Quantum Computing Lectures for Nano and Quantum Workshop

Morten Hjorth-Jensen[1,2]

[1]Department of Physics and Center for Computing in Science Education, University of Oslo, Norway
[2]Department of Physics and Astronomy and Facility for Rare Isotope Beams, Michigan State University, East Lansing, Michigan, USA

Cali, Colombia, December 4-8, 2023

Spectral Decomposition, Measurements and Density matrices

1. Density matrices and measurements
2. Entropy and entaglement
3. Additional material on gates and using Qiskit

Introduction

In order to study entanglement and why it is so important for quantum computing, we need to introduce some basic measures and useful quantities. These quantities are the spectral decomposition of hermitian operators, how these are then used to define measurements and how we can define so-called density operators (matrices). These are all quantities which will become very useful when we discuss entanglement and in particular how to quantify it. In order to define these quantities we need first to remind ourselves about some basic linear algebra properties of hermitian operators and matrices.

Basic properties of hermitian operators

The operators we typically encounter in quantum mechanical studies

1. Hermitian (self-adjoint) meaning that for example the elements of a Hermitian matrix $\boldsymbol{U}$ obey $u_{ij} = u^*_{ji}$.
2. Unitary $\boldsymbol{U}\boldsymbol{U}^\dagger = \boldsymbol{U}^\dagger\boldsymbol{U} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the unit matrix

3. The oparator $\boldsymbol{U}$ and its self-adjoint commute (often labeled as normal operators), that is $[\boldsymbol{U}, \boldsymbol{U}^\dagger] = 0$. An operator is **normal** if and only if it is diagonalizable. A Hermitian operator is normal.

Unitary operators in a Hilbert space preserve the norm and orthogonality. If $\boldsymbol{U}$ is a unitary operator acting on a state $|\psi_j\rangle$, the action of

$$|\phi_i\rangle = \boldsymbol{U}|\psi_j\rangle,$$

preserves both the norm and orthogonality, that is $\langle\phi_i|\phi_j\rangle = \langle\psi_i|\psi_j\rangle = \delta_{ij}$, as discussed earlier.

As example, consider the Pauli matrix σ_x. We have already seen that this matrix is a unitary matrix. Consider then an orthogonal and normalized basis $|0\rangle^\dagger = \begin{bmatrix}1 & 0\end{bmatrix}$ and $|1\rangle^\dagger = \begin{bmatrix}0 & 1\end{bmatrix}$ and a state which is a linear superposition of these two basis states

$$|\psi_a\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle.$$

A new state $|\psi_b\rangle$ is given by

$$|\psi_b\rangle = \sigma_x|\psi_a\rangle = \alpha_0|1\rangle + \alpha_1|0\rangle.$$

Spectral Decomposition. An important techanicality which we will use in the discussion of density matrices, entanglement, quantum entropies and other properties is the so-called spectral decomposition of an operator.

Let $|\psi\rangle$ be a vector in a Hilbert space of dimension n and a hermitian operator $\boldsymbol{A}$ defined in this space. Assume $|\psi\rangle$ is an eigenvector of $\boldsymbol{A}$ with eigenvalue λ, that is

$$\boldsymbol{A}|\psi\rangle = \lambda|\psi\rangle = \lambda\boldsymbol{I}|\psi\rangle,$$

where we used $\boldsymbol{I}|\psi\rangle = 1|\psi\rangle$. Subtracting the right hand side from the left hand side gives

$$[\boldsymbol{A} - \lambda\boldsymbol{I}]\,|\psi\rangle = 0,$$

which has a nontrivial solution only if the determinant $\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0$.

We define now an orthonormal basis $|i\rangle = \{|0\rangle, |1\rangle, \ldots, |n-1\rangle$ in the same Hilbert space. We will assume that this basis is an eigenbasis of $\boldsymbol{A}$ with eigenvalues λ_i

We expand a new vector using this eigenbasis of $\boldsymbol{A}$

$$|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i|i\rangle,$$

with the normalization condition $\sum_{i=0}^{n-1} |\alpha_i|^2$. Acting with $\boldsymbol{A}$ on this new state results in

$$\boldsymbol{A}|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i\boldsymbol{A}|i\rangle = \sum_{i=0}^{n-1} \alpha_i\lambda_i|i\rangle.$$

If we then use that the outer product of any state with itself defines a projection operator we have the projection operators

$$\boldsymbol{P}_\psi = |\psi\rangle\langle\psi|,$$

and

$$\boldsymbol{P}_j = |j\rangle\langle j|,$$

we have that

$$\boldsymbol{P}_j|\psi\rangle = |j\rangle\langle j| \sum_{i=0}^{n-1} \alpha_i |i\rangle = \sum_{i=0}^{n-1} \alpha_i |j\rangle\langle j|i\rangle,$$

which results in

$$\boldsymbol{P}_j|\psi\rangle = \alpha_j|j\rangle,$$

since $\langle j|i\rangle$. With the last equation we can rewrite

$$\boldsymbol{A}|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i\lambda_i|i\rangle = \sum_{i=0}^{n-1} \lambda_i \boldsymbol{P}_i|\psi\rangle,$$

from which we conclude that

$$\boldsymbol{A} = \sum_{i=0}^{n-1} \lambda_i \boldsymbol{P}_i.$$

This is the spectral decomposition of a hermitian and normal operator. It is true for any state and it is independent of the basis. The spectral decomposition can in turn be used to exhaustively specify a measurement, as we will see in the next section.

As an example, consider two states $|\psi_a\rangle$ and $|\psi_b\rangle$ that are eigenstates of $\boldsymbol{A}$ with eigenvalues λ_a and λ_b, respectively. In the diagonalization process we have obtained the coefficients α_0, α_1, β_0 and β_1 using an expansion in terms of the orthogonal basis $|0\rangle$ and $|1\rangle$. That is we have

$$|\psi_a\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle,$$

and

$$|\psi_b\rangle = \beta_0|0\rangle + \beta_1|1\rangle,$$

with corresponding projection operators

$$\boldsymbol{P}_a = |\psi_a\rangle\langle\psi_a| = \begin{bmatrix} |\alpha_0|^2 & \alpha_0\alpha_1^* \\ \alpha_1\alpha_0^* & |\alpha_1|^* \end{bmatrix},$$

and

$$\boldsymbol{P}_b = |\psi_b\rangle\langle\psi_b| = \begin{bmatrix} |\beta_0|^2 & \beta_0\beta_1^* \\ \beta_1\beta_0^* & |\beta_1|^* \end{bmatrix},$$

resulting in the following spectral decomposition of $\boldsymbol{A}$

$$\boldsymbol{A} = \lambda_a|\psi_a\rangle\langle\psi_a| + \lambda_b|\psi_b\rangle\langle\psi_b|,$$

which written out in all its details reads

$$\boldsymbol{A} = \lambda_a \begin{bmatrix} |\alpha_0|^2 & \alpha_0\alpha_1^* \\ \alpha_1\alpha_0^* & |\alpha_1|^* \end{bmatrix} + \lambda_b \begin{bmatrix} |\beta_0|^2 & \beta_0\beta_1^* \\ \beta_1\beta_0^* & |\beta_1|^* \end{bmatrix}.$$

Measurements

Armed with the spectral decomposition, we are now ready to discuss how to compute measurements of observables. When we make a measurement, quantum mechanics postulates that mutually exclusive measurement outcomes correspond to orthogonal projection operators.

We assume now we can contruct a series of such orthogonal operators based on $|i\rangle \in \{|0\rangle, |1\rangle, \ldots, |n-1\rangle$ computational basis states. These projection operators $\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_{n-1}$ are all idempotent and sum to one

$$\sum_{i=0}^{n-1} \boldsymbol{P}_i = \boldsymbol{I}.$$

As an example, consider the basis of two qubits $|0\rangle$ and $|1\rangle$ with the corresponding sum

$$\sum_{i=0}^{1} \boldsymbol{P}_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Based on the spectral decomposition discussed above, we can define the probability of eigenvalue λ_i as

$$\text{Prob}(\lambda_i) = |\boldsymbol{P}_i|\psi\rangle|^2,$$

where $|\psi_a\rangle$ is a quantum state representing the system prior to a specific measurement. We can rewrite this as

$$\text{Prob}(\lambda_i) = \langle\psi|\boldsymbol{P}_i^\dagger \boldsymbol{P}_i|\psi\rangle = \langle\psi|\boldsymbol{P}_i|\psi\rangle.$$

The total probability for all measurements is the sum overt all probabilities

$$\sum_{i=0}^{n-1} \text{Prob}(\lambda_i) = 1.$$

We can in turn define the post-measurement normalized pure quantum state as, for the specific outcome λ_i, as

$$|\psi'\rangle = \frac{\boldsymbol{P}_i|\psi\rangle}{\sqrt{\langle\psi|\boldsymbol{P}_i|\psi\rangle}}.$$

As an example, consider the binary system states $|0\rangle$ and $|1\rangle$ with corresponding projection operators

$$\boldsymbol{P}_0 = |0\rangle\langle 0|,$$

and

$$\boldsymbol{P}_1 = |1\rangle\langle 1|,$$

with the properties

$$\sum_{i=0}^{1} \boldsymbol{P}_i^\dagger \boldsymbol{P}_1 = \boldsymbol{I},$$

$$\boldsymbol{P}_0^\dagger \boldsymbol{P}_0 = \boldsymbol{P}_0^2 = \boldsymbol{P}_0,$$

and

$$\boldsymbol{P}_1^\dagger \boldsymbol{P}_1 = \boldsymbol{P}_1^2 = \boldsymbol{P}_1.$$

Assume thereafter that we have a state $|\psi\rangle$ which is a superposition of the above two qubit states

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

The probability of finding either $|0\rangle$ or $|1\rangle$ is then

$$\boldsymbol{P}_{\psi(0)} = \langle\psi|\boldsymbol{P}_0^\dagger \boldsymbol{P}_0|\psi\rangle = |\alpha|^2,$$

and similarly we have

$$\boldsymbol{P}_{\psi(1)} = \langle\psi|\boldsymbol{P}_1^\dagger \boldsymbol{P}_1|\psi\rangle = |\beta|^2.$$

If we set

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right),$$

we have $|\alpha|^2 = |\beta|^2 = 1/2$. In general for this system we have

$$|\psi_0'\rangle = \frac{\boldsymbol{P}_0|\psi\rangle}{\sqrt{\langle\psi|\boldsymbol{P}_0|\psi\rangle}} = \frac{\alpha}{|\alpha|}|0\rangle,$$

and

$$|\psi_1'\rangle = \frac{\boldsymbol{P}_1|\psi\rangle}{\sqrt{\langle\psi|\boldsymbol{P}_1|\psi\rangle}} = \frac{\beta}{|\beta|}|1\rangle.$$

In general we have that

$$\boldsymbol{P}_{\psi(x)} = \langle\psi|\boldsymbol{P}_x^\dagger \boldsymbol{P}_x|\psi\rangle, ,$$

which we can rewrite as

$$\text{Prob}(\psi(x)) = \text{Tr}\left[\boldsymbol{P}_x^\dagger \boldsymbol{P}_x|\psi\rangle\langle\psi|\right].$$

The last equation can be understood better through the following example with a state $|\psi\rangle$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

which results in a projection operator

$$|\psi\rangle\langle\psi| = \begin{bmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{bmatrix}.$$

We have that

$$\boldsymbol{P}_0^\dagger \boldsymbol{P}_0 = \boldsymbol{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

and computing the matrix product $\boldsymbol{P}_0|\psi\rangle\langle\psi|$ gives

$$\boldsymbol{P}_0|\psi\rangle\langle\psi| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{bmatrix} = \begin{bmatrix} |\alpha|^2 & \alpha\beta^* \\ 0 & 0 \end{bmatrix},$$

and taking the trace of this matrix, that is computing

$$\text{Prob}(\psi(0)) = \text{Tr}\left[\boldsymbol{P}_0^\dagger \boldsymbol{P}_0|\psi\rangle\langle\psi|\right] = |\alpha|^2,$$

we obtain the same results as the one we had earlier by computing the probablility for 0 given by the expression

$$\boldsymbol{P}_{\psi(0)} = \langle\psi|\boldsymbol{P}_0^\dagger \boldsymbol{P}_0|\psi\rangle = |\alpha|^2.$$

It is straight forward to show that

$$\text{Prob}(\psi(1)) = \text{Tr}\left[\boldsymbol{P}_1^\dagger \boldsymbol{P}_1|\psi\rangle\langle\psi|\right] = |\beta|^2,$$

which we also could have obtained by computing

$$\boldsymbol{P}_{\psi(1)} = \langle\psi|\boldsymbol{P}_1^\dagger \boldsymbol{P}_1|\psi\rangle = |\beta|^2.$$

We can now extend these expressions to the complete ensemble of measurements. Using the spectral decomposition we have that the probability of an outcome $p(x)$ is

$$p(x) = \sum_{i=0}^{n-1} p_i \boldsymbol{P}_{\psi_i(x)},$$

where p_i are the probabilities of a specific outcome. Add later a digression on marginal probabilities.

With these prerequisites we are now ready to introduce the density matrices, or density operators.

Density matrices/operators

The last equation can be rewritten as

$$p(x) = \sum_{i=0}^{n-1} p_i \boldsymbol{P}_{\psi_i(x)} = \sum_{i=0}^{n-1} p_i \text{Tr}\left[\boldsymbol{P}_x^\dagger \boldsymbol{P}_x|\psi_i\rangle\langle\psi_i|\right],$$

and we define the **density matrix/operator** as

$$\rho = \sum_{i=0}^{n-1} p_i|\psi_i\rangle\langle\psi_i|,$$

we can rewrite the first equation above as

$$p(x) = \text{Tr}\left[\boldsymbol{P}_x^\dagger \boldsymbol{P}_x \rho\right].$$

If we can define the state of a system in terms of the density matrix, the probability of a specific outcome is then given by

$$p(x)_\rho = \mathrm{Tr}\left[\boldsymbol{P}_x^\dagger \boldsymbol{P}_x \rho\right].$$

A density matrix in a Hilbert space with n states has the following properties (which we state without proof)

1. There exists a probability $p_i \geq 0$ with $\sum_i p_i = 1$,
2. There exists an orthonormal basis ψ_i such that we can define $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$,
3. We have $0 \leq \rho^2 \leq 1$ and
4. The norm $||\rho||_2 \leq 1$.

With the density matrix we can also define the state the system collapses to after a measurement, namely

$$\rho_x' = \frac{\boldsymbol{P}_x \rho \boldsymbol{P}_x^\dagger}{\mathrm{Tr}[\boldsymbol{P}_x^\dagger \boldsymbol{P}_x \rho]}.$$

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.

The Schmidt decomposition is again a linear decompostions which allows us to express a vector in terms of tensor product of two inner product spaces. In quantum information theory and quantum computing it is widely used as away to define and describe entanglement.

First entanglement encounter

Two-qubit system. 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

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

Bell states. The above computational basis states, which define an ONB, can in turn be used to define another ONB. As an example, consider the so-called Bell states

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}} \left[|00\rangle + |11\rangle \right] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}} \left[|00\rangle - |11\rangle \right] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix},$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}} \left[|10\rangle + |01\rangle \right] = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix},$$

and

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}} \left[|10\rangle - |01\rangle \right] = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}.$$

It is easy to convince oneself that these states also form an orthonormal basis.

Measuring one of the qubits of one of the above Bell states, automatically determines, as we will see below, the state of the second quabit. To convince ourselves about this, let us assume we perform a measurement on the qubit in system A by introducing the projections with outcomes 0 or 1 as

$$\boldsymbol{P}_0 = |0\rangle\langle 0|_A \otimes \boldsymbol{I}_B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

for the projection of the $|0\rangle$ state in system A and similarly

$$\boldsymbol{P}_1 = |1\rangle\langle 1|_A \otimes \boldsymbol{I}_B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

for the projection of the $|1\rangle$ state in system A.

We can then calculate the probability for the various outcomes by computing for example the probability for measuring qubit 0

$$\langle\Phi^+|\boldsymbol{P}_0|\Phi^+\rangle = \frac{1}{2}\left[\langle 00| + \langle 11|\right]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\left[|00\rangle + |11\rangle\right] = \frac{1}{2}.$$

Similarly, we obtain

$$\langle\Phi^+|\boldsymbol{P}_1|\Phi^+\rangle = \frac{1}{2}\left[\langle 00| + \langle 11|\right]\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\left[|00\rangle + |11\rangle\right] = \frac{1}{2}.$$

After the above measurements the system is in the states

$$|\Phi_0'\rangle = \sqrt{2}\left[|0\rangle\langle 0|_A \otimes \boldsymbol{I}_B\right]|\Phi^+\rangle = |00\rangle,$$

and

$$|\Phi_1'\rangle = \sqrt{2}\left[|1\rangle\langle 1|_A \otimes \boldsymbol{I}_B\right]|\Phi^+\rangle = |11\rangle.$$

We see from the last two equations that the state of the second qubit is determined even though the measurement has only taken place locally on system A.

If we on the other hand consider a state like

$$|00\rangle = |0\rangle_A \otimes |0\rangle_B,$$

this is a pure product state of the single-qubit, or single-particle states, of two qubits (particles) in system A and system B, respectively. We call such a state for a **pure state**. Quantum states that cannot be written as a mixture of other states are called pure quantum states, while all other states are called mixed quantum states.

A state like one of the Bell states (where we introduce the subscript AB to indicate that the state is composed of single states from two subsystem)

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\left[|00\rangle_{AB} + |11\rangle_{AB}\right],$$

is on the other hand a mixed state and we cannot determine whether system A is in a state 0 or 1. The above state is a superpostion of the states $|00\rangle_{AB}$ and $|11\rangle_{AB}$ and it is not possible to determine individual states of systems A and B, respectively.

We say that the state is entangled. This yields the following definition of entangled states: a pure bipartite state $|\psi\rangle_{AB}$ is entangled if it cannot be written as a product state $|\psi\rangle_A \otimes |\phi\rangle_B$ for any choice of the states $|\psi\rangle_A$ and $|\phi\rangle_B$. Otherwise we say the state is separable.

As an example, considere an ansatz for the ground state of the helium atom with two electrons in the lowest $1s$ state (hydrogen-like orbits) and with spin

$s = 1/2$ and spin projections $m_s = -1/2$ and $m_s = 1/2$. The two single-particle states are given by the tensor products of their spatial $1s$ single-particle states $|\phi_{1s}\rangle$ and and their spin up or spin down spinors $|\xi_{sm_s}\rangle$. The ansatz for the ground state is given by a Slater determinant with total orbital momentum $L = l_1 + l_2 = 0$ and totalt spin $S = s_1 + s_2 = 0$, normally labeled as a spin-singlet state. This ansatz for the ground state is then written as, using the compact notations

$$|\psi_i\rangle = |\phi_{1s}\rangle_i \otimes |\xi\rangle_{s_i m_{s_i}} = |1s, s, m_s\rangle_i,$$

with i being electron 1 or 2, and the tensor product of the two single-electron states as $|1s, s, m_s\rangle_1 |1s, s, m_s\rangle_2 = |1s, s, m_s\rangle_1 \otimes |1s, s, m_s\rangle_2$, we arrive at

$$\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2; s_1, s_2) = \frac{1}{\sqrt{2}} \left[|1s, 1/2, 1/2\rangle_1 |1s, 1/2, -1/2\rangle_2 - |1s, 1/2, -1/2\rangle_1 |1s, 1/2, 1/2\rangle_2 \right].$$

This is also an example of a state which cannot be written out as a pure state. We call this for an entangled state as well.

A so-called maximally entangled state for a bipartite system has equal probability amplitudes

$$|\Psi\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d-1} |ii\rangle.$$

We call a bipartite state composed of systems A and B (these systems can be single-particle systems, or single-qubit systems representing low-lying states of complicated many-body systems) for separable if its density matrix ρ_{AB} can be written out as the tensor product of the individual density matrices ρ_A and ρ_B, that is we have for a given probability distribution p_i

$$\rho_{AB} = \sum_i p_i \rho_A(i) \otimes \rho_B(i).$$

If we cannot write the density matrix in this form, we say the system AB is entangled. In order to see this, we can use the so-called Schmidt decomposition, which is essentially an application of the singular-value decomposition.

The Schmidt decomposition allows us to define a pure state in a bipartite Hilbert space composed of systems A and B as

$$|\psi\rangle = \sum_{i=0}^{d-1} \sigma_i |i\rangle_A |i\rangle_B,$$

where the ampltitudes σ_i are real and positive and their squared values sum up to one, $\sum_i \sigma_i^2 = 1$. The states $|i\rangle_A$ and $|i\rangle_B$ form orthornormal bases for systems A and B respectively, the amplitudes λ_i are the so-called Schmidt coefficients and the Schmidt rank d is equal to the number of Schmidt coefficients and is smaller or equal to the minimum dimensionality of system A and system B, that is $d \leq \min(\dim(A), \dim(B))$.

The proof for the above decomposition is based on the singular-value decomposition. To see this, assume that we have two orthonormal bases sets for

systems A and B, respectively. That is we have two ONBs $|i\rangle_A$ and $|j\rangle_B$. We can always construct a product state (a pure state) as

$$|\psi\rangle = \sum_{ij} c_{ij}|i\rangle_A|j\rangle_B,$$

where the coefficients c_{ij} are the overlap coefficients which belong to a matrix $\boldsymbol{C}$. If we now assume that the dimensionalities of the two subsystems A and B are the same d, we can always rewrite the matrix $\boldsymbol{C}$ in terms of a singular-value decomposition with unitary/orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ of dimension $d \times d$ and a matrix $\boldsymbol{\Sigma}$ which contains the (diagonal) singular values $\sigma_0 \leq \sigma_1 \leq \ldots 0$ as

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\dagger.$$

This means we can rewrite the coefficients c_{ij} in terms of the singular-value decomposition

$$c_{ij} = \sum_k u_{ik}\sigma_k v_{kj},$$

and inserting this in the definition of the pure state $|\psi\rangle$ we have

$$|\psi\rangle = \sum_{ij}\left(\sum_k u_{ik}\sigma_k v_{kj}\right)|i\rangle_A|j\rangle_B,$$

which we rewrite as

$$|\psi\rangle = \sum_k \sigma_k \left(\sum_i u_{ik}|i\rangle_A\right) \otimes \left(\sum_j v_{kj}|j\rangle_B\right),$$

which we identify simply as, since the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ represent unitary transformations,

$$|\psi\rangle = \sum_k \sigma_k |k\rangle_A|k\rangle_B.$$

It is straight forward to prove this relation in case systems A and B have different dimensionalities. Once we know the Schmidt decomposition of a state, we can immmediately say whether it is entangled or not. If a state ψ has is entangled, then its Schmidt decomposition has more than one term. Stated differently, the state is entangled if the so-called Schmidt rank is is greater than one. There is another important property of the Schmidt decomposition which is related to the properties of the density matrices and their trace operations and the entropies. In order to introduce these concepts let us look at the two-qubit Hamiltonian described here.

Density matrix and simple Hamiltonian

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

$$H_0|00\rangle = \epsilon_{00}|00\rangle,$$
$$H_0|10\rangle = \epsilon_{10}|10\rangle,$$
$$H_0|01\rangle = \epsilon_{01}|01\rangle,$$

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

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

Entropies and density matrices

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.

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 classical or Shannon information entropy is defined as

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

The quantum mechanical Von Neumann entropy is defined as

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

For a derivation of the these quantities see whiteboard notes

Pure states

A pure state has entropy equal zero while entangled state have an entropy larger than zero. The von-Neumann entropy for the subsystems 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.

```
%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 doum 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
```

Additional material

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:

$$|q\rangle \xrightarrow{U} U|q\rangle. \tag{1}$$

This action would be represented as the following quantum circuit

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

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

$$@C = 1em@R = 1em|q_0\rangle A \qquad B \tag{3}$$
$$|q_1\rangle C \qquad D \tag{4}$$

This circuit implies the following mathematical statement

$$|q_0 q_1\rangle \to (B \otimes D)(A \otimes C)|q_0 q_1\rangle \tag{5}$$
$$\to (BA) \otimes (DC)|q_0 q_1\rangle \tag{6}$$
$$\to BA|q_0\rangle DC|q_1\rangle. \tag{7}$$

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

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

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

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

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

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

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

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

and has the following matrix form

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

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

$$|x\rangle 1 \qquad 1 \qquad |y\rangle \tag{29}$$
$$|y\rangle \qquad -1 \qquad |x\rangle \tag{30}$$

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

$$R_z(\pi/2) \qquad R_y(\pi/2)1 \qquad U_1 1 \qquad R_y^*(\pi/2)R_z^*(\pi/2) \tag{33}$$
$$R_z(\pi/2) \qquad\qquad U_2 \qquad R_z^*(\pi/2) \tag{34}$$

Introduction to Qiskit

```
#!/usr/bin/env python
# coding: utf-8
import numpy as np
import qiskit as qk
from scipy.optimize import minimize


# # Initialize registers and circuit

n_qubits = 1 #Number of qubits
n_cbits = 1 #Number of classical bits (the number of qubits you want to measure at the end of the
qreg = qk.QuantumRegister(n_qubits) #Create a quantum register
creg = qk.ClassicalRegister(n_cbits) #Create a classical register
circuit = qk.QuantumCircuit(qreg,creg) #Create your quantum circuit


circuit.draw() #Draw circuit. It is empty
```

Thereafter we perform operations on qubit

```
circuit.x(qreg[0]) #Applies a Pauli X gate to the first qubit in the quantum register
circuit.draw()
```

and select a qubit to measure and encode the results to a classical bit

```
#Measure the first qubit in the quantum register
#and encode the results to the first qubit in the classical register
circuit.measure(qreg[0],creg[0])
circuit.draw()
```

Thereafter we execute the circuit

```
backend = qk.Aer.get_backend('qasm_simulator')
#This is the device you want to use. It is an ideal simulation of a quantum device


job = backend.run(circuit,shots=1000) #Run the circuit 1000 times
result = job.result()
counts = result.get_counts()
print(counts)
circuit.clear()
circuit.draw()


circuit.h(qreg[0]) #Apply a Hadamard gate to the first qubit of the quantum register
circuit.measure(qreg,creg)
print(circuit.draw())


job = backend.run(circuit,shots=1000)
result = job.result()
counts = result.get_counts()
print(counts)
circuit.clear()
```

Now we create a two-qubit circuit and set up a Bell state

```
n_qubits = 2
n_cbits = 2
qreg = qk.QuantumRegister(n_qubits)
creg = qk.ClassicalRegister(n_cbits)
circuit = qk.QuantumCircuit(qreg,creg)
circuit.draw()



circuit.h(qreg[0])
circuit.cx(qreg[0],qreg[1])
#This is a controlled operation. Apply a Pauli X gate to the second qubit (qreg[1]) if the first
#is in the |1> state. Else do nothing

circuit.draw()

circuit.measure(qreg,creg)
circuit.draw()


job = backend.run(circuit,shots=1000)
result = job.result()
counts = result.get_counts()
print(counts)
circuit.clear()
```

We apply a rotation to a qubit

```
theta = np.pi/3
circuit.rx(theta, qreg[0]) #R_x(theta) rotation on the first qubit (qreg[0])
circuit.measure(qreg,creg)
print(circuit.draw())
job = backend.run(circuit,shots=1000)
result = job.result()
counts = result.get_counts()
circuit.clear()
print(counts)
```

Now we want to find the lowest eigenvalue of

$$H = c_1 Z_0 + c_2 Z_1 + c_3 X_0 Y_1$$

We will use

$$< \psi | H | \psi >= c_1 < \psi | Z_0 | \psi > + c_2 < \psi | Z_1 | \psi > + c_3 < \psi | X_0 Y_1 | \psi >$$

```
I = np.eye(2)
X = np.array([[0,1],[1,0]])
Y = np.array([[0,-1j],[1j,0]])
Z = np.array([[1,0],[0,-1]])
H = np.kron(Z,I) + np.kron(I,Z) + np.kron(X,Y)
eigvals,eigvecs = np.linalg.eigh(H)
print(eigvals[0])
```

```
c_1 = 1
c_2 = 1
c_3 = 1

h_1 = [c_1,[0],['z']]
h_2 = [c_2,[1],['z']]
h_3 = [c_3,[0,1],['x','y']]
H = [h_1,h_2,h_3]
H
```

```
H[0]
```

We create the ansatz

```
def ansatz(theta,n_qubits):
    qreg = qk.QuantumRegister(n_qubits)
    circuit = qk.QuantumCircuit(qreg)
    for i in range(n_qubits):
        circuit.ry(theta[i],qreg[i])
    for i in range(n_qubits-1):
        circuit.cx(qreg[i],qreg[i+1])
    return(circuit)
qreg = qk.QuantumRegister(n_qubits)
circuit = qk.QuantumCircuit(qreg)
circuit.h(qreg[:2])
```

```
print('Before ansatz')
print(circuit.draw())
theta = np.random.randn(2)
n_qubits = 2
circuit = circuit.compose(ansatz(theta,n_qubits))
print('After ansatz')
circuit.draw()
```

Now we change measurement basis

```
def basis_change(h_i,n_qubits):
    qreg = qk.QuantumRegister(n_qubits)
    circuit = qk.QuantumCircuit(qreg)

    for qubit,operator in zip(h_i[1],h_i[2]):
        if operator == 'x':
            circuit.h(qreg[qubit])
        if operator == 'y':
            circuit.sdg(qreg[qubit])
            circuit.h(qreg[qubit])
    return(circuit)
n_qubits = 2
qreg = qk.QuantumRegister(n_qubits)
circuit = qk.QuantumCircuit(qreg)
theta = np.random.randn(n_qubits)
circuit = circuit.compose(ansatz(theta,n_qubits))
print('Ansatz circuit')
circuit.draw()
circuit = circuit.compose(basis_change(H[2],n_qubits))
print('After basis transformation:')
print(circuit.draw())
```

Get energy for given rotational parameters *theta*

```
def get_energy(theta):
    n_qubits = 2
    qreg = qk.QuantumRegister(n_qubits)
    circuit = qk.QuantumCircuit(qreg)
    circuit = circuit.compose(ansatz(theta,n_qubits))
    circuit_list = []
    for idx,h_i in enumerate(H):
        basis_change_circuit = basis_change(h_i,n_qubits)
        new_circuit = circuit.compose(basis_change_circuit)
        creg = qk.ClassicalRegister(len(h_i[1]))
        new_circuit.add_register(creg)
        new_circuit.measure(qreg[h_i[1]],creg)
        circuit_list.append(new_circuit)
    shots = 10000
    job = backend.run(circuit_list,shots=shots)
    E = np.zeros(len(circuit_list))
    for i in range(len(circuit_list)):
        result = job.result()
        counts = result.get_counts(i)
        for key,value in counts.items():
            e = 1
            for bit in key:
                if bit == '0':
                    e *= 1
```

```
                if bit == '1':
                    e *= -1
            E[i] += e*value
        E[i] *= H[i][0]
    E /= shots
    return(np.sum(E))

theta = np.random.randn(2)
get_energy(theta)
```

Minimize energy with Scipy

```
theta = np.random.randn(2)
res = minimize(get_energy, theta, method='Powell',tol=1e-12)
get_energy(res.x)
```

We define a more flexible ansatz

```
def ansatz(theta,n_qubits):
    qreg = qk.QuantumRegister(n_qubits)
    circuit = qk.QuantumCircuit(qreg)
    idx = 0
    for i in range(n_qubits):
        circuit.ry(theta[idx],qreg[i])
        idx += 1
    for i in range(n_qubits-1):
        circuit.cx(qreg[i],qreg[i+1])
    for i in range(n_qubits):
        circuit.rx(theta[idx],qreg[i])
        idx += 1
    for i in range(n_qubits-1):
        circuit.cx(qreg[i],qreg[i+1])
    return(circuit)
theta = np.random.randn(4)
res = minimize(get_energy, theta, method='Powell',tol=1e-16)
get_energy(res.x)
```

Minimize energy with gradient descent

$$\frac{\partial E(\theta_1,\ldots,\theta_i,\ldots,\theta_p)}{\partial\theta_i} = \frac{E(\theta_1,\ldots,\theta_i+\pi/2,\ldots,\theta_p) - E(\theta_1,\ldots,\theta_i-\pi/2,\ldots,\theta_p}{2}$$

```
epochs = 200
theta = np.random.randn(4)
for epoch in range(epochs):
    print(epoch,get_energy(theta))
    grad = np.zeros_like(theta)
    for idx in range(theta.shape[0]):
        theta_temp = theta.copy()
        theta_temp[idx] += np.pi/2
        E_plus = get_energy(theta_temp)
        theta_temp[idx] -= np.pi
        E_minus = get_energy(theta_temp)
        grad[idx] = (E_plus - E_minus)/2
    theta -= 0.1*grad
```