    plt.ylabel('Entropy')
    plt.show
```

Shannon information entropy

We start our discussions with the classical information entropy, or just Shannon entropy, before we move over to a quantum mechanical way to define the entropy based on the density matrices discussed earlier.

We define a set of random variables $X = \{x_0, x_1, \ldots, x_{n-1}\}$ with probability for an outcome $x \in X$ given by $p_X(x)$, the information entropy is defined as

$$S = -\sum_{x \in X} p_X(x) \log_2 p_X(x).$$

Von Neumann entropy

$$S = -\mathrm{Tr}[\rho \log_2 \rho].$$

Quantum gates, circuits and simple algorithms

Quantum gates are physical actions that are applied to the physical system representing the qubits. Mathematically, they are complex-valued, unitary matrices which act on the complex-values normalized vectors that represent qubits. As the quantum analog of classical logic gates (such as AND and OR), there is a corresponding quantum gate for every classical gate; however, there are quantum gates that have no classical counter-part. They act on a set of qubits and, changing their state. That is, if U is a quantum gate and $|q\rangle$ is a qubit, then acting the gate U on the qubit $|q\rangle$ transforms the qubit as follows:

$$|q\rangle \overset{U}{\to} U|q\rangle. \tag{1}$$

This action would be represented as the following quantum circuit

$$|q\rangle U \qquad\qquad\qquad\qquad U|q\rangle \tag{2}$$

Quantum circuits are diagrammatic representations of quantum algorithms. The horizontal dimension corresponds to time; moving left to right corresponds to forward motion in time. They consist of a set of qubits $|q_n\rangle$ which are stacked vertically on the left-hand side of the diagram. Lines, called quantum wires, extend horizontally to the right from each qubit, representing its state moving forward in time. Additionally, they contain a set of quantum gates that are applied to the quantum wires. Gates are applied chronologically, left to right. With this, we can see that the quantum circuit above (2) implies that the quantum gate U is being applied to the qubit in state $|q\rangle$.

To explain what quantum circuits represent mathematically, consider the following circuit

$$@C = 1em@R = 1em|q_0\rangle A \qquad\qquad B \tag{3}$$

$$|q_1\rangle C \qquad\qquad D \tag{4}$$

This circuit implies the following mathematical statement

$$\begin{aligned} |q_0 q_1\rangle &\rightarrow (B \otimes D)(A \otimes C)|q_0 q_1\rangle \qquad (5)\\ &\rightarrow (BA) \otimes (DC)|q_0 q_1\rangle \qquad (6)\\ &\rightarrow BA|q_0\rangle DC|q_1\rangle. \qquad (7) \end{aligned}$$

Note that the mathematical form is in reverse order from circuit form ($AB \leftrightarrow BA$). This is because the operator closest to the state (furthest to the right) acts first. Additionally, we are able to write the actions of the top two gates and the bottom two as acting separately on each qubit as every gate here is a single-qubit gate (acting on only one qubit). The same would not be true for certain two-qubit gates which would entangle the states of the two qubits, not allowing their state to be written in a separable form. Finally, we define the depth of a quantum circuit as the number of columns of gates. The circuit above thus has a depth of 2 because it contains two columns of gates, namely $A \otimes C$ and $B \otimes D$.

Single-Qubit Gates

A single-qubit gate is a physical action that is applied to one qubit. It can be represented by a matrix U from the group SU(2). Any single-qubit gate can be parameterized by three angles: θ, ϕ, and λ as follows

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{pmatrix}. \tag{8}$$

Widely used gates

There are several widely used quantum gates, include the following: The Pauli gates correspond to the Pauli matrices

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{9}$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{10}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{11}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{12}$$

which satisfy the relation

$$\sigma\tau = i\epsilon_{\sigma\tau\upsilon}\upsilon, \tag{13}$$

for $\sigma, \tau, \upsilon \in \{X, Y, Z\}$. These gates form a basis for the algebra $\mathfrak{su}(2)$. Exponentiating them will thus give us a basis for SU(2), the group within which all single-qubit gates live. These exponentiated Pauli gates are called rotation gates $R_\sigma(\theta)$ because they rotate the quantum state around the axis $\sigma = X, Y, Z$ of the Bloch sphere by an angle θ. They are defined as

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \tag{14}$$

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \tag{15}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \tag{16}$$

Because they form a basis for SU(2), any single-qubit gate can be decomposed into three rotation gates. Indeed

$$R_z(\phi)R_y(\theta)R_z(\lambda) = \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix} \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \begin{pmatrix} e^{-i\lambda/2} & 0 \\ 0 & e^{i\lambda/2} \end{pmatrix} \tag{17}$$

$$= e^{-i(\phi+\lambda)/2} \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{pmatrix}, \tag{18}$$

which is, up to a global phase, equal to the expression for an arbitrary single-qubit gate (8).

Two-Qubit Gates

A two-qubit gate is a physical action that is applied to two qubits. It can be represented by a matrix U from the group SU(4). One important type of two-qubit gates are controlled gates, which work as follows: Suppose U is a single-qubit gate. A controlled-U gate (CU) acts on two qubits: a control qubit $|x\rangle$ and a target qubit $|y\rangle$. The controlled-U gate applies the identity I or the single-qubit gate U to the target qubit if the control gate is in the zero state 0 or the one state 1, respectively. The control qubit is not acted upon. This can be represented as follows:

$$CUxy = \begin{cases} xy & \text{if } |x\rangle = 0 \\ |x\rangle U|y\rangle & \text{if } |x\rangle = 1 \end{cases}. \tag{19}$$

The action of a controlled-U gate CU can be represented in a quantum circuit as follows

$$@C = 1em@R = 3em|x\rangle 1 \qquad\qquad\qquad\qquad |x\rangle \tag{20}$$

$$|y\rangle U \qquad\qquad\qquad \begin{cases} |y\rangle, & |x\rangle = 0 \\ U|y\rangle, & a = |x\rangle \end{cases} \tag{21}$$

It can be written in matrix form by writing it as a superposition of the two possible cases, each written as a simple tensor product

$$CU = 00 \otimes I + 11 \otimes U \tag{22}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix}. \tag{23}$$

One of the most fundamental controlled gates is the CNOT gate. It is defined as the controlled-X gate CX and thus flips the state of the target qubit if the control qubit is in the zero state 0. It can be written in matrix form as follows:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{24}$$

A widely used two-qubit gate that goes beyond the simple controlled function is the SWAP gate. It swaps the states of the two qubits it acts upon

$$\text{SWAP}xy = yx, \tag{25}$$

as depicted in the quantum circuit below

$$|x\rangle \qquad\qquad\qquad |y\rangle \tag{26}$$

$$|y\rangle \qquad\qquad\qquad |x\rangle, \tag{27}$$

and has the following matrix form

$$\mathrm{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{28}$$

It can be decomposed into a series of three CNOTs, each of which has its directionality flipped from the previous

$$|x\rangle 1 \qquad\qquad 1 \qquad\qquad |y\rangle \tag{29}$$

$$|y\rangle \qquad\qquad -1 \qquad\qquad |x\rangle \tag{30}$$

As for arbitrary two-qubit gates $U \in \mathrm{SU}(4)$, they can be optimally decomposed (up to a global phase) into the following sequence citeref:kak involving three parameters, fifteen elementary one-qubit gates and three CNOT gates

$$U_1 \qquad 1R_y(\theta_1) \qquad R_y(\theta_2) \qquad 1U_3 \tag{31}$$

$$U_2 \qquad R_z(\theta_3) \qquad -1 \qquad U_4 \tag{32}$$

where U_1, U_2, U_3, U_4 are single-qubit gates, each of which can be decomposed into three elementary one-qubit gates (rotation gates). Additionally, $\theta_1, \theta_2, \theta_3$ are parameters to be determined by the arbitrary two-qubit gate to be decomposed. Two-qubit gates that are restricted to $U \in \mathrm{SO}(4)$ can be decomposed into a shorter depth circuit consisting of just twelve elementary single-qubit gates and two CNOT gates

$$R_z(\pi/2) \qquad R_y(\pi/2)1 \qquad U_1 1 \qquad R_y^*(\pi/2)R_z^*(\pi/2) \tag{33}$$

$$R_z(\pi/2) \qquad\qquad U_2 \qquad\qquad R_z^*(\pi/2) \tag{34}$$

Lecture next week we discuss simple algorithms and quantum circuits.

1. Introduction to Qiskit
2. Defining one-, two- and three-qubit gates
3. Setting up quantum circuits and simple algorithms