Notes on Quantum Computing

Nahum Sá

October 27, 2021

Abstract

These are my notes about quantum computing.

Contents

1 Quantum Information

This will be a short introduction on quantum information in order to have a theoretical basis to understand most Quantum Algorithms.

1.1 Postulates of Quantum Mechanics

As stated in Aaronson [1], there are two ways of introducing quantum mechanics: The physicist's way explaining the history behind the discovery of quantum theory and stating postulates of quantum theory as an endpoint, or showing that quantum mechanics is a generalization of probability theory. Here I will take the physicist's position and skip the history stuff and introduce the postulates of quantum theory.

The quantum mechanics postulates are, according to Nielsen and Chuang [14]:

1. Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the state space of the system. The system is completely described by its state vector, which is a unit vector in the system's state space.

2. The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2 , $|\psi(t_2)\rangle = U\,|\psi(t_1)\rangle$.

3. Quantum measurements are described by a collection of POVMs $\{M_m\}$ of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by:

   $$p(m) = \langle\psi|\, M_m^\dagger M_m \,|\psi\rangle \tag{1}$$

   And the system after the measurement is:

   $$\frac{M_m\,|\psi\rangle}{\langle\psi|\, M_m^\dagger M_m\,|\psi\rangle} \tag{2}$$

   Where $\sum_m M_m^\dagger M_m = \mathbb{I}$.

4. The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n: $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \ldots |\psi_n\rangle$

To sum up, the first postulate states that wave functions lives in Hilbert Space, the second one states that evolutions are unitary, the third one states that the wavefunction collapses when it's measured (For a good Everretian this might seems strange) and the final one states that composite systems are described by tensor products.

1.2 Qubit

One qubit is a subspace with two dimensions, that means we can map a qubit to $\mathbf{C}^2$. We can choose a basis in order to span this vector space, we will choose the standard computational basis that will be associated to bits:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad and \;\; |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3}$$

The general state of a qubit is a superposition of those states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{4}$$

This state can be parameterized by two parameters and construct what we call a "Bloch Sphere":

$$|\psi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} sin\left(\frac{\theta}{2}\right)|1\rangle \tag{5}$$

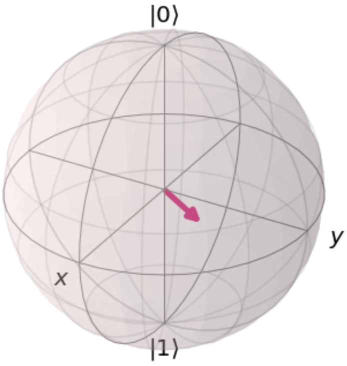


Figure 1: Graphical representation of the Bloch sphere.

1.3 Density Matrix

Density matrix is a generalization of quantum states (For the sake of simplicity, some people call the density matrix of a system the state of a system, because we can always purify it, see Sec 1.5). I will introduce this concept with an example that will be easily generalized.

Consider that you send a state $|+\rangle$ with probability p and a state $|0\rangle$ with probability $(1-p)$, where: $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, how would you describe this state as a ket? No!

Since you do not know which state came out exactly, you need to consider this uncertainty into our formulation, this is done using a density matrix ρ:

$$\rho = p\,|+\rangle\langle+| + (1-p)\,|0\rangle\langle 0| = \frac{p}{2}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix} + (1-p)\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} = \begin{pmatrix}1-\frac{p}{2} & \frac{p}{2}\\ \frac{p}{2} & \frac{p}{2}\end{pmatrix} \tag{6}$$

Let's consider $p = \frac{1}{2}$:

$$\rho = \begin{pmatrix}\frac{3}{4} & \frac{1}{4}\\ \frac{1}{4} & \frac{1}{4}\end{pmatrix} \tag{7}$$

What is the probability to measure the state $|0\rangle$?

By intuition since we have probability $\frac{1}{2}$ of finding the state on $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and probability $\frac{1}{2}$ of finding the state on $|0\rangle$, the probability of measuring $|0\rangle$ in the state $|+\rangle$ is $\frac{1}{2}$ given by the second postulate of quantum mechanics, therefore the probability is given by:

$$p(0) = \frac{1}{2}\overbrace{\frac{1}{2}}^{|+\rangle} + \frac{1}{2}\overbrace{1}^{|0\rangle} = \frac{3}{4} = \langle 0|\,\rho\,|0\rangle = Tr(\rho\,|0\rangle\langle 0|) \tag{8}$$

Exercise 1: What is the probability of measuring the state $|1\rangle$?

Exercise 2: What is the probability of measuring the state $|1\rangle$ or the state $|0\rangle$?

Now we can generalize what a density matrix is. Consider a quantum system with i states $|\psi_i\rangle$ with respective probabilities p_i. The density matrix of this system is given by:

$$\rho = \sum_i p_i\,|\psi_i\rangle\langle\psi_i| \tag{9}$$

The evolution of a quantum state is given by an unitary transformation U, then the evolution of the density matrix is given by:

$$\rho = \sum_i p_i\,|\psi_i\rangle\langle\psi_i| \xrightarrow{U} \sum_i p_i U\,|\psi_i\rangle\langle\psi_i|\,U^\dagger = U\rho U^\dagger \tag{10}$$

Measurements, as shown above, also can be generalized for the density operator formalism. Suppose the measurement operators M_m. If the initial state was $|\psi_i\rangle$, the probability of measuring m given i is:

$$p(m|i) = \langle\psi_i| M_m^\dagger M_m |\psi_i\rangle = Tr(M_m^\dagger M_m |\psi_i\rangle \langle\psi_i|) \tag{11}$$

By the laws of probability:

$$\begin{aligned} p(m) &= \sum_i p(m|i)p_i \\ &= \sum_i p_i \; Tr(M_m^\dagger M_m |\psi_i\rangle \langle\psi_i|) \\ &= Tr(M_m^\dagger M_m \rho) \end{aligned} \tag{12}$$

Therefore, if you want to know the value of any observable A, you have:

$$\langle A\rangle = Tr(A\rho) \tag{13}$$

The class of operators that are density operators are characterized by the following useful theorem(From [14]):

Theorem 1.1 (Characterization of Density Operators)**.** *An operator ρ is the density operator associated to some ensemble $\{ \; p_i \; , |\psi_i\rangle \; \}$ if and only if it satisfies the conditions:*

- ***Unity Trace**: $Tr(\rho) = 1$*
- ***Positivity**: $\rho \geq 0$*

Proof. Suppose $\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|$. Then

- **Unity Trace**:

$$Tr(\rho) = Tr(\sum_i p_i |\psi_i\rangle \langle\psi_i|) = \sum_i p_i \overbrace{Tr(|\psi_i\rangle \langle\psi_i|)}^{=1} = \sum_i p_i = 1 \tag{14}$$

- **Positivity**: Suppose $|\phi\rangle$ is an arbitrary state. Then:

$$\begin{aligned} \langle\phi| \rho |\phi\rangle &= \sum_i p_i \langle\phi|\psi_i\rangle \langle\psi_i|\phi\rangle \\ &= \sum_i p_i | \langle\phi|\psi_i\rangle |^2 \\ &\geq 0 \end{aligned} \tag{15}$$

Now suppose ρ is an operator that the trace is unity and is positive. Since ρ is positive, it must have a spectral decomposition:

$$\rho = \sum_i \lambda_i |i\rangle \langle i| \tag{16}$$

From the unity of the trace, we have that $\sum_i \lambda_i = 1$. Therefore, we have the ensemble $\{\ \lambda_i\ ,\ |i\rangle\ \}$ that gives rise to the density operator ρ. □

Question: Can we distinguish between two ensembles?

No! For example the ensembles:

1. $\{\ (\frac{1}{2}, |0\rangle), (\frac{1}{2}, |1\rangle)\ \} \rightarrow\ \ \rho = \frac{1}{2}(|0\rangle\langle 0| + |0\rangle\langle 0|) = \frac{\mathbb{1}}{2}$
2. $\{\ (\frac{1}{2}, |+\rangle), (\frac{1}{2}, |-\rangle)\ \} \rightarrow\ \ \rho = \frac{1}{2}(|+\rangle\langle +| + |-\rangle\langle -|) = \frac{\mathbb{1}}{2}$

Exercise: Work out the details of the above ensembles.

We can discriminate between two types of states:

- **Pure States**: States that we have complete knowledge.

$$\rho = |\psi_i\rangle\langle\psi_i|\ \ ,\ p_i = 1\ ,\ p_{j\neq i} = 0 \tag{17}$$

- **Mixed States**: States that we do not have complete knowledge.

$$\rho = \sum_i |\psi_i\rangle\langle\psi_i| \tag{18}$$

With at least two p_i's that are different than 0.

We can quantify the purity of a state using the following measure:

$$P(\rho) = Tr(\rho^2) \tag{19}$$

Example 1 (Pure State): $\rho = |\psi\rangle\langle\psi|$

$$P(|\psi\rangle\langle\psi|) = Tr((|\psi\rangle\langle\psi|)^2) = Tr(|\psi\rangle\overbrace{\langle\psi|\,|\psi\rangle}^{=1}\langle\psi|) = Tr(|\psi\rangle\langle\psi|) = 1 \tag{20}$$

Example 2 (Mixed State): Since ρ is hermitian we can consider its spectral decomposition $\rho = \sum_i \lambda_i |\phi_i\rangle\langle\phi_i|$, where $\langle\phi_i|\phi_j\rangle = \delta_{ij}$, $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$. Therefore:

$$P(\rho) = Tr((\sum_i \lambda_i |\phi_i\rangle\langle\phi_i|)^2) = Tr(\sum_i\sum_j \lambda_i\lambda_j |\phi_i\rangle\overbrace{\langle\phi_i|\,|\phi_j\rangle}^{=\delta_{ij}}\langle\phi_j|) = Tr(\sum_i \lambda_i^2 |\phi_i\rangle\langle\phi_i|) = \sum_i \lambda_i^2 \leq 1 \tag{21}$$

Exercise: Find the purity of the maximally mixed state $\rho = \mathbb{1}/d$, where d is the dimension of the finite Hilbert Space.

If you tried to do the exercise, you will find that the purity of a d-dimensional state lies between two fixed values $\frac{1}{d} \leq P(\rho) \leq 1$.

1.4 Qubit Revisited

In section 1.2 we introduced the concept of qubits, but we didn't have the density operator formalism yet, now we take a more detailed look on qubits and the bloch sphere.

The density matrix of a qubit is a 2x2 matrix:

$$\rho = \begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix} = \sum_{i=0}^{1}\sum_{j=0}^{1} \rho_{ij} \left|i\right\rangle \left\langle j\right| \tag{22}$$

Since it is a density matrix, we have that: $\rho_{00} + \rho_{11} = 1$ and $\rho_{ij}^* = \rho_{ji}$

We can expand the density matrix on the pauli matrices basis: $\mathcal{B}_{2\text{x}2} = \{\mathbb{1}, \sigma_x, \sigma_y, \sigma_z\}$.

The Pauli Matrices are:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad , \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{23}$$

And they have the following properties:

- Hermitian: $\sigma_i = \sigma_i^\dagger$, $\forall i$;
- $Tr(\sigma_i) = 0$, $\forall i \in \{x, y, z\}$;
- $Tr(\sigma_i^\dagger \sigma_j) = 2\delta_{ij}$;
- $\sigma_i \sigma_j = \delta_{ij}\mathbb{1} + i\epsilon_{ijk}\sigma_k$, $i, j, k \in \{x, y, j\}$, in particular $\sigma_i^2 = \mathbb{1}$.

Writing the qubit on the Pauli Matrices basis, we have:

$$\rho = r_0\mathbb{1} + r_1\sigma_x + r_2\sigma_y + r_3\sigma_z \tag{24}$$

Applying the unit trace condition, we have:

$$Tr(\rho) = 1 \Rightarrow 2r_0 = 1 \Rightarrow r_0 = \frac{1}{2} \tag{25}$$

Therefore:

$$\rho = \frac{1}{2}\mathbb{1} + r_1\sigma_x + r_2\sigma_y + r_3\sigma_z \tag{26}$$

Using that the density matrix is Hermitian, we have that:

$$\rho = \rho^\dagger \Rightarrow \frac{1}{2}\mathbb{1} + r_1\sigma_x + r_2\sigma_y + r_3\sigma_z = \frac{1}{2}\mathbb{1} + r_1^*\sigma_x + r_2^*\sigma_y + r_3\sigma_z^* \Rightarrow r_1, r_2, r_3 \in \mathbb{R} \tag{27}$$

We choose $r_1 = \frac{1}{2}r_x$, $r_2 = \frac{1}{2}r_y$, $r_3 = \frac{1}{2}r_z$, then our qubit density matrix is written as:

$$\rho = \frac{1}{2}\Big(\mathbb{1} + r_x\sigma_x + r_y\sigma_y + r_z\sigma_z\Big) = \frac{1}{2}\Big(\mathbb{1} + \mathbf{r}\cdot\sigma\Big) \tag{28}$$

Writing as a matrix, we have:

$$\rho = \frac{1}{2}\begin{pmatrix} 1 + r_z & r_x - ir_y \\ r_x + ir_y & 1 - r_z \end{pmatrix} \tag{29}$$

So now, in order to know where the state is in the Bloch sphere we just write the density matrix and find r_x, r_y and r_z.

It is interesting to note that pure and mixed states stay on different regions of the Bloch sphere, let's calculate the purity of an generic qubit:

$$\begin{aligned} P(\rho) = Tr(\rho^2) = Tr\left[\left(\frac{1}{2}(\mathbb{1} + \mathbf{r}\cdot\sigma\right)^2\right] &= \frac{1}{4}\,Tr\Big(\mathbb{1} + \mathbf{r}\cdot\sigma\Big)\Big(\mathbb{1} + \mathbf{r}\cdot\sigma\Big) \\ &= \frac{1}{4}\,Tr\Big(\mathbb{1} + 2\mathbf{r}\cdot\sigma + \sum_{ij} r_i r_j \sigma_i\sigma_j\Big) \\ &= \frac{1}{4}\,Tr\Big(\mathbb{1} + 2\mathbf{r}\cdot\sigma + \sum_{i\neq j} r_i r_j \sigma_i\sigma_j + \sum_i r_i^2 \overbrace{\sigma_i^2}^{=\mathbb{1}}\Big) \\ &= \frac{1}{4}(2 + 2\sum_i r_i^2) = \frac{1}{2}(1 + |\mathbf{r}|^2) \end{aligned} \tag{30}$$

Therefore we have two situations:

- Pure States: If $|\mathbf{r}| = 1$, therefore it is in the spherical shell.

- Mixed States: If $|\mathbf{r}| < 1$, those states are inside the Bloch sphere in a spherical shell(”isopure shells”).

1.5 Purification

Given a density matrix $\rho_A \in \mathcal{L}(\mathcal{H}_A)$, there always is an pure state $|\psi_{AB}\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ such that: $\rho_A = Tr_B(|\psi_{AB}\rangle \langle\psi_{AB}|)$.

For instance, consider $\rho_A = \sum_{i=1}^N p_i |\phi_i\rangle \langle\phi_i|$ with $p_i \geq 0$ and $\sum_{i=1}^N p_i = 1$. We can suppose that there is an orthonormal basis $\{|i\rangle\}_{i=1}^N \in \mathcal{H}_B$, then we can take the pure state as:

$$|\psi_{AB}\rangle = \sum_{i=1}^N \sqrt{p_i} |\phi_i\rangle \otimes |i\rangle \tag{31}$$

Checking that this is a purification of ρ_A:

$$\begin{aligned} Tr_B(|\psi_{AB}\rangle \langle\psi_{AB}|) &= Tr_B\big\{ \sum_{i,j} \sqrt{p_i p_j} |\phi_i\rangle \langle\phi_j| \otimes |i\rangle \langle j| \,\big\} \\ &= \sum_{i,j} \sqrt{p_i p_j} |\phi_i\rangle \langle\phi_j| \overbrace{Tr\big(|i\rangle \langle j| \big)}^{\delta_{ij}} \\ &= \sum_i p_i |\phi_i\rangle \langle\phi_i| \\ &= \rho_A \end{aligned} \tag{32}$$

It is important to note that there are infinite purifications for the density matrix $\rho_A \in \mathcal{L}(\mathcal{H}_A)$.

Theorem 1.2 (Non unicity of purifications)**.** *If $|\psi_{AB}\rangle$ is such that $\rho_A = Tr_B(|\psi_{AB}\rangle \langle\psi_{AB}|)$, then $\mathbb{1} \otimes U_B |\psi_{AB}\rangle$ is also a valid purification with $U_B^\dagger U_B = \mathbb{1}$.*

Proof. Suppose that $\rho_A = Tr_B(|\psi_{AB}\rangle \langle\psi_{AB}|)$, then:

$$\begin{aligned} &Tr_B\big\{(\mathbb{1} \otimes U_B) |\psi_{AB}\rangle \langle\psi_{AB}| (\mathbb{1} \otimes U_B^\dagger)\big\} = \\ &Tr_B\big\{ \, |\psi_{AB}\rangle \langle\psi_{AB}| (\mathbb{1} \otimes U_B^\dagger)(\mathbb{1} \otimes U_B)\big\} = \\ &Tr_B(|\psi_{AB}\rangle \langle\psi_{AB}|) = \rho_A \end{aligned} \tag{33}$$

Where we used the cyclic property of the trace. □

1.6 Entanglement

Definition 1.1 (Product State)**.** A given state $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ is said to be a product state if $\exists\, |\phi\rangle \in \mathcal{H}_A$ and $\exists\, |\chi\rangle \in \mathcal{H}_B$, such that: $|\psi\rangle = |\phi\rangle \otimes |\chi\rangle$.

Example 1 (Product State): Consider the following state:

$$\begin{aligned} |\psi\rangle =& \frac{1}{2}\Bigg(|00\rangle + |01\rangle \Bigg) \\ =& \frac{1}{2}\Bigg(|0\rangle \otimes (|0\rangle + |1\rangle)\Bigg) \end{aligned} \tag{34}$$

Definition 1.2 (Entangled State)**.** A given state $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ is said to be entangled if $\nexists\, |\phi\rangle \in \mathcal{H}_A$ and $\nexists\, |\chi\rangle \in \mathcal{H}_B$, such that: $|\psi\rangle = |\phi\rangle \otimes |\chi\rangle$.

Example 1 (Entangled State): Consider the following state:

$$\begin{aligned} |\psi\rangle =& \frac{1}{2}\Bigg(|00\rangle + |01\rangle + |10\rangle - |11\rangle \Bigg) \\ =& \frac{1}{2}\Bigg(|0\rangle \otimes (|0\rangle + |1\rangle) + |1\rangle \otimes (|0\rangle - |1\rangle)\Bigg) \end{aligned} \tag{35}$$

1.7 No-go Theorems

1.7.1 No communication Theorem

Consider that you have a composite physical system with $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ and those states are finite dimensional. A general state on $\mathcal{H}$ is given by: $\rho = \sum_i \sigma_i \otimes \omega_i$, where $\sigma_i \in \mathcal{L}(\mathcal{H}_A)$ and $\omega_i \in \mathcal{L}(\mathcal{H}_B)$.

The observer in A makes a measurement only on her subsystem:

$$P(\rho) = \sum_j (K_j \otimes \mathbb{1}_B)^\dagger \rho (K_j \otimes \mathbb{1}_B) \tag{36}$$

Where K_j are Kraus Operators(See 1.8) with the property: $\sum_j K_j K_j^\dagger = \mathbb{1}$.

Now the state for an observer in B is given by the partial trace over A:

$$\rho_B = Tr_A(P(\rho)) \tag{37}$$

So let's see if there is any difference on the B state:

$$\begin{aligned}
\rho_B &= Tr_A(\sum_j (K_j \otimes \mathbb{1}_B)^\dagger \rho (K_j \otimes \mathbb{1}_B)) \\
&= Tr_A(\sum_i \sum_j K_j^\dagger \sigma_i K_j \otimes \omega_i) \\
&= \sum_i \sum_j Tr(K_j^\dagger \sigma_i K_j) \otimes \omega_i \\
&= \sum_i Tr(\overbrace{\sum_j K_j K_j^\dagger}^{=\mathbb{1}} \sigma_i) \otimes \omega_i \\
&= \sum_i Tr(\sigma_i) \otimes \omega_i \\
&= Tr_A(\rho)
\end{aligned} \tag{38}$$

Since the state before and after the local operations are indistinguishable for B, an observer in A doesn't have a way to communicate with and observer in B by using local operations. **There is no faster than light communication on Quantum Mechanics!**.

1.7.2 No Cloning Theorem

Theorem 1.3 (No cloning Theorem)**.** *There is no unitary U acting on $\mathcal{H} \otimes \mathcal{H}$ such that for all $|\phi\rangle \in \mathcal{H}$ we have:*

$$U |\phi\rangle \otimes |0\rangle = |\phi\rangle \otimes |\phi\rangle \tag{39}$$

This theorem can be represented by the following circuit:

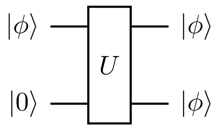


Figure 2: No-Cloning Theorem Circuit

Proof. This theorem is proved by contradiction. Suppose that exists this unitary, so given $|\phi\rangle, |\psi\rangle \in \mathcal{H}$, we have that:

$$\begin{aligned} U|\phi\rangle \otimes |0\rangle &= |\phi\rangle \otimes |\phi\rangle \\ U|\psi\rangle \otimes |0\rangle &= |\psi\rangle \otimes |\psi\rangle \end{aligned} \tag{40}$$

Taking the scalar product:

$$\begin{aligned} (\langle\phi| \otimes \langle\phi|)(|\psi\rangle \otimes |\psi\rangle) &= (\langle\phi| \otimes \langle 0| U^\dagger)(U|\psi\rangle \otimes |0\rangle) \\ \langle\phi|\psi\rangle \langle\psi|\phi\rangle &= \langle\phi|\psi\rangle \langle 0|0\rangle \\ \Rightarrow \langle\phi|\psi\rangle^2 &= \langle\phi|\psi\rangle \end{aligned} \tag{41}$$

This is a equation $x^2 - x = 0$, so there are two possible solutions:

- $\langle\phi|\psi\rangle = 0$, therefore $|\phi\rangle$, $|\psi\rangle$ are ortogonal;

- $\langle\phi|\psi\rangle = 1$, therefore $|\phi\rangle$, $|\psi\rangle$ are identical;

This is a contradiction, because we supposed that U would work for every state.

□

The No-Cloning theorem is really important in Quantum Cryptography. Because you cannot clone quantum states, it is impossible for a eavesdropper to hear your communication without you knowing. See BB84 Protocol [6].

1.8 Channels

Suppose a system with an environment state on a global unitary $U : \mathcal{H}_A \otimes \mathcal{H}_B \rightarrow \mathcal{H}_A \otimes \in H_B$:

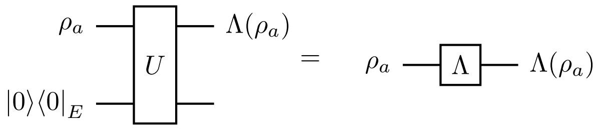


Figure 3: Representation of quantum channels.

So we throw away the E part because we are only interested on part A. Therefore we will trace out the environment E:

$$\lambda(\rho_A) = Tr_B(U \rho_A \otimes |0\rangle\langle 0|_E \, U^\dagger) \tag{42}$$

Let's derive the formula for this channel:

$$U = \sum_{k,l,m,n} U^{kl}_{mn} \, |kl\rangle\langle mn| \tag{43}$$

Using equation 43 on equation 42:

$$\begin{aligned}
\Lambda(\rho_A) &= \sum_i (\mathbb{1} \otimes \langle i|) \left[\sum_{k,l,m,n} U^{kl}_{mn} \, |kl\rangle\langle mn| \, (\rho_A \otimes |0\rangle\langle 0|_E) \left(\sum_{r,s,p,q} U^{rs}_{pq} \, |rs\rangle\langle pq| \right)^\dagger \right] (\mathbb{1} \otimes |i\rangle) \\
&= \sum_i (\mathbb{1} \otimes \langle i|) \left[\sum_{k,l,m,n} U^{kl}_{mn} \, |kl\rangle\langle mn| \, (\rho_A \otimes |0\rangle\langle 0|_E) \left(\sum_{r,s,p,q} (U^{rs}_{pq})^* \, |pq\rangle\langle rs| \right) \right] (\mathbb{1} \otimes |i\rangle) \\
&= \sum_i \sum_{k,l,m,n} \sum_{r,s,p,q} U^{kl}_{mn} (U^{rs}_{pq})^* \, |k\rangle\langle m| \, \rho_A \, |p\rangle\langle r| \otimes \langle i|l\rangle \, \langle n|0\rangle \, \langle 0|q\rangle \, \langle s|i\rangle \\
&= \sum_i \sum_{k,m} \sum_{r,p} U^{ki}_{m0} (U^{ri}_{p0})^* \, |k\rangle\langle m| \, \rho_A \, |p\rangle\langle r| \\
&= \sum_i \left(\sum_{k,m} U^{ki}_{m0} \, |k\rangle\langle m| \right) \rho_A \left(\sum_{r,p} U^{ri}_{p0} \, |p\rangle\langle r| \right)^\dagger \\
&\equiv \sum_i K_i \rho_A K_i^\dagger
\end{aligned} \tag{44}$$

Then we define the Kraus Operator $K_i : \mathcal{H}_A \rightarrow \mathcal{H}_A$:

$$K_i = \sum_{k,m} U^{ki}_{m0} \;\; |k\rangle\langle m| = \langle i|_E \, U \, |0\rangle_E \tag{45}$$

Properties of quantum channels:

- Linearity:

$$\Lambda(\sum_j p_j \rho_j) = \sum_i K_i (\sum_j p_j \rho_j) K_i^\dagger = \sum_j \left(\sum_i K_i \rho_j K_i^\dagger \right) = \sum_j p_j \Lambda(\rho_j) \tag{46}$$

- Preserves Hermicity: If $\rho = \rho^\dagger \Rightarrow \Lambda(\rho)^\dagger = \Lambda(\rho)$

$$\Lambda(\rho)^\dagger = \left(\sum_i K_i \rho K_i^\dagger\right)^\dagger = \sum_i K_i \rho K_i^\dagger = \Lambda(\rho) \tag{47}$$

- Preserves the Trace:

$$Tr(\Lambda(\rho)) = Tr(\sum_i K_i \rho K_i^\dagger) = Tr(\sum_i K_i^\dagger K_i \rho) \tag{48}$$

Using $K_i = \langle i|_B U |0\rangle_B$, we have that:

$$\sum_i K_i^\dagger K_i = \sum_i \langle 0|_B U^\dagger |i\rangle_B \langle i|_B U |0\rangle_B = \langle i|_B U^\dagger U |0\rangle_B = \langle i|_B |0\rangle_B = \mathbb{1}_A \tag{49}$$

Therefore:

$$Tr(\Lambda(\rho)) = Tr(\rho) \tag{50}$$

- Preserves Positivity: $\rho \geq 0 \Rightarrow \Lambda(\rho) \geq 0$

$$\Lambda(\rho) = \sum_i K_i \rho K_i^\dagger = \sum_i K_i \sqrt{\rho}\sqrt{\rho} K_i^\dagger \tag{51}$$

Then, $\forall \, |\psi\rangle$:

$$\langle\psi| \Lambda(\rho) |\psi\rangle = \sum_i \langle\psi| K_i \sqrt{\rho}\sqrt{\rho} K_i^\dagger |\psi\rangle = \sum_i ||\sqrt{\rho} K_i |\psi\rangle || \geq 0 \tag{52}$$

But in order to define an physically valid quantum channel, we have that the channel must return states even when applied to subsystems.

Channels which $\forall n \in \mathbb{N} : \Lambda \otimes \mathbb{1}_n(\rho) \geq 0$ are said to be Completely Positive Maps.

1.9 Quantum Measures

1.9.1 Trace Distance

Trace distance can be interpreted as the probability of distinguishing between two density matrices. It is defined as the Schatten Norm for p=1:

$$T(\rho,\sigma) = \frac{1}{2}||\rho-\sigma||_1 = \frac{1}{2}\mathrm{Tr}\left[\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\right] \tag{53}$$

Since density matrices are Hermitian, we have that:

$$T(\rho,\sigma) = \frac{1}{2}\mathrm{Tr}\left[\sqrt{(\rho-\sigma)^2}\right] = \frac{1}{2}\sum_i \lambda_i \tag{54}$$

Where λ_i are the eigenvalues of $(\rho-\sigma)$.

Properties [14]:

- Is a metric on the space of density matrices.
- $0 \leq T(\rho,\sigma) \leq 1$ and $T(\rho,\sigma) = 1$ iff ρ and σ are orthogonal.
- It is Preserved under unitary transformations: $T(U\rho U^\dagger, U\sigma U^\dagger) = T(\rho,\sigma)$.
- It is convex in each of its inputs. $T(\rho, \sum_i p_i\sigma_i) \leq \sum_i p_i T(\rho,\sigma_i)$.

1.9.2 Von Neumann Entropy

Von Neumann entropy of quantum state ρ is defined as:

$$S(\rho) \equiv -\mathrm{Tr}(\rho \mathrm{log}\rho) \tag{55}$$

If λ_x are the eigenvalues of ρ, then von Neumann's entropy definition can be re-expressed as:

$$S(\rho) = -\sum_x \lambda_x \mathrm{log}\lambda_x \tag{56}$$

And we define $0\mathrm{log}0 \equiv 0$.

1.9.3 Quantum relative entropy

The Von Neumann Entropy of a density matrix is $S(\rho) = -\mathrm{Tr}\ \rho\ \mathrm{log}\rho$, then we can define the quantum relative entropy between two density matrices ρ and σ as [17], [14]:

$$S(\rho||\sigma) = -\mathrm{Tr}\big[\rho\ \mathrm{log}\sigma\big] - S(\rho) = \mathrm{Tr}\big[\ \rho\ \mathrm{log}\rho\big] - \mathrm{Tr}\big[\ \rho\ \mathrm{log}\sigma\big] = \mathrm{Tr}\big[\ \rho(\ \mathrm{log}\rho -\ \mathrm{log}\sigma)\big] \tag{57}$$

This can be used as an entanglement measurement, consider a composite finite Hilbert space $H = \bigotimes_k H_k$, the relative entropy of entanglement of ρ is defined by:

$$D_{\mathrm{REE}}(\rho) = \min_{\sigma} S(\rho||\sigma) \tag{58}$$

The minimum is taken over the family of separable states ($\rho = \sum_{i,j} p_{ij}\sigma_i \otimes \sigma_j$). We can intepret this as the quantity of distinguishability of the state ρ from separable states. When $D_{REE} = 0$ we have a separable state, because of Klein's inequality.

Theorem 1.4 (Klein's Inequality)**.**

$$S(\rho||\sigma) \geq 0 \text{ and } S(\rho||\sigma) = 0 \iff \rho = \sigma \tag{59}$$

Proof. [14] Let $\rho = \sum_i p_i |i\rangle\langle i|$ and $\sigma = \sum_j q_j |j\rangle\langle j|$ be the orthonormal compositions for ρ and σ. From the definition, we have that:

$$S(\rho||\sigma) = \sum_i p_i \mathrm{log} p_i - \sum_i \langle i| \rho \mathrm{log}\sigma |i\rangle \tag{60}$$

Let's simplify $\langle i| \rho \mathrm{log}\sigma |i\rangle$:

$$\langle i| \rho \mathrm{log}\sigma |i\rangle = \langle i| \left(\sum_j \log(q_j) |j\rangle\langle j| \right) |i\rangle = \sum_i \log(q_j) P_{ij} \tag{61}$$

Therefore, for the relative entropy, we have:

$$S(\rho||\sigma) = \sum_i p_i \mathrm{log} p_i - \sum_i \sum_j p_i \log(q_j) P_{ij} = \sum_i p_i \left(\log(p_i) - \sum_j \log(q_j) P_{ij} \right) \tag{62}$$

Note that the matrix P_{ij} has double stochasticity ($\sum_i P_{ij} = 1$ and $\sum_j P_{ij} = 1$) and $P_{ij} \geq 0$. Because log is a strictly concave function we have that $\sum_j P_{ij} \log(q_j) \leq log(r_i)$, where $r_i \equiv \sum_j P_{ij} q_j$ with equality iff $\exists j : P_{ij} = 1$. Therefore:

$$S(\rho||\sigma) \geq \sum_i p_i \mathrm{log} \frac{p_i}{r_i} \tag{63}$$

With the equality iff for each i there exists an j such that $P_{ij} = 1$, that is iff P_{ij} is a permutation matrix. This has the same form of the classical relative entropy, we have that:

$$S(\rho||\sigma) \geq 0 \tag{64}$$

With equality iff $p_i = r_i$ $\forall i$ and P_{ij} is a permutation matrix. For this condition we can relabel the eigenstates of σ such that P_{ij} becomes the identity matrix, therefore ρ and σ are diagonal on the same basis and have the same eigenvalues, thus $\rho = \sigma$ iff $S(\rho||\sigma) = 0$. □ □

1.9.4 Quantum mutual information

Consider a bipartite quantum system $H_{AB} = H_A \otimes H_B$ and let $\rho_{AB} \in \mathcal{L}(H_{AB})$. As defined on section 1.9.3, the Von Neumann Entropy is $S(\rho) = -\mathrm{Tr}\ \rho\ \mathrm{log}\rho$. The reduced state of ρ_{AB} on a subsystem A is given by the partial trace: $\rho^A = \mathrm{Tr}_B \rho_{AB}$. The reduced Von Neumann Entropy is given by $S(\rho^A)$. Now the Quantum mutual informatin is defined is:

$$I(A:B) = S(\rho^A) + S(\rho^B) - S(\rho^{AB}) = S(\rho^{AB}||\rho^A \otimes \rho^B) \tag{65}$$

Where $S(\cdot||\cdot)$ is the quantum relative entropy (Sec 1.9.3).

2 Quantum Computing

Now we will focus on Quantum Computing and Algorithms for the Circuit model.

2.1 Quantum Gates

2.1.1 One Qubit Gates

Since everything must be done in the realm of Quantum Mechanics the one qubit gates should be unitary 2x2 matrices, following the 2nd postulate.

The first gate is the *Hadamard Gate*, defined as the matrix:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{66}$$

This gate changes the computational basis $\{0, 1\}$ (Z basis) to a new basis: $\{+, -\}$ (X basis).

$$\begin{aligned} H\left|0\right\rangle &= \frac{1}{\sqrt{2}}\big(\left|0\right\rangle + \left|1\right\rangle\big) \equiv \left|+\right\rangle \\ H\left|0\right\rangle &= \frac{1}{\sqrt{2}}\big(\left|0\right\rangle - \left|1\right\rangle\big) \equiv \left|-\right\rangle \end{aligned} \tag{67}$$

Another important property of the Hadamard gate is that $H^2 = \mathbb{1}$, so the inverse transformation is the Hadamard gate itself, $H^\dagger = H$.

The next gate is the *Phase Shift Gate*:

$$R_z(\delta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix} \tag{68}$$

This gate adds a phase only if the state is $\left|1\right\rangle$, Therefore for a general state $\left|\psi\right\rangle = \alpha\left|0\right\rangle + \beta\left|1\right\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$:

$$R_z(\delta)\left|\psi\right\rangle = \alpha\left|0\right\rangle + e^{i\delta}\beta\left|1\right\rangle \tag{69}$$

Since this is a relative phase, it is observed when you measure on the Z basis.

It is important to notice that every single qubit unitary operation can be made by only Phase Shift and Hadamard Gates: For instance:

$$R_z(\frac{\pi}{2} + \phi)HR_z(\theta)H\left|0\right\rangle = e^{i\frac{\theta}{2}}\big(cos\frac{\theta}{2}\left|0\right\rangle + e^{i\phi}\left|1\right\rangle\big) \tag{70}$$

Let's show this:

$$
\begin{aligned}
R_z(\frac{\pi}{2}+\phi)HR_z(\theta)H\left|0\right\rangle =&R_z(\frac{\pi}{2}+\phi)HR_z(\theta)\frac{1}{\sqrt{2}}\big(\left|0\right\rangle+\left|1\right\rangle\big)\\
=&R_z(\frac{\pi}{2}+\phi)H\frac{1}{\sqrt{2}}\big(\left|0\right\rangle+e^{i\theta}\left|1\right\rangle\big)\\
=&R_z(\frac{\pi}{2}+\phi)\frac{1}{2}\big((\left|0\right\rangle+\left|1\right\rangle)+e^{i\theta}(\left|0\right\rangle-\left|1\right\rangle)\big)\\
=&R_z(\frac{\pi}{2}+\phi)e^{i\frac{\theta}{2}}\big(cos\frac{\theta}{2}\left|0\right\rangle-isin\frac{\theta}{2}\left|1\right\rangle)\big)\\
=&e^{i\frac{\theta}{2}}\big(cos\frac{\theta}{2}\left|0\right\rangle-e^{i\frac{\pi}{2}}e^{i\phi}isin\frac{\theta}{2}\left|1\right\rangle)\big)\\
=&e^{i\frac{\theta}{2}}\big(cos\frac{\theta}{2}\left|0\right\rangle+e^{i\phi}sin\frac{\theta}{2}\left|1\right\rangle)\big)
\end{aligned}
\tag{71}
$$

The most general class of 1 qubit unitary transformation are the rotations of the Bloch Sphere. Consider an operator $\mathcal{O}$ such that $\mathcal{O}^2=\mathbb{1}$ and the Taylor expansion of the following operator:

$$e^{-i\alpha\mathcal{O}}=\left[1-\frac{1}{2!}\alpha^2+\dots\right]\mathbb{1}-i\left[\alpha-\frac{1}{3!}\alpha^3+\dots\right]\mathcal{O}=cos\ \alpha\ \mathbb{1}-isin\ \alpha\ \mathcal{O} \tag{72}$$

So if you want to rotate counter clockwise about the Z direction, we use the Pauli Z matrix:

$$e^{-i\frac{\delta}{2}Z}=cos\ \frac{\delta}{2}\ \mathbb{1}-isin\ \frac{\delta}{2}\ Z=e^{-i\frac{\delta}{2}}\begin{pmatrix}1&0\\0&e^{i\delta}\end{pmatrix}\equiv R_z(\delta) \tag{73}$$

This is the same definition given above for the phase shift gate with a global phase that can be ignored because it is of no physical significance.

If you want to rotate counter clockwise about the X direction, we use the Pauli X matrix:

$$e^{-i\frac{\delta}{2}X}\equiv R_x(\delta) \tag{74}$$

If you want to rotate counter clockwise about the Y direction, we use the Pauli Y matrix:

$$e^{-i\frac{\delta}{2}Y}\equiv R_y(\delta) \tag{75}$$

A rotation counter clockwise about an arbitrary direction can be done combining rotations about X,Y and Z axis:

$$R_n(\epsilon)\approx R_x(n_x\epsilon)R_y(n_y\epsilon)R_z(n_z\epsilon) \tag{76}$$

The taylor expansion gives:

$$R_n(\epsilon)\approx\mathbb{1}-i\frac{\epsilon}{2}(\mathbf{n}\cdot\sigma) \tag{77}$$

Then we have that:

$$R_n(\delta) = cos\,\frac{\delta}{2}\,\mathbb{1} - isin\,\frac{\delta}{2}\,(\mathbf{n}\cdot\sigma) \tag{78}$$

We can see that the Hadamard gate written in terms of rotations is written with $\delta = \pi$ and $\tilde{n} = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$:

$$H = \frac{1}{\sqrt{2}}(Z + X) \tag{79}$$

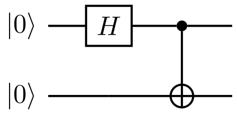
This transformation rotates the X-axis to Z and vice versa.

2.1.2 Two Qubit Gates

The most important two qubit gate is the Controled-NOT(C-NOT) gate. The *C-NOT* is a generalization of a XOR classic gate:

$$CNOT\,|A\rangle\,|B\rangle = |A\rangle\,|B \oplus A\rangle \tag{80}$$

This gate is responsible for entanglement in Circuit Quantum Computing. It is easy to show that using this gate we can construct an entangled state given by the following circuit:



In the first part we have:

$$|\psi_0\rangle = |00\rangle \tag{81}$$

After the Hadamard gate on the first qubit:

$$|\psi_1\rangle = H \otimes \mathbb{1}\,|\psi_0\rangle = \frac{1}{\sqrt{2}}\big(\,|0\rangle + |1\rangle\,\big) \otimes |0\rangle \tag{82}$$

Applying the CNOT gate:

$$|\psi_2\rangle = \text{CNOT}\,|\psi_1\rangle = \frac{1}{\sqrt{2}}\big(\,|0\rangle \otimes |0 \oplus 0\rangle + |1\rangle \otimes |0 \oplus 1\rangle\,\big) = \frac{1}{\sqrt{2}}\big(\,|00\rangle + |11\rangle\,\big) \tag{83}$$

So, using the CNOT gate we have just created an entangled state also known as one of the bell states.

2.1.3 Universality of Quantum Gates

The goal is to choose from a finite set of gates so that, by constructing a circuit choosing from this set only we can implement non-trivial and interesting (quantum) computations.

When we use a circuit of quantum gates to implement some desired unitary, it suffices to have an implementation that approximates the desired unitary to some specific level of accuracy. Suppose we approximate a desired unitary U by a unitary V. The error is defined to be:

$$E(U,V) = \max_{|\phi\rangle} ||(U-V)\,|\phi\rangle\,|| \tag{84}$$

Where $||\,|\phi\rangle\,|| = \sqrt{\langle\phi|\phi\rangle}$.

Theorem 2.1. $E(U_2U_1, V_2V_1) \leq E(U_2, V_2) + E(U_1, V_1)$

Proof.

$$\begin{aligned} E(U_2U_1, V_2V_1) &= ||(U_2U_1 - V_2V_1)\,|\phi\rangle\,|| \\ &= ||(U_2U_1 - V_2U_1)\,|\phi\rangle + (V_2U_1 - V_2V_1)\,|\phi\rangle\,|| \end{aligned} \tag{85}$$

Using that $||\,|a\rangle + |b\rangle\,|| \leq ||\,|a\rangle\,|| + ||\,|b\rangle\,||$:

$$\begin{aligned} E(U_2U_1, V_2V_1) &\leq ||(U_2U_1 - V_2U_1)\,|\phi\rangle\,|| + ||(V_2U_1 - V_2V_1)\,|\phi\rangle\,|| \\ &= E(U_2, V_2) + E(U_1, V_1) \end{aligned} \tag{86}$$

□ □

2.2 Quantum Simulation

We are concerned to find the solution of the Schrodinger Equation : $i\frac{d|\psi\rangle}{dt} = H|\psi\rangle$, which for a time-independent H the solution is:

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle \tag{87}$$

H is extremely hard to exponentiate, thus we need to find an approximate solution. A good start is to use a first-order approximation $|\psi(t+\Delta t\rangle \approx (I - iH\Delta t)|\psi(0)\rangle$, which is easier to do than exponentiating, therefore we can approximate $(I - iH\Delta t)$ using quantum gates. However, this approximation is not always satisfactory.

Efficient approximations can be obtained for many classes of Hamiltonians up to high order. Many quantum systems can be written as interactions of local Hamiltonians. For a system of n particles, $H = \sum_{k=1}^{L} H_k$ where each H_k acts on at most a constant number of particles, this restriction leads that $L = poly(n)$. Often H_k are one-body interactions such as X_j or two-body interactions such as X_jX_l.

For this kind of Hamiltonian we have that e^{-iHt} is hard to compute, as usual, but e^{-iH_kt} is rather easy since it acts on a smaller sub-system. There is only one problem: In general $e^{-iHt} \neq \prod_k e^{-iH_jt}$ when $[H_j, H_k] \neq 0$.

Let's show that if $[H_j, H_k] = 0$, then $e^{-iHt} = \prod_k e^{-iH_kt}$:

$$e^{-iHt} = e^{-i\sum_k H_kt}$$

Using the Zassenhaus formula, we have that $e^{t(A+B)} = e^{tA}e^{tB}\prod_{n=2}^{\infty} e^{t^n Z_n(A,B)}$,
where $Z_n(A,B) = \frac{1}{n!}\left[\frac{d}{dt^n}\left(e^{-t^{n-1}Z_{n-1}}\dots e^{t^2Z_2(A,B)}e^{-tB}e^{-tA}e^{t(A+B)}\right)\right]_{t=0}$.
Simplifying: $e^{t(A+B)} = e^{tA}e^{tB}e^{-\frac{t^2}{2}[A,B]}e^{\frac{t^3}{3!}(2[B,[A,B]]+[A,[A,B]])}\dots$. Thus we have that:

$$\begin{aligned} e^{-iHt} &= e^{-it(H_1+\sum_{k=2}^{L} H_k} = e^{-itH_1}e^{-it\sum_k H_k}e^{-\frac{t^2}{2}[H_1,\sum_k H_k]}\dots \\ &= e^{-itH_1}e^{-it\sum_k H_k} = \dots = \prod_{k=1}^{L} e^{-itH_k} \end{aligned}$$

We can also show that the restriction of H_k to involve at most a constant number of particles, c, implies that we have a polynomial amount of terms, precisely L is upper-bounded by a polynomial in n.

$$\begin{aligned} \#\ \text{of terms} &\leq \#\ \text{of cosets of n that has the size at most c} \\ L &\leq \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{c} \\ &\leq c\binom{n}{c} \leq \frac{n^c}{(c-1)!} \\ &\Rightarrow L \leq poly(c) \end{aligned}$$

The main algorithm for quantum simulation uses the Trotter Formula:

Theorem 2.2 (Trotter Formula)**.** *Let A, and B be Hermitian operators. Then for any real t:*

$$\lim_{n\to\infty}(e^{\frac{iAt}{n}}e^{\frac{iBt}{n}})^n = e^{i(A+B)t}$$

Proof. By definition: $e^{\frac{iAt}{n}} = I + \frac{1}{n}iAt + O\big(\frac{1}{n^2}\big)$, and thus:

$$e^{\frac{iAt}{n}}e^{\frac{iBt}{n}} = I + \frac{1}{n}i(A+B)t + O\bigg(\frac{1}{n^2}\bigg)$$

Thus:

$$(e^{\frac{iAt}{n}}e^{\frac{iBt}{n}})^n = I + \sum_{k=1}^{n}\binom{n}{k}\frac{1}{n^k}\big[i(A+B)t\big]^k + O\bigg(\frac{1}{n}\bigg)$$

Since $\binom{n}{k}\frac{1}{n^k} = \frac{1+O(\frac{1}{n})}{k!}$, this gives:

$$\lim_{n\to\infty}(e^{\frac{iAt}{n}}e^{\frac{iBt}{n}})^n = \lim_{n\to\infty}\sum_{k=0}^{n}\frac{(i(A+B)t)^k}{k!}\bigg(1+O\bigg(\frac{1}{n}\bigg)\bigg) + O\bigg(\frac{1}{n}\bigg) = e^{i(A+B)t}$$

□

We can also prove the following statements:

$$e^{i(A+B)\Delta t} = e^{iA\Delta t}e^{iB\Delta t} + O(\Delta t^2) \qquad e^{i(A+B)\Delta t} = e^{iA\Delta t/2}e^{iB\Delta t}e^{iA\Delta t/2} + O(\Delta t^3)$$

Using this formula, we can construct the Quantum Simulation Algorithm [14]:

Algorithm 1: Hamiltonian Simulation Algorithm

```
input  : (1) A Hamiltonian $H = \sum_k H_k$.
(2) An initial state $|\psi_0\rangle$ of the system at $t = 0$.
(3) A time $t_f$ to end the simulation and a desired accuracy $\delta$
output : A state $\left|\tilde{\psi}(t_f)\right\rangle$ such that $|\left\langle\tilde{\psi}(t_f)\right| e^{-iHt_f}|\psi_0\rangle|^2 \geq 1-\delta$
runtime: $O(poly(\frac{1}{\delta}))$ operations.
Choose a representation such that the state $\left|\tilde{\psi}\right\rangle$ of $n = poly(\log N)$ qubits approximate the system and the
  operators $e^{-iH_k\Delta t}$ have efficient Quantum Circuits approximations.
Select an approximation method (usually Trotter) and $\Delta t$ such that the error is acceptable ( $j\Delta t = t_f$ , $j \in \mathbb{Z}^+$ )
  and construct the corresponding circuit $U_{\Delta t}$ for the following iterative step:
  1. $\left|\tilde{\psi_0}\right\rangle \leftarrow |\psi_0\rangle$ , $j = 0$
  2. $\to \left|\tilde{\psi}_{j+1}\right\rangle = U_{\Delta t}\left|\tilde{\psi}_j\right\rangle$
  3. $\to j+=1$; goto 2 until $j\Delta t \geq t_f$
  4. $\to \left|\tilde{\psi}(t_f)\right\rangle = \left|\tilde{\psi}_j\right\rangle$
```

Example: Suppose that we have the following Hamiltonian:

$$H = Z_1 \otimes Z_2 \otimes \cdots \otimes Z_n$$

Which acts on a n qubit system. How can we simulate $e^{-iH\Delta t}$ efficiently?

First, let's suppose that $n = 2$ and see the action of the $Z \otimes Z = ZZ$ operator in the computational basis:

$$ZZ\,|00\rangle = |00\rangle \qquad ZZ\,|10\rangle = -\,|10\rangle$$
$$ZZ\,|01\rangle = -\,|01\rangle \qquad ZZ\,|11\rangle = |11\rangle$$

Thus the exponentiation of ZZ can be computed using the parity and one rotation along the Z axis:

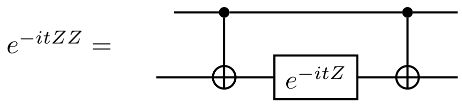


Figure 4: Representation of ZZ into a Quantum Circuit.

To calculate XX or YY, we can use that $X = HZX$ and $Y = (HS^\dagger)^\dagger Z(HS^\dagger)$ in order to make a change of basis:

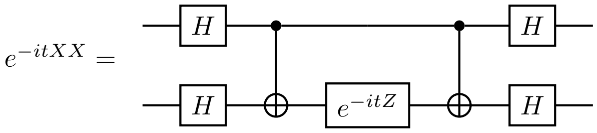


Figure 5: Representation of XX into a Quantum Circuit.

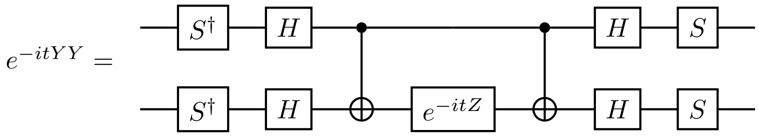


Figure 6: Representation of YY into a Quantum Circuit.

This can be extended for an arbitrary number of qubits! Thus, we apply the phase-shift $e^{-i\Delta t}$ if the parity of the n qubits in the computational basis is even; otherwise, the phase-shift should be $e^{i\Delta t}$. Therefore, we can efficiently simulate this kind of Hamiltonian into a Quantum Circuit by storing the parity in a auxiliary qubit and apply the phase-shift conditioned on the parity, then uncomputing the parity (to erase the auxiliary qubit). Let's show one example:

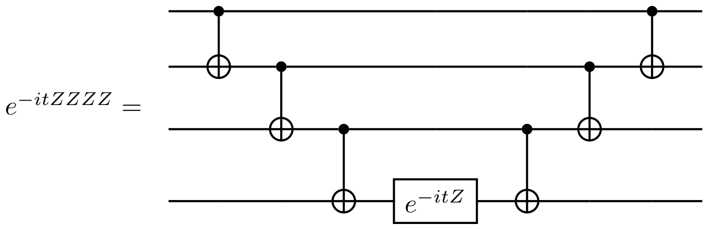


Figure 7: Representation of $ZZZZ$ into a Quantum Circuit.

Using this kind of circuit, we can efficiently simulate any Hamiltonian of the form:

$$H = \bigotimes_{k=1}^{n} \sigma_{c(k)}^{k}$$

Where $\sigma_{c(k)}^{k}$ is a Pauli matrix acting on the k-th qubit such that $c(k) \in \{I, X, Y, Z\}$.

2.3 Deutsch's Algorithm

Problem: Consider an oracle evaluating a 1 bit boolean function $f : \{0,1\} \to \{0,1\}$ we want to know if the function is constant ($f(0) = 0$ and $f(1) = 1$) or balanced ($f(0) = 1$ and $f(1) = 0$).

For a classical computer it is needed two queries, that means, you need to test 2 diferent bits and see the outputs, for instance if $f(0) = 0$ and $f(1) = 1$ you know that the function is constant.

For a quantum computer this can be done with only one query. This is done with the Deutsch algorithm, the idea is to check if it is balanced, if it is not balanced it is constant.

Solution: The Deutsch's Algorithm is the following:

1. Start with the state $|\psi\rangle = |10\rangle$:

2. Apply a Hadamard on both qubits, the state will be: $|\psi_2\rangle = \frac{1}{2}(|0\rangle - |1\rangle) \otimes (|0\rangle + |1\rangle)$

3. Apply a Unitary operator such that: $U_f \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|x\rangle = (-1)^{f(x)} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|x\rangle$. The phase factor is "kicked back" on the front of the state, this will be useful to evaluate $f(x)$ with only one query. The state after U_f is: $|\psi_3\rangle = \frac{1}{2}(|0\rangle - |1\rangle) \otimes \left((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right)$

4. Apply a Hadamard on the first Qubit: $|\psi_3\rangle = \frac{1}{2}(|0\rangle - |1\rangle) \otimes \left[\left((-1)^{f(0)} + (-1)^{f(1)}\right)|0\rangle + \left((-1)^{f(0)} - (-1)^{f(1)}\right)|1\rangle\right]$

So now if $f(0) = f(1)$ we will measure $|0\rangle = |f(0) \bigoplus f(1)\rangle$ and if $f(0) \neq f(1)$ we will measure $|1\rangle = |f(0) \bigoplus f(1)\rangle$.

In this implementation the oracle (U_f) is a C-NOT gate, this will measure if the function is balanced.

This can be extended for a function with more inputs, not only two. In this case, the solution is the Deutsch–Jozsa Algorithm.

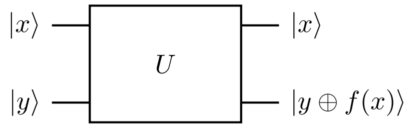


Figure 8: Circuit for the Oracle used on Deutsch's Algorithm.

2.4 Deutsch-Josza Algorithm

In the Deutsch's Algorithm we had only 1 bit boolean function, for the Deutsch-Josza case, we have a N bit boolean function $f : \{0,1\}^N \rightarrow \{0,1\}$, where f is constant or balanced:

- f is constant if $\forall x \in \{0,1\}^N$, $f(x) = b \in \{0,1\}$
- f is balanced if $f(x) = b$ for half of the inputs and $f(x) = b \oplus 1$ for the other half.

Classicaly, for a deterministic algorithm, it is needed $\frac{2^N}{2} + 1 = 2^{N-1} + 1$ queries.

The Deutsch-Josza Algorithm is represented by the following circuit:

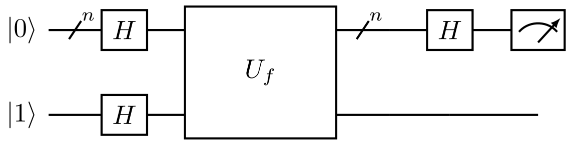
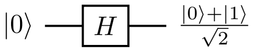


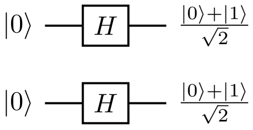
Figure 9: Deutsch-Josza Algorithm, the slash on the circuit represent N qubits.

Observe that:

For 1 qubit:



For 2 qubits:


$|0\rangle$
H
$\frac{|0\rangle+|1\rangle}{\sqrt{2}}$
$|0\rangle$
H
$\frac{|0\rangle+|1\rangle}{\sqrt{2}}$

Therefore, for N qubits we have a combination of all possible strings:

$$|\psi\rangle = \frac{1}{\sqrt{2^N}} \sum_{x=0}^{2^N-1} |x\rangle \tag{88}$$

So in the first step of the algorithm, we have:

$$|\psi_0\rangle = |0\rangle^{\otimes N} |1\rangle \tag{89}$$

After the Hadamard gates:

$$|\psi_1\rangle = H^{\otimes N+1} |\psi_0\rangle = \frac{1}{2^{N/2}} \sum_{x=0}^{2^N-1} |x\rangle \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \tag{90}$$

In the next step we use the oracle as on the Deutsch's Algorithm, since the oracle must be linear, we only need to check one term of the sum:

$$U_f |x\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) = \frac{|x\rangle |0 \oplus f(x)\rangle - |x\rangle |1 \oplus f(x)\rangle}{\sqrt{2}} \tag{91}$$

We have two cases:

- If $f(x) = 0$, then:

$$U_f \left|x\right\rangle \left(\frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}\right) = \left|x\right\rangle \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) \tag{92}$$

- If $f(x) = 1$, then:

$$U_f \left|x\right\rangle \left(\frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}\right) = \left|x\right\rangle \left(\frac{-\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}\right) = -\left|x\right\rangle \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) \tag{93}$$

We can represent both cases by:

$$U_f \left|x\right\rangle \left(\frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}\right) = (-1)^{f(x)} \left|x\right\rangle \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) \tag{94}$$

Therefore, after the Oracle:

$$\left|\psi_2\right\rangle = U_f \left|\psi_1\right\rangle = \frac{1}{2^{N/2}} \sum_{x=0}^{2^N-1} (-1)^{f(x)} \left|x\right\rangle \otimes \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) \tag{95}$$

Before the next step, observe that:

$$\begin{aligned} H\left|0\right\rangle &= \frac{(-1)^{0\cdot 0}\left|0\right\rangle + (-1)^{0\cdot 1}\left|0\right\rangle}{\sqrt{2}} \\ H\left|1\right\rangle &= \frac{(-1)^{1\cdot 0}\left|0\right\rangle + (-1)^{1\cdot 1}\left|0\right\rangle}{\sqrt{2}} \end{aligned} \tag{96}$$

Therefore:

$$H\left|b\right\rangle = \sum_{b'=0}^{1} \frac{(-1)^{b\odot b'}\left|b'\right\rangle}{\sqrt{2}} \tag{97}$$

Where $\odot$ is the bitwise sum.

Now we can apply the Haddamard gate to all N qubits:

$$\begin{aligned} \left|\psi_3\right\rangle =& H^{\otimes N} \otimes \mathbb{1} \left|\psi_2\right\rangle = \frac{1}{2^{N/2}} \sum_{x_1,\dots,x_N=0}^{1} (-1)^{f(x)} H\left|x_1\right\rangle \otimes H\left|x_2\right\rangle \otimes \cdots \otimes H\left|x_N\right\rangle \otimes \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) \\ =& \frac{1}{2^{N/2}} \sum_{x=0}^{2^N-1} (-1)^{f(x)} \left(\sum_{z_1=0}^{1} \frac{(-1)^{z_1\cdot x_1}\left|z_1\right\rangle}{\sqrt{2}}\right) \otimes \left(\sum_{z_2=0}^{1} \frac{(-1)^{z_2\cdot x_2}\left|z_2\right\rangle}{\sqrt{2}}\right) \otimes \cdots \otimes \left(\sum_{z_N=0}^{1} \frac{(-1)^{z_N\cdot x_N}\left|z_N\right\rangle}{\sqrt{2}}\right) \otimes \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) \\ =& \frac{1}{2^N} \sum_{x=0}^{2^N-1} (-1)^{f(x)} \sum_{z=0}^{2^N-1} (-1)^{z_1\cdot x_1+\cdots+z_N\cdot x_N} \left|z_1 \dots z_N\right\rangle \otimes \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) \end{aligned} \tag{98}$$

Therefore:

$$\left|\psi_3\right\rangle = \frac{1}{2^N} \sum_{x=0}^{2^N-1} \sum_{z=0}^{2^N-1} (-1)^{f(x)} (-1)^{z\odot x} \left|z\right\rangle \otimes \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) \tag{99}$$

The probability to measure $|00\dots00\rangle$ is given by:

$$Pr(00\dots00) = \left|\frac{1}{2^N}\sum_{x=0}^{2^N-1}(-1)^{f(x)}\right|^2 \tag{100}$$

We have two cases as before:

- If f is constant:

$$\sum_{x=0}^{2^N-1}(-1)^{f(x)} = (-1)^{f(x)}2^N \tag{101}$$

  Therefore: $Pr(00\dots00) = 1$

- If f is balanced:

$$\sum_{x=0}^{2^N-1}(-1)^{f(x)} = \frac{1}{2}\sum_{x=0}^{(2^N-1)/2} 1 - \frac{1}{2}\sum_{x=0}^{(2^N-1)/2} 1 = 0 \tag{102}$$

  Therefore: $Pr(00\dots00) = 0$

So we know that the function is constant if we measure $|00\dots00\rangle$, any other result the function is balanced. Since with only one query we can know if the function is constant or balanced, we have an exponential gain compared with the classical deterministic case.

2.5 Grover's Algorithm

Grover's algorithm is useful for searching an unstructured database with N elements. For instance, you have a phone number and want to find the corresponding name associated to this phone number, since it is unstructured you need to check every element (in the worst case scenario), but if you have the solution it is easy to check, this shows that the problem is NP.

In order to show Grover's algorithm we need to rephrase as an oracle problem: Labelling each element of the database $\{0, 1, \ldots, N-1\}$ and x_0 the unknown marked item. The oracle f computes the following binary function:

$$f : \{0,1\}^N \to \{0,1\} \quad , \quad \text{with} \qquad f(x) = \begin{cases} 1, & \text{if } x = x_0 \\ 0, & \text{otherwise} \end{cases} \tag{103}$$

For a classical computer, the probability to find x_0 is $\frac{1}{N}$, so to find x_0 with probability p is needed $pN = \mathcal{O}(N)$ oracle queries. Grovers showed that on a quantum computer we can have a quadratic speedup, then we will need $\mathcal{O}(\sqrt{N})$ queries. This is not massive, but we can compare this speedup with the breakthrough that Fast Fourier Transform(FFT) did for signalling processing.

2.5.1 General Algorithm

Here we suppose a quantum oracle with the ability to recognize solutions to the search problem. This recognition is signaled by making use of an oracle qubit. More precisely:

$$|x\rangle\,|q\rangle \to_O |x\rangle\,|q \oplus f(x)\rangle \tag{104}$$

The oracle $|q\rangle$ is a single qubit which is flipped if $f(x) = 1$ and is unchanged otherwise. We can check if whether x is a solution by preparing $|x\rangle\,|0\rangle$ and seeing if the oracle qubit flips when using the oracle.

It is common to use $|q\rangle = \frac{1}{\sqrt{2}}\big(\,|0\rangle - |1\rangle\,\big)$. The action of the oracle is:

$$|x\rangle\,\frac{1}{\sqrt{2}}\big(\,|0\rangle - |1\rangle\,\big) \to_O (-1)^{f(x)}\,|x\rangle\,\frac{1}{\sqrt{2}}\big(\,|0\rangle - |1\rangle\,\big) \tag{105}$$

Using this type of oracle qubit it remains the same when applying the oracle and can be omitted from the following steps of the algorithm. With this convention:

$$|x\rangle \underset{O}{\longrightarrow} (-1)^{f(x)}\,|x\rangle \tag{106}$$

The oracle marks the solution by shifting the phase. For a N item search problem with M solutions, we need to apply the search oracle $O(\sqrt{\frac{N}{M}})$ times in order to obtain the solution.

The algorithm with the Grover operator G is the following:

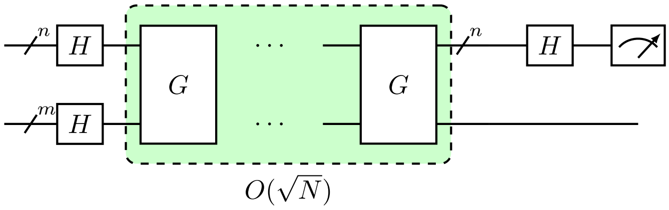


Figure 10: Circuit for the Grover Algorithm, the unitary G are called Grover Iterations or Grover Operator.

The algorithm begins with $|0\rangle^{\otimes N}$. The Haddamard transform put in the equal superposition state, which we call $|\psi\rangle$:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{107}$$

The algorithm consists of repeated applications of the grover operator G. This subroutine can be broken up into 4 steps:

(1) Apply the oracle O.

(2) Apply the Haddamard transform $H^{\otimes N}$.

(3) Perform a conditional phase shift on the computer, with every computational basis state except $|0\rangle$ receiving a phase shift of -1: $|x\rangle \xrightarrow{M} -(-1)^{\delta_{x0}} |x\rangle$.

(4) Apply the Haddamard transform $H^{\otimes N}$.

The circuit for the Grover operator is the following:

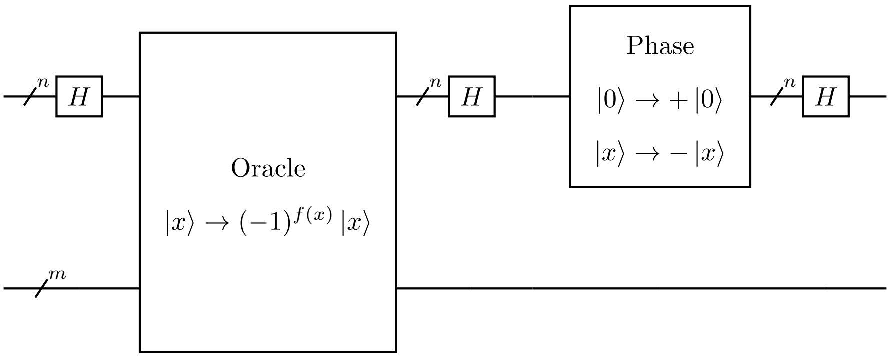

n
H
Oracle
$|x\rangle \to (-1)^{f(x)} |x\rangle$
m
n
H
Phase
$|0\rangle \to +|0\rangle$
$|x\rangle \to -|x\rangle$
n
H

Figure 11: Grover Operator.

This operator for the conditional phase shift on step (2) is simply $M = 2|0\rangle\langle 0| - \mathbb{I}$, because we have:

$$M = \begin{pmatrix} 2-1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & -1 \end{pmatrix} \tag{108}$$

Steps 2 and 4 needs $n = \log(N)$ operations each. Step 3, the conditional phase shift, can be implemented using $O(n)$ gates. But the cost of the Oracle depends on the application, on the bright side, we only need to call the oracle once. The steps 2,3,4 combined are:

$$H^{\otimes n}\big(2\,|0\rangle\langle 0| - \mathbb{I}\big)H^{\otimes n} = 2\,|\psi\rangle\langle\psi| \tag{109}$$

Where $|\psi\rangle$ is the weighted superposition. Thus: $G = \big(2\,|\psi\rangle\langle\psi| - \mathbb{I}\big)O$. Let's see how the operator $2\,|\psi\rangle\langle\psi| - \mathbb{I}$ acts on an arbitrary state: $|\eta\rangle = \sum_k \alpha_k\,|k\rangle$:

$$\begin{aligned}\Big(2\,|\psi\rangle\langle\psi| - \mathbb{I}\Big)\,|\eta\rangle &= \Big(2\,|\psi\rangle\langle\psi| - \mathbb{I}\Big)\sum_k \alpha_k\,|k\rangle \\ &= \left[\frac{2}{N}\sum_{x,y}|x\rangle\langle y| - \mathbb{I}\right]\sum_k \alpha_k\,|k\rangle = \frac{2}{N}\sum_{x,y,k}\alpha_k\,|x\rangle\langle y|\,|k\rangle - \sum_k \alpha_k\,|k\rangle \\ &= \frac{2}{N}\sum_{x,k}\alpha_k\,|x\rangle - |\eta\rangle = 2\overbrace{\sum_k \frac{\alpha_k}{N}}^{\equiv\langle\alpha\rangle}\sum_x |x\rangle - |\eta\rangle \\ &= 2\sum_x \langle\alpha\rangle\,|x\rangle - \sum_k \alpha_k\,|k\rangle = \sum_k\Big[-\alpha_k + 2\langle\alpha\rangle\Big]\,|k\rangle\end{aligned} \tag{110}$$

For this reason, sometimes this operator is called the inversion about the mean.

We can have a geometric visualization of the algorithm that will help us to show that the algorithm needs $O(\sqrt{N})$ gates, for this we want to show that the Grover Operator can be regarded as a rotation in the 2D space spanned by the starting vector $|\psi\rangle$ and the state with the uniform superposition of solutions to the search problem. We adopt the convention$\sum_x^{'}$ to be the sum over all x which are solutions to the search problem, and $\sum_x^{''}$ to be the sum over all x which are not solutions. We also define the normalized states:

$$|\alpha\rangle \equiv \frac{1}{\sqrt{N-M}}\sum_x^{''}|x\rangle \qquad |\beta\rangle \equiv \frac{1}{\sqrt{M}}\sum_x^{'}|x\rangle \tag{111}$$

We can rewrite $|\psi\rangle$ in terms of $|\alpha\rangle$ and $|\beta\rangle$:

$$\begin{aligned}|\psi\rangle &= \frac{1}{\sqrt{N}}\sum_x |x\rangle = \frac{1}{\sqrt{N}}\left(\sum_x^{'}|x\rangle + \sum_x^{''}|x\rangle\right) \\ &= \frac{1}{\sqrt{N}}\Big(\sqrt{M}\,|\beta\rangle + \sqrt{N-M}\,|\alpha\rangle\Big) \\ &= \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{M}{N}}\,|\beta\rangle\end{aligned} \tag{112}$$

So that the initial state is spanned by $|\alpha\rangle$ and $|\beta\rangle$. The effect of G can be understood in a beautiful way by showing that the oracle O performs a reflection about the vector $|\alpha\rangle$ and $|\beta\rangle$. That is:

$$O(a\,|\alpha\rangle + b\,|\beta\rangle) = a\,|\alpha\rangle - b\,|\beta\rangle \tag{113}$$

Similarly, $2\,|\psi\rangle\langle\psi| - I$ also performs a reflection in the plane about the vector $|\psi\rangle$. The product of two reflections is a rotation!

Thus $G^k |\psi\rangle$ remains in the space spanned by $|\alpha\rangle$ and $|\beta\rangle$. Let $\cos\left(\frac{\theta}{2}\right) = \sqrt{\frac{N-M}{N}}$, so that $|\psi\rangle = \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle$. Let's see what happens on a grover iteration:

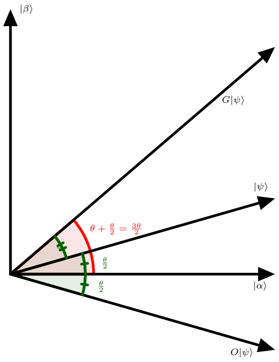


Figure 12: Geometric picture of the Grover Iteration.

We see from the figure that:

$$G|\psi\rangle = \cos\frac{3\theta}{2}|\alpha\rangle + \sin\frac{3\theta}{2}|\beta\rangle \tag{114}$$

So the rotation is equal to θ. It follows that after k iterations of G, we have:

$$G^k|\psi\rangle = \cos\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\beta\rangle \tag{115}$$

Thus, in the $\{|\alpha\rangle, |\beta\rangle\}$ basis, we can write the Grover iteration as:

$$G = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{116}$$

Where $\theta \in [0, \frac{\pi}{2}]$ and $\sin\theta = \sin\left(\frac{\theta}{2} + \frac{\theta}{2}\right) = 2\sin\frac{\theta}{2}\cos\frac{\theta}{2} \Rightarrow \sin\theta = 2\frac{\sqrt{M(N-M)}}{N}$

In conclusion, the grover iteration acts as a rotation towards $|\beta\rangle$ in the space spanned by $\{|\alpha\rangle, |\beta\rangle\}$, so repeated applications of G are required for us to measure $|\beta\rangle$ with high probability.

2.5.2 Performance

How many times we need to repeat the Grover operator in order to rotate $|\psi\rangle$ near $|\beta\rangle$?

The initial system starts with $|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle$, so rotating through $\arccos\sqrt{\frac{M}{N}}$ radians takes the system to $|\beta\rangle$. Let denote the closest integer to the real number x as $\text{CI}(x)$, where we round halves down. Then repeating the grover iteration $R = \text{CI}\left(\frac{\arccos\sqrt{\frac{M}{N}}}{\theta}\right)$ times rotates $|\psi\rangle$ within an angle $\frac{\theta}{2} \leq \frac{\pi}{4}$ of $|\beta\rangle$.

Observation of the state in the computational basis then yields a solution to the search problem with a probability at least $\frac{1}{2}$. For specific values of M and N it is possible to achieve higher probability of success. For example, if $M << N$, we have that $\sin\theta \approx \theta$, and thus the angular error in the final state is at most $\frac{\theta}{2} \approx \sqrt{\frac{M}{N}}$, giving probability of at most $\frac{M}{N}$.

The form of R is an exact expression for the query complexity of the search algorithm, but we can have a simpler expression that summarizes the essential behaviour of R. Note that $R \leq \lceil \frac{\pi}{2\theta} \rceil$, because $\arccos\sqrt{\frac{M}{N} \leq \frac{\pi}{2}}$, so we can give an upper bound on R. Firstly we assume $M \leq \frac{N}{2}$, thus $\frac{\theta}{2} \geq \sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$. Then we have a upper bound for R:

$$R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil \tag{117}$$

Thus, $R = O\big(\sqrt{\frac{N}{M}}\big)$ oracle calls must be performed in order to obtain a solution with high probability. We have a polynomial speed-up over the classical algorithm.

We only considered the case when $M < \frac{N}{2}$, what happens if this is not the case? In this case the angle θ gets smaller and we need more interactions as M increases. But since we expect to be easy to find a solution (because $M \geq \frac{N}{2}$) we can randomly pick a solution and check if it is a solution using the oracle. This succeeds with probability at least $\frac{1}{2}$, and requires only one query of the oracle, but only works if we know the number of solutions in advance.

If we don't know that $M \geq \frac{N}{2}$. We can double the number of elements in the search space by adding N extra items, none which are solutions. This is done by adding a single qubit $|q\rangle$ to the search index, doubling the numbers of items to be searched to $2N$. A new augmented oracle O' can be constructed in a way that marks an item only if it is a solution to the search problem and the extra bit is set to zero; The new search problem has only M solutions out of $2N$, so using $R = \frac{\pi}{4}\sqrt{\frac{2N}{M}}$ queries of O' we get a solution of high probability, since we are using the Big-O notation we are ignoring constants that multiply the scaling and we have the same scaling $O\big(\sqrt{\frac{N}{M}}\big)$.

2.5.3 Searching for 1 Item on N=4 elements

To understand the more general algorithm we start with an example:

The circuit for this example is on [5]:

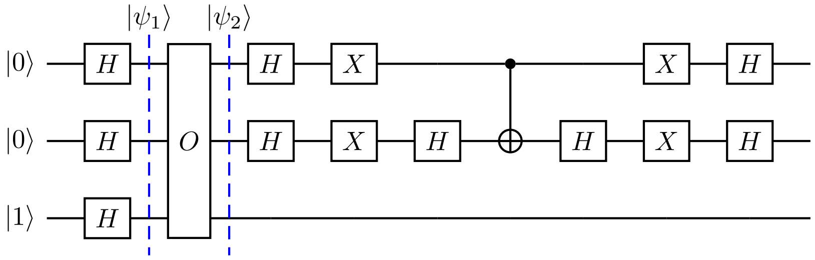


Figure 13: Grover Algorithm for N=4.

So we start with two qubits in the state $|00\rangle$ and an ancillary state $|1\rangle$:

$$|\psi_0\rangle = |00\rangle\,|1\rangle \tag{118}$$

Then we apply in all qubits Hadamard gates in order to have all possible bit strings:

$$|\psi_1\rangle = H^{\otimes 3}|\psi_0\rangle = \frac{1}{2}\big(|00\rangle + |01\rangle + |10\rangle + |11\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \tag{119}$$

The oracle query works to mark down the searched state:

$$O|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle \tag{120}$$

Since the ancillary state is $\frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big)$ it is the same if $f(x) = 0$ but changes sign if $f(x) = 1$. Suppose the item that we are searching is $x_0 = (1, 0) \rightarrow |10\rangle$, after the oracle query we have:

$$|\psi_2\rangle = O|\psi_1\rangle = \frac{1}{2}\big(|00\rangle + |01\rangle - |10\rangle + |11\rangle\big)\frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \tag{121}$$

This is the same process of kicking back the sign that we have on the Deutsch's Algorithm. Since the ancillary state is the same, we will ignore it on the calculation.

Since in Quantum Mechanics the probability of measuring a state is the absolute value squared we cannot distinguish between the four state superposition. The first part of the algorithm is to mark the state that we are interested and the last part is to amplify its probability to be measured.

This is done by means of the unitary transformation:

$$D_{ij} = -\delta_{ij} + \frac{2}{2^N} \tag{122}$$

For $N = 2$, we have that:

$$D = \frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix} \tag{123}$$

It is usually not easy to find the circuit representation of unitary transformations, but after some work and some experience we can decompose D:

$$D = H^{\otimes 2}\ D'\ H^{\otimes 2} \tag{124}$$

Where D' is:

$$D' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{125}$$

This is a controlled phase shift through an angle π in the coefficient in front of the basis element $|00\rangle$. Again we need to decompose D':

$$D' = X^{\otimes 2}\ (I \otimes H)\ CNOT\ (I \otimes H)\ X^{\otimes 2} \tag{126}$$

This comes from the $CMINUS$ gate, that is:

$$CMINUS = (I \otimes H)\ CNOT\ (I \otimes H) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{127}$$

The NOT (X) gates places the phase factor in front of the $|00\rangle$ instead of the $|11\rangle$.

So applying D on our state $|\psi_2\rangle$, we have:

$$D\ \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{128}$$

So if we measure the two qubits, we have that the outcome will be $|10\rangle$ with 100% certainty.

For a Classical algorithm we would need on average $N_c = \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{2} \cdot 3 = 2.25$ queries. In the Quantum algorithm we would only need one query, thus $N_Q = 1$.

2.5.4 Implementation on Qiskit

Here I will follow the Qiskit Book [3] and we will focus on the implementation for $N = 4$ as the example before, but the algorithm implementation is not hard to scale up for any N.

The first part of the algorithm is to build an oracle that marks the desired state, let's consider the marked up state $|11\rangle$.

Therefore, we need that the oracle acts as follows:

$$O\,|s\rangle = O\ \frac{1}{2}\ \Big(|00\rangle + |01\rangle + |10\rangle + |11\rangle \Big) = \frac{1}{2}\ \Big(|00\rangle + |01\rangle + |10\rangle - |11\rangle \Big) \tag{129}$$

This is the same as a controled Z gate, that means if your first qubit is $|1\rangle$, then you apply an Z operator on the seccond qubit:

$$\mathrm{CZ} = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes Z \tag{130}$$

Therefore, the circuit for this part is:

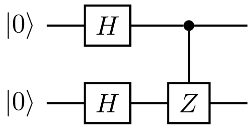


Figure 14: Grover implementation on Qiskit, step 1.

The next part is the amplitude augmentation, since the only the sign is changed, there is no difference between this state and the state with all superpositions upon measurement. We need to grow the marked state probability to be measured.

This can be done by the reflection: $D = 2\left|s\right\rangle\left\langle s\right| - \mathbb{1}$. As shown on the previous section, we have that: $D = H^{2\otimes} D' H^{2\otimes}$, where:

$$D' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{131}$$

Therefore:

$$D'\left|s\right\rangle = \frac{1}{2}\left(\left|00\right\rangle - \left|01\right\rangle - \left|10\right\rangle - \left|11\right\rangle\right) \tag{132}$$

We know that the state $\left|00\right\rangle$ is the only one that changes the sign and we also know that we can change the sign of $\left|11\right\rangle$ by an CZ gate. The other signs can be changed using Z gates on each qubit, because $Z\left|i\right\rangle = (-1)^i\left|i\right\rangle$.

Therefore the circuit for the reflection is the following:

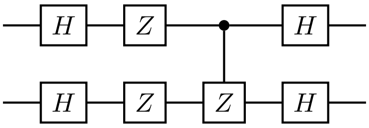


Figure 15: Grover implementation on Qiskit, step 2.

Now we have our full circuit:

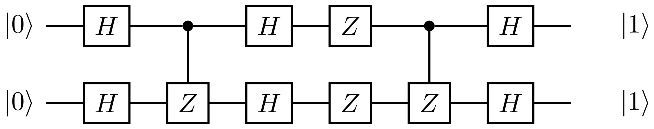


Figure 16: Grover implementation on Qiskit, full circuit.

The qiskit implementation is on the following notebook.

2.5.5 Optimality of Quantum Search

It has been shown previously that Grover's search algorithm can search N item with query complexity $O(\sqrt{N})$. We can ask a question: What would happen if we had a quantum algorithm that is $O(\text{log}N)$?

If this was possible we could solve **NP**-Complete problems efficiently using a quantum computer, which means $NP \subseteq BQP$. To see if there exists a quantum search algorithm better than Grover's algorithm we need to check the lower bound of the algorithm, therefore we would like to show that the Quantum Search Problem requires $\Omega(\sqrt{N})$ queries.

Suppose that the algorithm starts with $|\psi\rangle$ and we want to find a single solution $|x\rangle$. To determine this solution we apply the oracle $O_x = \mathbb{1} - 2\,|x\rangle\,\langle x|$ and the following circuits:

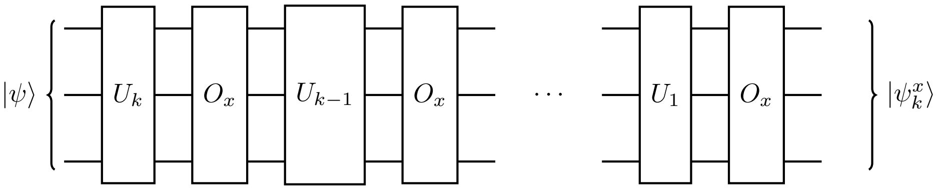


Figure 17: .

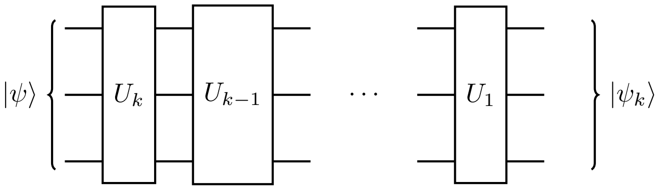


Figure 18: .

We want to bound a quantity that we call deviation (D_k):

$$\underbrace{D_k}_{\text{Deviation}} = \sum_x || \overbrace{\psi_k^x}^{|\psi_k^x\rangle} - \underbrace{\psi_k}_{|\psi_k\rangle} ||^2 \tag{133}$$

For a sanity check, we have $\psi_k^x = \psi_k \Rightarrow D_k = 0$. To show the lower bound, we want to prove two steps:

(a) We want to prove a bound on D_k which states that it can't grow faster than $O(k^2)$;

(b) Prove that D_k must be $\Omega(N)$ if it is possible to distinguish N alternatives.

We will prove statement (a) by induction:

Let's show that is valid for 0: $D_0 = \sum_x ||\psi_0^x - \psi_0||^2$, but $\psi_0^x = |\psi\rangle = \psi_0$, then: $D_0 = 0$

Suppose that it is valid for k: $D_k \leq 4k^2$

Let's prove this statement is valid for k+1:

$$D_{k+1} = \sum_x ||O_x\psi_k^x - \psi_k||^2 = \sum_x ||O_x(\psi_k^x - \psi_k) + (O_x - I)\psi_k||^2 \tag{134}$$

Using that $||b + c||^2 \leq ||b||^2 + 2||b|| \cdot ||c|| + ||c||^2$:

$$D_{k+1} \leq \sum_x \left(||\psi_k^x - \psi_k||^2 + 4||\psi_k^x - \psi_k||\; ||\,\langle x|\psi\rangle\,|| + 4|\,\langle x|\psi_k\rangle\,|^2 \right) \tag{135}$$

$$D_{k+1} \leq D_k + 4\sum_x ||\psi_k^x - \psi_k||\; ||\langle x|\psi\rangle|| + 4 \tag{136}$$

Using Cauchy-Schwarz: $\big|\sum_x u_x\bar{v}_x\big|^2 \leq \big|\sum_x u_x\big|^2\big|\sum_{x'} v_{x'}\big|^2$

$$\Big|\sum_x ||\psi_k^x - \psi_k||\;|\langle x|\psi_k\rangle|\Big|^2 \leq \Big|\overbrace{\sum_x ||\psi_k^x - \psi_k||^2}^{D_k}\Big|\;\Big|\overbrace{\sum_{x'} |\langle x'|\psi_k\rangle|^2}^{=1}\Big| \tag{137}$$

Thus:

$$\sum_x ||\psi_k^x - \psi_k||\;|\langle x|\psi_k\rangle| \leq \sqrt{D_k} \tag{138}$$

Applying eq 138 on eq 136:

$$D_{k+1} \leq D_k + 4\sqrt{D_k} + 4 \tag{139}$$

By hypothesis $D_k \leq 4k^2$, therefore:

$$D_{k+1} \leq 4k^2 + 8k + 4 = 4(k+1)^2 \tag{140}$$

Then since it is valid for $k = 0$ and assuming k is valid then $k+1$ is also valid we've shown that $D_k \leq 4k^2$ by induction.

□

We want to show that the probability of success can only be high if D_k is $\Omega(N)$. We suppose $|\langle x|\psi_k^x\rangle|^2 \geq \frac{1}{2}\ \forall x$ so that an observation is a solution to the problem with probability at least $\frac{1}{2}$. If we add a phase to $|x\rangle$ the probability of success is the same, so we may assume: $\langle x|\psi_k^x\rangle = |\langle x|\psi_k^x\rangle|$.

$$||\psi_k^x - x||^2 = \langle\psi_k^x|\psi_k^x\rangle - 2|\langle x|\psi_k^x\rangle| + \langle x|x\rangle = 2 - 2\underbrace{|\langle x|\psi_k^x\rangle|}_{|\langle x|\psi_k^x\rangle|^2 \geq \frac{1}{2}} \tag{141}$$

Therefore: $||\psi_k^x - x||^2 \leq 2 - \sqrt{2}$

Let's define two new metrics: $E_k \equiv \sum_x ||\psi_k^x - x||^2$ and $F_k \equiv \sum_x ||x - \psi_k||^2$. Then let's check the bound of D_k:

$$D_k = \sum_x ||(\psi_k^x - x) + (x - \psi_k)||^2 \tag{142}$$

$$D_k \geq \sum_x ||\psi_k^x - x||^2 - 2\sum_x ||\psi_k^x - x|| \cdot ||x - \psi_k|| + \sum_x ||x - \psi_k||^2 \tag{143}$$

$$D_k \geq E_k + F_k - 2\sum_x ||\psi_k^x - x|| \cdot ||x - \psi_k|| \tag{144}$$

From Cauchy-Schwarz, we have:

$$\sum_x ||\psi_k^x - x|| \cdot ||x - \psi_k|| \leq \underbrace{\left(\sum_x ||\psi_k^x - x||^2\right)^{\frac{1}{2}}}_{\sqrt{E_k}} \underbrace{\left(\sum_x ||x - \psi_k||^2\right)^{\frac{1}{2}}}_{\sqrt{F_k}} \tag{145}$$

Therefore:

$$\sum_x ||\psi_k^x - x|| \cdot ||x - \psi_k|| \leq \sqrt{E_k F_k} \tag{146}$$

Then, we can finally find the lower bound for D_k:

$$D_k \geq E_k + F_k - 2\sqrt{E_k F_k} = (\sqrt{F_k} - \sqrt{E_k})^2 \tag{147}$$

Now let's check the lower bound for F_k:

$$\sum_x ||\psi - x||^2 = \sum_x \underbrace{|\langle\psi|\psi\rangle|^2}_{=1} + \underbrace{|\langle x|x\rangle|^2}_{=1} - (\langle x|\psi\rangle + \langle\psi|x\rangle) \tag{148}$$

$$= 2N - \sum_x (\langle x|\psi\rangle + \langle\psi|x\rangle) \tag{149}$$

We can use that $\sum_x(\langle x|\psi\rangle + \langle\psi|x\rangle) \leq 2\sum_x |\langle x|\psi\rangle|$ and Schwarz-Cauchy:

$$(\sum_x |\langle x|\psi\rangle| \cdot 1)^2 \leq \sum_x \underbrace{|\langle x|\psi\rangle|^2}_{=1} \cdot \sum_{x'} 1 = N \tag{150}$$

Therefore:

$$\sum_x ||\psi - x||^2 \geq 2N - 2\sqrt{N} \tag{151}$$

We have shown that $F_k \geq 2N - 2\sqrt{N}$ and $E_k \leq (2-\sqrt{2})N$. Combining these two results on D_k, we have:

$$D_k \geq (\sqrt{2N} - \sqrt{(2-\sqrt{2})N})^2 \geq (\sqrt{2} - \sqrt{2-\sqrt{2}})^2 N \tag{152}$$

Therefore, we've proved that:

$$D_k \geq cN \tag{153}$$

We've shown that $D_k \leq 4k^2$ and $D_k \geq cN$. Combining those results:

$$4k^2 \geq cN \Rightarrow k \geq \sqrt{\frac{cN}{4}} \tag{154}$$

Therefore, to achieve probability of at least $\frac{1}{2}$, we need $\Omega(\sqrt{N})$ gates. □

Then Grover's Algorithm is optimal, there is no way that we can improve it. There is no indication that we can solve

NP-Complete problems on Quantum Computers. Therefore, NP $\not\subset$ BQP relative to an oracle.

2.6 Quantum Fourier Transform

Fourier transform is a very important tool to signal processing. Here we will build a quantum version of a Discrete Fourier Transform that will be used in many quantum algorithms.

2.6.1 Discrete Fourier Transform

Discrete fourier transform is the map between two strings $F : (x_0, x_1, \ldots, x_{N-1}) \to ((y_0, y_1, \ldots, y_{N-1}))$.

$$F(x_k) = y_k = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} x_l \omega_N^{lk} \tag{155}$$

Where: $\omega_N^{lk} = exp\left(2\pi i \frac{lk}{N}\right)$

2.6.2 Quantum Fourier Transform

Quantum Fourier transform does the same thing, but using quantum states: $F : \sum_{i=0}^{N-1} x_i |i\rangle \to \sum_{i=0}^{N-1} y_i |i\rangle$

$$F(x_k) = y_k = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} x_l \omega_N^{lk} \tag{156}$$

Where: $\omega_N^{lk} = exp\left(2\pi i \frac{lk}{N}\right)$

This can be represented by an unitary matrix:

$$F = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \omega_N^{lk} |y\rangle \langle x| \tag{157}$$

To find the circuit representation of the Fourier transform we need to see how it works on 2^N qubits first:

$$\begin{aligned}
F\left(|m\rangle_n\right) &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{km} |k\rangle_n \\
&= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} exp\left(i \frac{2\pi}{N} mk\right) |k\rangle_n \\
&= \frac{1}{\sqrt{N}} \sum_{k_{n-1}=0}^{1} \cdots \sum_{k_0=0}^{1} exp\left(i \frac{2\pi}{N} m \sum_{l=1}^{n} \frac{k_{n-l}}{2^l}\right) |k_{n-1} \ldots k_0\rangle_n \\
&= \frac{1}{\sqrt{N}} \sum_{k_{n-1}=0}^{1} \cdots \sum_{k_0=0}^{1} \bigotimes_{l=1}^{n} exp\left(i \frac{2\pi}{N} m \frac{k_{n-l}}{2^l}\right) |k_{n-l}\rangle_n \\
&= \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left[|0\rangle + exp\left(i \frac{2\pi m}{2^l}\right) |1\rangle\right]
\end{aligned} \tag{158}$$

Now we use the binary representation of $\frac{m}{2^l}$:

$$\begin{aligned}\frac{m}{2^l} &= \sum_{p=1}^{n} m_{n-p}2^{n-p-l} = m_{n-1}2^{n-1-l} + \cdots + m_l 2^0 + \cdots + m_0 2^{-l} \equiv m_{n-1}\ldots m_l.m_{l-1}\ldots m_0 \\ &= \sum_{p=1}^{n-l} m_{n-p}2^{n-p-l} + \sum_{p=1}^{l} \frac{m_{l-p}}{2^l}\end{aligned} \tag{159}$$

Therefore:

$$exp\left[i\frac{2\pi m}{2^l}\right] = exp\left[i2\pi \sum_{p=1}^{n-l} m_{n-p}2^{n-p-l}\right] exp\left[i2\pi \sum_{p=1}^{l} \frac{m_{l-p}}{2^l}\right] = exp\left[i2\pi \sum_{p=1}^{l} \frac{m_{l-p}}{2^l}\right] \tag{160}$$

Now, we have that:

$$F(|m\rangle_n) = \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left[|0\rangle + exp\left[i2\pi \sum_{p=1}^{l} \frac{m_{l-p}}{2^l}\right]|1\rangle\right] \tag{161}$$

Consider first acting on 2 qubits(n=2):

$$\begin{aligned}F(|m\rangle_n) &= \frac{1}{\sqrt{4}} \bigotimes_{l=1}^{2} \left[|0\rangle + exp\left[i2\pi \sum_{p=1}^{l} \frac{m_{l-p}}{2^l}\right]|1\rangle\right] \\ &= \frac{1}{2}\left[|0\rangle + exp\left[i2\pi 0.m_0\right]|1\rangle\right] \otimes \left[|0\rangle + exp\left[i2\pi 0.m_1 m_0\right]|1\rangle\right]\end{aligned} \tag{162}$$

Consider the following gate:

$$\begin{aligned}R_k^{(0,1)}|m\rangle|0\rangle &= |0\rangle \\ R_k^{(0,1)}|m\rangle|1\rangle &= exp\left[i2\pi\frac{m}{2^k}\right]|1\rangle\end{aligned} \tag{163}$$

We can write the QFT as:

$$F = SWAP[H^{(0)}R_2^{(0,1)}H^{(1)}|m\rangle_2] \tag{164}$$

Where the SWAP gate changes the order of the qubits. This is represented by the following circuit:

This can be easily generalized for n qubits.

2.7 Quantum Phase Estimation

Suppose a unitary operator U has an eigenvector $|u\rangle$ with value $e^{2\pi i\phi}$ (This happens, because all unitary operators have eigenvalues v_i s.t $|v_i| = 1$). The goal of the algorithm is to estimate the phase ϕ. To perform this, we assume oracles capable of preparing the state $|u\rangle$ and perform the controlled U operation.

For this algorithm we will use 2 registers, the first contains t qubits initially in the state $|0\rangle$. the number of qubits, t, is chosen depending on two factors:

1) Number of digits of accuracy for ϕ;

2) With what probability we wish the phase estimation to be successful.

The second register begins in the state $|u\rangle$ and contains as many qubits as necessary to store $|u\rangle$, that is $\dim(U)/2$. We apply the following circuit:

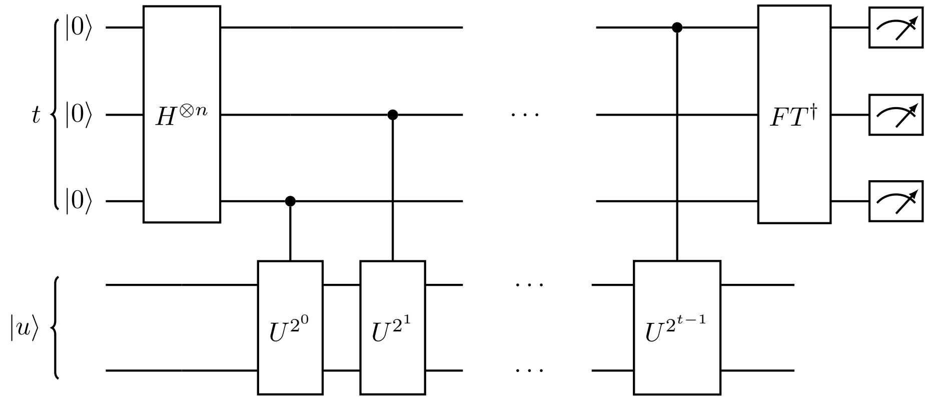


Figure 19: Circuit of the Quantum Phase Estimation.

Then we have:

$$\frac{1}{2^{t/2}}\bigg(|0\rangle + e^{2\pi i(2^{t-1}\phi}|1\rangle\bigg)\dots\bigg(|0\rangle + e^{2\pi i(2^{0}\phi}|1\rangle\bigg) = \frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e^{2\pi i\phi k}|k\rangle \tag{165}$$

Which is in the Fourier Basis, so the second stage of the algorithm is to return to the computational basis using the inverse of the Fourier Transform. Then the 3rd and final stage is to read out the first register by doing a measurement on the computational basis.

Then the Quantum Phase Estimation circuit is the following:

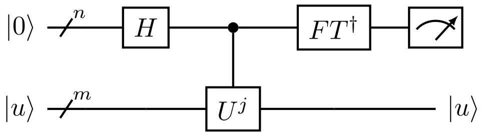


Figure 20: Full circuit of the Quantum Phase Estimation.

To understand better how it works, suppose ϕ may be expressed exactly by t bits, $\phi = 0.\phi_1\phi_2\dots\phi_t$, then before the Fourier transform inverse, we have:

$$\frac{1}{2^{t/2}}\Bigg[|0\rangle + e^{2\pi i 0.\phi_1}\Bigg]\dots\Bigg[|0\rangle + e^{2\pi i 0.\phi_1\phi_2\dots\phi_t}\Bigg] \tag{166}$$

Then we apply the Fourier transform, thus we get exactly $|\phi_1\phi_2\dots\phi_t\rangle$! Then:

$$\frac{1}{2^{t/2}}\sum_{j=0}^{2^t-1} e^{2\pi i\phi j}|j\rangle|u\rangle \xrightarrow{FT^\dagger} |\phi_1\phi_2\dots\phi_t\rangle|u\rangle \tag{167}$$

2.7.1 Performance and Requirements

Let b be the integer in the range 0 to 2^t-1 s.t. $\frac{b}{2^t} = 0.b_1\dots b_t$ is the best t bit approximation to ϕ which is less than ϕ. The difference $\delta \equiv \phi - \frac{b}{2^t}$ between ϕ and $\frac{b}{2^t}$ satisfies $0 \le \delta \le 2^{-t}$. We aim to show that at the end of the phase estimation procedure we get a result which is close to b, and this enable us to estimate ϕ accurately, with high probability. After the $FT^\dagger$:

$$|\psi\rangle = \frac{1}{2^t}\sum_{k,l=0}^{2^t-1} e^{-\frac{2\pi ikl}{2^t}}e^{2\pi i\phi k}|l\rangle \tag{168}$$

Let α_t be the amplitude of $|(b+l) \mod 2^t\rangle$:

$$|\alpha_t\rangle \equiv \big\langle (b+l)mod2^t\big|\psi\big\rangle = \frac{1}{2^t}\sum_{k=0}^{2^t-1}\Bigg[e^{2\pi i(\phi - \frac{b+l}{2^t}}\Bigg]^k \tag{169}$$

Using the geometric series $\sum_{k=0}^{n} x^k = \frac{1-x^n}{1-x}$, we have that:

$$\alpha_t = \frac{1}{2^t}\Bigg[\frac{1-e^{2\pi i(2^t\phi-(b+l)}}{1-e^{2\pi i(\phi-\frac{b+l}{2^t}}}\Bigg] = \frac{1}{2^t}\Bigg[\frac{1-e^{2\pi i(2^\delta-l)}}{1-e^{2\pi i(\delta-\frac{l}{2^t}}}\Bigg] \tag{170}$$

Suppose that the outcome of the measurement is m. We aim to bound the probability of obtaining a value of m such that $|m-b|>e$, where e is a positive integer characterizing our desired tolarance error. The probability of observing such m is given by:

$$p(|m-b|>e) = \sum_{-2^{t-1}<l\le-(e+1)}|\alpha_l|^2 + \sum_{e+1<l\le 2^t}|\alpha_l|^2 \tag{171}$$

For any $\theta\in\mathbb{R}$, $|1-e^{i\theta}|\le 2$, so:

$$|\alpha_l| \le \frac{2}{2^t|1-e^{2\pi i(\delta-\frac{l}{2^t}}} \tag{172}$$

We also have that $|1-e^{i\theta}| \ge \frac{2|\theta|}{\pi}$, whenever $\theta\in[-\pi,\pi]$. But when $-2^{t-1}<l\le 2^{t-1}$ we have that $-\pi \le 2\pi(\delta-\frac{l}{2^t}) \le \pi$, thus:

$$|\alpha_t| \le \frac{1}{2^{t+1}(\delta-\frac{l}{2^t})} \tag{173}$$

Applying this on 171:

$$p(|m-b|>e) \leq \frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(e+1)} \frac{1}{(l-2^t\delta)^2} + \sum_{l=e+1}^{2^{t-1}} \frac{1}{(l-2^t\delta)^2}\right] \tag{174}$$

By definition $0 \leq 2^t\delta \leq 1$, we have:

$$\begin{aligned} p(|m-b|>e) \leq & \frac{1}{4}\left[\sum_{l=-2^{t-1}+1}^{-(e+1)} \frac{1}{l^2} + \sum_{l=e+1}^{2^{t-1}} \frac{1}{(l-1)^2}\right] \\ \leq & \frac{1}{2}\sum_{l=e}^{2^{t-1}-1} \frac{1}{l^2} \\ \leq & \frac{1}{2}\int_{e-1}^{2^{t-1}-1} \frac{dl}{l^2} = \frac{1}{2(e-1)} \end{aligned} \tag{175}$$

Suppose we wish to approximate ϕ to an accuracy 2^{-n}, that is, we choose $e = 2^{t-n} - 1$. By making use of $t = n + p$ in the phase estimation, then the probability of obtaining an approximation correct to this accuracy is:

$$p(|m-b|>e) \leq \frac{1}{2(2^{t-n}-2)} = \frac{1}{2(2^p-2)} \tag{176}$$

Thus to obtain ϕ with accuracy of n bits with probability of success at least $1-\epsilon$, we choose:

$$\begin{aligned} \epsilon = \frac{1}{2(2^p-2)} \Rightarrow 2^p = 2 + \frac{1}{2\epsilon} \\ p = \log\left(2 + \frac{1}{2\epsilon}\right) \end{aligned} \tag{177}$$

Therefore we can choose t according to our desired probability:

$$t = n + \log\left(2 + \frac{1}{2\epsilon}\right) \tag{178}$$

It is important to point that it is not needed necessarily to initialize the ancilla into the eigenvector of the unitary matrix for the QPE. Let's suppose $|\psi\rangle = \sum_u c_u |u\rangle$ and the eigenstate $|u\rangle$ has eigenvalues $e^{2\pi i\phi_u}$. Then:

$$\text{QPE}\,|0\rangle\,|\psi\rangle = \sum_u c_u\,|\phi_u\rangle\,|u\rangle \tag{179}$$

This work because the QPE is linear, so you get the QPE for each eigenvector $|u\rangle$. In this case, reading the first register we have a good approximation to ϕ_u, where u is chosen at random with probability $|c_u|^2$.

2.8 Order Finding

For any $x, N \in \mathbb{Z}^+$, with no common factors, that is $\gcd(x, N) = 1$, the order of x modulo N is defined as the least positive integer, r, such that $x^r \equiv 1 \mod N$. The order-finding problems is to determine the order for some specified x and N.

This problem is believed to be hard on classical computers, there is no known algorithm to solve it using polynomial resources in the $O(L)$ bits needed to specify the problem, where $L \equiv \lceil \log N \rceil$. In the other hand, there is a quantum algorithm for order finding using polynomial resources that uses the phase estimation algorithm applied to the following unitary operator:

$$M_x |y\rangle = |xy \mod N\rangle \tag{180}$$

with $y \in \{0,1\}^L$. Note that when $N \leq y \leq 2^L - 1$, we use the convention that $xy \mod N$ is just y, therefore U only acts non-trivially when $0 \leq y \leq N-1$.

Note that M_x is unitary: ($\tilde{y} \in \mathbb{Z}_N$

$$M_x M_{x^-1} |\tilde{y}\rangle = \big|x(x^-1y) \mod N\big\rangle = |\tilde{y}\rangle$$

$$M_{x^-1} M_x |\tilde{y}\rangle = \big|x^-1(xy) \mod N\big\rangle = |\tilde{y}\rangle$$

What are the eigenvectors and eigenvalues of M_x?

Consider the following states:

- $|\psi_0\rangle$:

$$|\psi_0\rangle = \frac{1}{\sqrt{r}}\Big[\big|\tilde{1}\big\rangle + |\tilde{x}\rangle + \cdots + \big|\tilde{x}^{r-1}\big\rangle\Big]$$

$$M_x |\psi_0\rangle = \frac{1}{\sqrt{r}}\Big[|\tilde{x}\rangle + \big|\tilde{x}^2\big\rangle + \cdots + \big|\tilde{x}^{r-1}\big\rangle + |\tilde{x}^r\rangle\Big] = 1\,|\psi_0\rangle$$

- $|\psi_1\rangle$, given that $w_r = e^{\frac{2\pi i}{r}}$:

$$|\psi_1\rangle = \frac{1}{\sqrt{r}}\Big[\big|\tilde{1}\big\rangle + \omega_r^{-1}|\tilde{x}\rangle + \cdots + \omega_r^{-(r-1)}\big|\tilde{x}^{r-1}\big\rangle\Big]$$

$$\begin{aligned}M_x |\psi_1\rangle =&\frac{1}{\sqrt{r}}\Big[|\tilde{x}\rangle + \omega_r^{-1}\big|\tilde{x}^2\big\rangle + \cdots + \omega_r^{-(r-1)}|\tilde{x}^r\rangle\Big]\\ =&\frac{\omega_r}{\sqrt{r}}\Big[\omega_r^{-1}|\tilde{x}\rangle + \omega_r^{-1}\big|\tilde{x}^2\big\rangle + \cdots + \overbrace{\omega_r^{-(r)}}^{=1}\big|\tilde{1}\big\rangle\Big]\\ =&\omega_r|\psi_1\rangle\end{aligned}$$

Thus in general the eigenstate and its eigenvalues are given by:

$$|u_s\rangle \equiv \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp\left[-\frac{2\pi isk}{r}\right]\left|\tilde{x}^k\right\rangle \tag{181}$$

for $s \in [0, r-1]$. Lets show this:

$$\begin{aligned}
M_x\,|u_s\rangle =& \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp\left[\frac{-2\pi isk}{r}\left|\tilde{x}^{k+1}\right\rangle\right]\\
=& \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp\left[\frac{-2\pi is(k-1)}{r}\left|\tilde{x}^{k}\right\rangle\right]\\
=& \exp\left[\frac{2\pi is}{r}\right]\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp\left[\frac{-2\pi isk}{r}\left|\tilde{x}^{k}\right\rangle\right]\\
=& \exp\left[\frac{2\pi is}{r}\right]|u_s\rangle
\end{aligned}$$

Therefor we can use phase estimation to otain the corresponding eigenvalues $\exp\left(\frac{2\pi is}{r}\right)$, which we can obtain the order r!

To show that M_x is unitary is just to show that $x \in \mathbb{Z}_n$, which is trivial.

There are two important requirements for us to use phase estimation:

1. We need to prepare $|u_s\rangle$ efficiently which we require to know r, but we can do this with a trick:

$$\begin{aligned}
\sum_{s=0}^{r-1}|u_s\rangle =& \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\sum_{k=0}^{r-1}\exp\left[\frac{-2\pi iks}{r}\right]\left|\tilde{x}^k\right\rangle \ , \ \sum_{s=0}^{r-1}e^{\frac{-2\pi iks}{r}} = \delta_{k,0}r\\
=& \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}r\delta_{k,0}\left|\tilde{x}^k\right\rangle = \sqrt{r}\left|\tilde{x}^0\right\rangle\\
=& \sqrt{r}\left|\tilde{x}^0\right\rangle
\end{aligned}$$

Therefore:

$$|1\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle \tag{182}$$

2. We must have an efficient way to construct a controlled-U^{2^j} operation for any integer j. We want to compute:

$$|z\rangle\,|y\rangle \to |z\rangle\,U^{z_t 2^{t-1}}\ldots U^{z_1 2^0}\,|y\rangle$$

The basic idea is to reversibly compute the function $x^z \mod N$ of z in a third register and then to reversibly multiply the contents on the second register by $x^z \mod N$, using the trick of uncomputation to erase the content of the third register upon completion.

The algorithm for modular exponentiation has two phases:

P1) Compute $x^2 \mod N = \tilde{x}^2$ by squaring $\tilde{x}$ and square to compute $\tilde{x}^4$, $\tilde{x}^6$, $\ldots$ until you can compute $\tilde{x}^{2^j}$. We use $t = 2L + 1\rceil \log(2 + \frac{1}{2\epsilon}\lceil = O(L)$ squaring operations at a cost of $O(L^2)$ each, therefore having a cost of $O(L^3)$ for P1.

P2) We can see that:

$$\tilde{x}^z = (\tilde{x}^{z_t 2^{t-1}}) \ldots (\tilde{x}^{z_1 2^0})$$

Therefore we perform $t-1$ modular multiplications at a cost $O(L^2)$ each, this is computed using $O(L^3)$ gates. Thus, we can construct a reversible circuit with t bit register and a L bit register which can be translated to a quantum circuit with $O(L^3)$ gates.

The algorithm is:

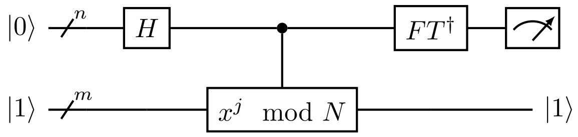


Figure 21: Full circuit for modular exponentiation.

Let's see how this algorithm works:

$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |j\rangle\, U^j\, |1\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |j\rangle\, U^j \left(\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle \right) \\ &= \frac{1}{r} \sum_{j=0}^{r-1} \sum_{s=0}^{r-1} |j\rangle\, e^{\frac{2\pi i s j}{r}}\, |j\rangle \left(\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{\frac{-2\pi i k s}{r}} \left|\tilde{x}^k\right\rangle \right) \\ &= \frac{1}{r^{\frac{3}{2}}} \sum_{j=0}^{r-1} \sum_{k=0}^{r-1} \left(\overbrace{\sum_{s=0}^{r-1} e^{\frac{2\pi i (j-k)}{r}}}^{r\delta_{j,k}} |j\rangle \left|\tilde{x}^k\right\rangle \right) \\ &= \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |j\rangle \left|\tilde{x}^j\right\rangle \end{aligned} \tag{183}$$

We can show that:

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{\frac{2\pi i s j}{r}}\, |u_s\rangle = \left|\tilde{x}^j\right\rangle$$

$$\frac{1}{r} \sum_{k=0}^{r-1} e^{\frac{2\pi i s j}{r}}$$

$$\begin{aligned} \sum_{k=0}^{r-1} e^{\frac{2\pi i k s}{r}} \left|\tilde{x}^k\right\rangle &= \frac{1}{r} \sum_{k=0}^{r-1} \overbrace{\left(\sum_{s=0}^{r-1} e^{\frac{2\pi i s (j-k)}{r}} \right)}^{r\delta_{k,j}} \left|\tilde{x}^k\right\rangle \\ &= \sum_{k=0}^{r-1} \delta_{k,j} \left|\tilde{x}^k\right\rangle = \left|\tilde{x}^j\right\rangle \end{aligned}$$

Thus we can rewrite equation 183 as:

$$
\begin{aligned}
|\psi\rangle = \frac{1}{\sqrt{r}}\sum_{j=0}^{r-1}|j\rangle\left|\tilde{x}^j\right\rangle &= \frac{1}{r}\sum_{s=0}^{r-1}\overbrace{\sum_{j=0}^{r-1}e^{\frac{2\pi isj}{r}}}^{\mathrm{FT}^\dagger}|j\rangle|u_s\rangle \\
&= \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left|\overline{s/r}\right\rangle|u_s\rangle
\end{aligned}
$$

After measuring the 1st register we get exactly $\left|\overline{s/r}\right\rangle$. This is not yet our answer for finding r we need to know what $\overline{s}$ is, this is done using the continued fraction expansion.

When we say continued fraction expansion, we mean that we want to write $\frac{A}{B}$, $\gcd(A,B)=1$ as:

$$
\frac{A}{B} = a_0 + \frac{1}{a_1 + \frac{1}{\dots + \frac{1}{a_m}}} = [a_0, a_1, \dots, a_m] \tag{184}
$$

Let's see one example of the continued fraction expansion:

$$
\begin{aligned}
\frac{31}{13} &= 2 + \frac{5}{13} = 2 + \frac{1}{\frac{13}{5}} = 2 + \frac{1}{2+\frac{3}{5}} \\
&= 2 + \frac{1}{2+\frac{1}{1+\frac{2}{3}}} = 2 + \frac{1}{2+\frac{1}{1+\frac{1}{2}}} = [2,2,1,1,2]
\end{aligned}
$$

This is done by $O(L^2)$ steps for division with $O(L)$ inverting steps, so computed with $O(L^3)$ operations.

There is one step left, we need to show that obtaining r from the continued fraction expansion is efficient, we thus demonstrate the following theorem:

Theorem 2.3. *Suppose p/q is a rational number such that:*

$$
\left|\frac{p}{q} - x\right| \leq \frac{1}{2q^2}
$$

Then p/q is convergent given the continued fraction for x, and thus can be computed in $O(L^3)$ operations using the continued fractions algorithm.

Proof. Let $p/q = [a_0, \dots, a_n]$ be the continued fraction expansion for p/q. ad define $p_j \equiv a_j p_{j-1} + p_{j-2}$ and $q_j \equiv a_j q_{j-1} + q_{j-2}$, so that $\frac{p_n}{q_n} = \frac{p}{q}$. We define δ by the following equation:

$$
x \equiv \frac{p_n}{q_n} + \frac{\delta}{2q_n^2} \tag{185}
$$

We also define λ by the following equation:

$$
\lambda \equiv 2\left(\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta}\right) - \frac{q_{n-1}}{q_n} \tag{186}
$$

we define λ because it satisfies the following equation:

$$x = \frac{\lambda p_n + p_{n-1}}{\lambda q_n + q_{n-1}} \xrightarrow{x}= [a_0, \dots, a_n, \lambda]$$

Firstly we need to show that: $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$, this will be done by induction that this is true for $n \geq 1$, using that $\gcd(p_n, q_n) = 1$

- For $n = 1$:

$$(p_0 = a_0 \ , \ q_0 = 1 \ , \ \text{and} p_1 = 1 + a_0 a_1)$$

$$q_1 p_0 - p_1 q_0 = (a_1 q_0) a_0 - (1 + a_0 a_1) = -1$$

- Suppose for n is valid:

$$q_n p_{n-1} - p_n q_{n-1} = (-1)^n$$

- For n=n+1:

$$\begin{aligned} q_{n+1} p_n - p_{n+1} q_n =& (a_n q_n + q_{n-1}) p_n - (a_n p_n + p_{n-1}) q_n \\ =& a_n q_n p_n + p_n q_{n-1} - a_n p_n q_n - p_{n-1} q_n \\ =& q_{n-1} p_n - p_{n-1} q_n \\ =& -(q_n p_{n-1} - p_n q_{n-1}) = (-1)(-1)^n \\ =& (-1)^{n+1} \end{aligned}$$

Therefore we can choose n being even:

$$\lambda = \frac{2}{\delta} - \frac{q_{n-1}}{q_n} > 2 - 1 > 1$$

Therefore λ is a rational number greater than 1, so has a simple continued fraction, $\lambda = [b_0, \dots, b_m]$ and so $x = [a_0, \dots, a_n, b_0, \dots, b_m]$ is a simple finite continued fraction for x with p/q as a convergent. □

For our problem $x = \phi$ and $\frac{p}{q} = \frac{s}{r}$. Since ϕ is an approximation of $\frac{s}{r}$ accurate to $2L + 1$ bits, it follows that $\left|\frac{s}{r} - \phi\right| \leq 2^{-2L-1} \leq \frac{1}{2r^2}$, since $r \leq N \leq 2^L$. Thus, the theorem applies.

Summarizing, given ϕ the continued fractions algorithm produces both numbers s' and r' efficiently with no common factors such that $\frac{s'}{r'} = \frac{s}{r}$. The number r' is our candidate for the order which we can verify by doing $x^{r'} \mod N$ and see if it is equal to 1.

Performance How can this algorithm fail?

- The phase estimation gives a bad estimate fo $\frac{s}{r}$. This occurs with probability ϵ and we can surpass this error growing our circuit.

- It might be that s and r have a common factor, then we would find r' that is a factor of r, but we have three ways around this problem:

  1. The probability of a randomly chosen $s \in [0, r-1]$ to be prime is at least $\frac{1}{2\log r} > \frac{1}{2\log N}$. Thus, repeating the algorithm $2\log N$ times we will, with high probability, observe a phase $\frac{s}{r}$ such that s and r are coprime.

  2. If $r' \neq r$ is guaranteed to be a factor of r, unless $s = 0$ which this possibility occurs with a probability $\frac{1}{r} \leq \frac{1}{2}$ which can be discounted with a few repetitions.

     Suppose we replace a by $a' \equiv a^{r'} \mod N$. Then the order of a' is $\frac{r}{r'}$. We can repeat the algorithm, and try to compute the order of a', if we succeed we compute the order of a, because $r = r'\frac{r}{r'}$. If we fail, we iterate until we determine the order of a. At most $\log r = \log L$ iterations are required, since each repetition reduces the order of the current candidate $a^{'\cdots}$ by a factor of at least 2.

  3. The other idea is to repeat the phase estimation and continued fraction twice, we would obtain s_1' , r_1' and s_2' , r_2'. Provided that s_1' and s_2' has no common factors, r may be extracted by taking the least common multiple of r_1 and r_2. The probability of s_1' and s_2' have no common factors is given by:

     $$1 - \sum_q p(q|s_1')p(q|s_2')$$

     where the sum is over all primes q and $p(x|y)$ is the probability of x dividing y.

     If q divides s_1' then it must also divide the true value of s, s_1, so to upper bound $p(q|s_1')$ it suffices to upper bound $p(q|s_1)$, where s_1 is chosen at random from 0 through r-1. Since $p(q|s_1) \leq \frac{1}{s}$ and $p(q|s_2) \leq \frac{1}{s}$.

     $$1 - \sum_q p(q|s_1')p(q|s_2') \geq 1 - \sum_q \frac{1}{q^2} \geq \frac{1}{4}$$

     And thus the probability of obtaining the correct r is $\frac{1}{4}$.

2.9 Factoring

The factoring problem can be reduced to the order-finding problem, this occurs into two basic steps realized by Peter Shor in 1994 [16]:

1. We need to show that we can compute a factor of N if we can find a non-trivial solution $x \not\equiv \pm 1 \mod N$ to the equation $x^2 \equiv 1 mod N$.

2. Show that a randomly chosen y co-prime to N is quite likely to have an order r which is even, and such that $y^{\frac{r}{2}} \not\equiv \pm 1 \mod N$, and thus $x \equiv y^{\frac{r}{2}} \mod N$ is a non-trivial solution to $x^2 \equiv 1 \mod N$.

These two steps are supported by the following theorems:

Theorem 2.4. *Suppose N is an L bit composite number, and x is a non-trivial solution to the equation $x^2 \equiv 1 \mod N$ in the range $1 \leq x \leq N$, that is neither $x \equiv 1 \mod N$ nor $x \equiv N - 1 \equiv -1 \mod N$. Then at least one of* $\gcd(x-1, N)$ *and* $\gcd(x+1, N)$ *is a non-trivial factor of N thatcan be computed using $O(L^3)$ operations.*

Theorem 2.5. *Suppose $N = \prod_{i=1}^{m} p_i^{\alpha_i}$ is the prime factorization of an odd composite positive integer. Let x be an integer chosen uniformly at random, subject to the requirements that $1 \leq x \leq N-1$ and x co-prime to N. Let r be the order of x* $\mod N$*. Then:*

$$p(r \text{ is even and } x^{\frac{r}{2}} \not\equiv -1 \mod N) \geq 1 - \frac{1}{2^m}$$

All steps but the order finding subroutine can be efficiently computed on a classical computer, thus we have the following algorithm:

Algorithm 2: Shor's Factoring Algorithm

```
input    : A composite number N
output   : A non-trivial factor of N
runtime: $O((\log N)^3)$ operations succeed with probability $O(1)$
if $N$ is even then
    if $N = a^b$ for integers $a \geq 1$ and $b \geq 2$ then
        Randomly choose $x \in [1, N-1]$
        if gcd$(x, N) > 1$ then
            Quantum step: Use order-finding subroutine to find the order r of $x \mod N$
            if $r$ is even and $x^{\frac{r}{2}} \not\equiv -1 \mod N$ then
                Compute gcd$(x^{\frac{r}{2}} - 1, N)$ and gcd$(x^{\frac{r}{2}} + 1, N)$, and test if one of these is a non-trivial factor returning that factor.
            else
                The algorithm fails.
            end
        else
            Return the factor gcd$(x, N)$
        end
    else
        Return the factor a
    end
else
    Return the factor 2
end
```

Example 1. *Let's factor the number N=91.*

It is obvious that N is not even and does not have the form $N = a^b$. Now we need to choose a random number x such that $x \in [1, N-1]$, we choose $x = 4$ and we get $\gcd(4, 91) = 1$ *thus 4 is co-prime with 91. Now we need to find the order of $x \mod N$, this would be done in a quantum computer using order finding and we would get $r = 6$.*

Now for the final step:

$$x^{6/2} \mod 91 \equiv 64 \mod 91 \not\equiv -1 \mod 91$$

And we get that $\gcd(64 - 1, 91) = 7$. *Therefore:* $91 = 7 * 3$*!*

2.10 Period Finding

Problem: Suppose f is a periodic function producing a single bit as output and such that $f(x+r) = f(x)$, for some unknown $r \in (0, 2^L)$, where $x, r \in \{0, 1, 2, \dots\}$.

Given a quantum blackbox U, which perform $U|x\rangle|y\rangle \to |x\rangle|y \oplus x\rangle$, how many black box queries and other operations are needed to determine r? Here we present and algorithm to solve with only one query and $O(L^2)$ other operations.

Algorithm 3: Period finding algorithm

input :

1. A blackbox which perform $U|x\rangle|y\rangle = |x\rangle|x + f(x)\rangle$
2. A state that store the function evaluation, initialized at $|0\rangle$
3. $t = O(L + \log(\frac{1}{\epsilon})$ qubits initialized at $|0\rangle$

output : The least integer $r > 0$ such that $f(x+r) = f(x)$

runtime: One use of U, and $O(L^2)$ operations. Succeeds with probability of $O(1)$.

1. $|0\rangle|0\rangle \to$ initial state.
2. Create superposition:
$$\frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|0\rangle$$
3. Apply U:
$$\frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|f(x)\rangle \approx \frac{1}{\sqrt{r2^t}} \sum_{l=0}^{r-1} \sum_{x=0}^{2^t-1} e^{\frac{2\pi i l x}{r}} |x\rangle \left|\hat{f}(l)\right\rangle$$
4. Apply reverse Fourier Transform to 1st register:
$$\frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} \left|\tilde{l/r}\right\rangle \left|\hat{f}(l)\right\rangle$$
5. Measure first register to get $\tilde{l/r}$.
6. Apply continuous fraction to get r.

The key for this algorithm is step **3**:

$$\left|\hat{f}(l)\right\rangle = \frac{1}{\sqrt{r}} \sum_{x=0}^{r-1} e^{\frac{-2\pi i l x}{r}} |f(x)\rangle$$

Where $\left|\tilde{f}(x)\right\rangle$ is the fourier transform of $|f(x)\rangle$. This is based on:

$$|f(x)\rangle = \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} e^{\frac{2\pi i l x}{r}} \left|\tilde{f}(l)\right\rangle$$

Where

$$\sum_{l=0}^{r-1} e^{\frac{2\pi i l x}{r}} = \begin{cases} r, & \text{if } x = kr \ , \ k \in \mathbb{Z} \\ 0, & \text{otherwise} \end{cases}$$

2.11 Hidden Subgroup Problem

Problem: Given a known group G and a function $f : G \to S$, where S is a some finite set. Suppose f has the property that there exists a subgroup $H \leq G$ such that f is a constant within each coset, and distinct on different cosets $f(g) = f(g')$ iff $gH = g'H$. We need to find H.

We assume f can be computed efficiently (in polynomial time in $\log |G|$. Since H may be large, "Finding H" typically means finding a generating set for H.

It is interesting to note that such an abstract problem has many useful examples that are mapped onto this problem:

- **Simon's Problem:** Here G is the additive group $\mathbb{Z}_2^n = \{0,1\}^n$ of size 2^n, $H = \{0, s\}$ for a "hidden" $s \in 0, 1^n$, and satisfies $f(x) = f(y)$ iff $x - y \in H$. Finding s solves Simon's problem.

- **Period Finding:** Given an x that is coprime to N and associated function $f : \mathbb{Z} \to \mathbb{Z}_N^*$ by $f(a) \equiv x^a \mod N$, find the period r of f. Since $\langle x \rangle$ is the size-r subgroup of the group $\mathbb{Z}_N^*$, the period r divides $|\mathbb{Z}_N^*| = \phi(N)$ [Lagrange's Theorem]. Hence we can restrict the domain of f to $\mathbb{Z}_{\phi(N)}^*$. This problem is an instance of the HSP:

  Let $G = \mathbb{Z}_{\phi(N)}$ and consider its subgroup $H = \langle r \rangle$ of all multiples of r up to $\phi(N)$ [$H = r\mathbb{Z}_{\phi(n)} = \{0, r, 2r, \ldots, \phi(N) - r\}$]. Note that becaus of its periodicity, f is constant on each coset $s + H$ and distinct on different cosets. Also, f is efficiently computable by repeated squaring. Since H is generated by r, finding the generator of H solves the period-finding problem.

We will construct an efficient quantum algorithm if the group is Abelian, but first we need to learn some representation theory.

2.11.1 Representation Theory

The idea is to replace group elements by matrices in order to use linear algebra in group theory.

Definition 2.1 (Representation of a Group)**.** A d-dimensional representation of an multiplicative group G is a map $f : g \mapsto \rho(g)$ from G to the set of all dxd invertible complex matrices, satisfying: $\rho(gh) = \rho(g)\rho(h)$, $\forall g, h \in G$.

The latter property makes rho a homomorphism. The *character* corresponding to ρ is the map $\chi_\rho : G \to \mathbb{C}$ defined by $\chi_\rho(g) = \text{Tr}\big(\rho(g)\big)$.

Let's now consider G abelian. We may consider $d = 1$ without loss of generality, then ρ and χ_ρ are the same function. THe complex values of $\chi_\rho(g)$ have modulus 1, because $|\chi(g^k)| = |\chi(g)|^k$, $\forall k \in \mathbb{Z}$.

Theorem 2.6 (Basis Theorem)**.** *Every finite abelian group G is isomorphic to a direct product $\mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_k}$ of cyclic groups.*

Let's consider just one cyclic group $(\mathbb{Z}_N, +)$. The discrete fourier transform is an $N \times N$ matrix, ignoring the normalizing factor the k-th column may be viewed as a map $\chi_k : \mathbb{Z}_N \to \mathbb{C}$ defined by $\chi_k(j) = \omega_N^{jk}$, where $\omega_N = e^{\frac{2\pi i}{N}}$. Let's check if it is a representation of $\mathbb{Z}_N$:

$$\chi_k(j+j') = \omega_N^{(j+j')k} = e^{\frac{2\pi i}{N}[(j+j')k]} = e^{\frac{2\pi ijk}{N}} e^{\frac{2\pi ij'k}{N}}$$
$$= \omega_N^{jk}\omega_N^{j'k} = \chi_k(j)\chi_k(j')$$

In fact, the N characters corresponding to N columns of the Fourier matrix are all characters of $\mathbb{Z}_N$. For abelian groups G that are (isomorphic to) a product $\mathbb{Z}_{N_1} \times \ldots \mathbb{Z}_{N_k}$ of cyclic groups, the $|G| = N_1 \times \ldots N_k$ characters are just the product of the characters of the individual cyclic groups $\mathbb{Z}_{N_j}$.

The set of all characters of G forms a group $\hat{G}$ with operations of pointwise multiplication. This is called the dual group of G. If $H \leq G$, then the following is a sugroup of $\hat{G}$ of size $\frac{|G|}{|H|}$:

$$H^{\perp} = \{\ \chi_k \mid \chi_k(h) = 1\ \forall h \in H\}$$

We can now interpret the QFT in terms of characters. For $k \in \mathbb{Z}_N$:

$$|\chi_k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \chi_k |j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jk} |j\rangle$$

Then QFT maps $\mathbb{C}^N$ to the orthonormal basis of characters:

$$F_N : |k\rangle \mapsto |\chi_k\rangle$$

2.11.2 Algorithm for Hidden Subgroup

Now we can construct the algorithm for the hidden subgroup problem:

1. Start with $|\psi\rangle = |0\rangle |0\rangle$, where the two registers have $|G|$ and $|S|$ respectively.

2. Create a uniform superposition over G in the first register:

$$|\psi\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |0\rangle$$

3. Compute f on superposition, with $f : G \to S$:

$$|\psi\rangle = \frac{1}{\sqrt{G}} \sum_{g \in G} |g\rangle |f(g)\rangle$$

4. Measure the 2nd register. THis yields some value $f(s)$ for unknown $s \in G$. The 1st register collapses to a superposition over g with the same f-value as s (i.e. the coset $s + H$):

$$|\psi\rangle = \frac{1}{\sqrt{H}} \sum_{h \in H} |s + h\rangle$$

5. Apply the "QFT" corresponding to G:

$$|\psi\rangle = \frac{1}{\sqrt{H}} \sum_{h \in H} |\chi_{s+h}\rangle$$

6. Measure and output the resulting g.

The key ingredient of this algorithm is that step 5 maps the uniform superposition over the cosets $s + H$ to a uniform superposition over the labels of $H^{\perp}$:

$$\begin{aligned}
\frac{1}{\sqrt{H}} \sum_{h \in H} |\chi_{s+h}\rangle &= \frac{1}{\sqrt{|H||G|}} \sum_{h \in H} \sum_{g \in G} \chi_{s+h}(g) \, |g\rangle \\
&= \frac{1}{\sqrt{|H||G|}} \sum_{g \in G} \chi_s(g) \sum_{h \in H} \chi_h(g) \, |g\rangle \\
&= \sqrt{\frac{|H|}{|G|}} \sum_{g : \chi_g \in H^{\perp}} \chi_s(g) \, |g\rangle
\end{aligned}$$

The last equality follows from the orthogonality of the group H (χ_g restricted to H is a character to H, and it's the constant-1 character iff $\chi_g \in H^{\perp}$):

$$\sum_{h \in H} \chi_h(g) = \sum_{h \in H} \chi_g(h) = \begin{cases} |H| \ , \ \text{if } \chi_g \in H^{\perp} \\ 0 \ , \ \text{if } \chi_g \notin H^{\perp} \end{cases}$$

The above algorithm samples uniformly from the elements of $H^{\perp}$. Each such elements $\chi_g \in H^{\perp}$ gives us a constraint on H because $\chi_g(h) = 1 \ \forall h \in H$. Generating a small number of such elements will give sufficient information to find the generators of H itself. Let's consider our HSP examples:

- **Simon's Problem:** $G = \mathbb{Z}_2^N = \{0,1\}^N$ and $H\{0, s\}$. Setting up the superposition over G can e done by $H^{\otimes N}$ on the initial state $|0\rangle^{\otimes N}$. The "QFT" corresponding to G is just $H^{\otimes N}$. The 2^N character functions are $\chi_g(h) = (-1)^{x \cdot g}$. The algorithm will uniformly sample from labels of elements of:

$$H^{\perp} = \{\chi_g \mid \chi_g(h) = 1 \ \forall h \in H\} = \{\chi_g \mid g \cdot s = 0\}$$

  Accordingly, the algorithm samples uniformly from $g \in \{0,1\}^N$ such that $g \cdot s \equiv 0 \mod 2$. Doing this an expected $O(N)$ times gives $n - 1$ linearly independent equations about s, from which we can find s using Gaussian Elimination.

- **Period Finding:** $G = \mathbb{Z}_{\phi(N)}$ and $H = \langle r \rangle$, and:

$$H^{\perp} = \{\chi_b \mid e^{\frac{2\pi i b h}{\phi(N)}} = 1 \forall h \in H\} = \{\chi_b \mid \frac{br}{\phi(N)} \in \{0, \ldots, r-1\}\}$$

  Accordingly, the output of the algorithm is an integer multiple $b = \frac{c\phi(N)}{r}$ of $\frac{\phi(N)}{r}$, for uniformly random $c \in \{0, \ldots r-1\}$

2.12 Quantum Counting

Now we want to find how many solutions, M, are on an N item search problem. For a classical computer we would need $\Omega(N)$ consultations to estimate M.

That problem has some important applications: Firstly, it is important to know M on the search problem, because if $M \geq \frac{N}{2}$ we can only sample it randomly to find the solution. Secondly, this method allows us to know if there is a solution at all for the N item search problem.

Quantum counting is an application of phase estimation to estimate the eigenvalues of the Grover operator, which enables us to find the number of solutions to the search problem.

2.12.1 Algorithm

Suppose $|a\rangle$ and $|b\rangle$ are two eigenvectors in the space spanned y $|\alpha\rangle$ and $|\beta\rangle$. Let θ be the angle of the rotation determined by the Grover operator, which is:

$$G = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad \text{with } \sin\theta = \frac{2\sqrt{M(N-M)}}{N}$$

The eigenvalues of G are $e^{i\theta}$ and $e^{i(2\pi-\theta)}$. We can assume that $M \leq \frac{N}{2}$ because we can use the augmented oracle, ensuring that $\sin^2\frac{\theta}{2} = \frac{M}{2N}$. The circuit for quantum counting is the following:

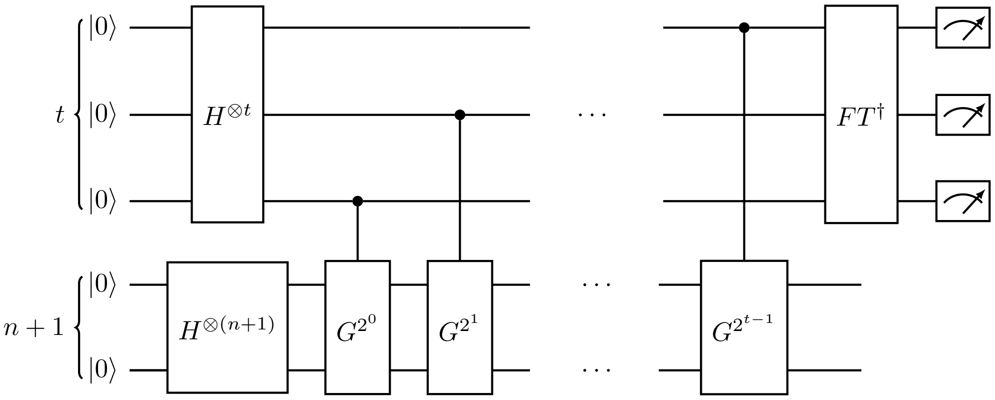


Figure 22: Quantum Counting Circuit.

The circuit estimates θ with m bits of accuracy, with a probability of success of at least $1-\epsilon$. The first register contains $t = m + \lceil \log\left(2 + \frac{1}{2\epsilon}\right) \rceil$ qubits, just as the phase estimation algorithm, and the second register has $n+1$ qubits to implement the augmented Grover operator. The second register is initialized at $|\psi\rangle = \sum_x |x\rangle$, this is a superposition of $|a\rangle$ and $|b\rangle$ thus this gives us an estimation of θ or $2\pi - \theta$ accurately with $|\Delta\theta| \leq 2^{-m}$, with probability at least $1-\epsilon$. The estimate of $2\pi - \theta$ is equivalent to θ, so we determine θ, so we certainly determine θ to an accuracy 2^{-m} with probability $1-\epsilon$.

Using that $\sin^2\left(\frac{\theta}{2}\right) = \frac{M}{2N}$ and our estimate of θ we can find M. How large is the error in M, that we will denote as ΔM?

$$\frac{|\Delta M|}{2N} = \left|\sin^2\left(\frac{\theta+\Delta\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)\right| = \left|\left(\sin\left(\frac{\theta+\Delta\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\right)\left(\sin\left(\frac{\theta+\Delta\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)\right)\right|$$

But we know that:

$$\left|\sin\left(\frac{\theta+\Delta\theta}{2}\right)+\sin\left(\frac{\theta}{2}\right)\right| \leq \frac{|\Delta\theta|}{2} \quad \text{and} \quad \left|\sin\left(\frac{\theta+\Delta\theta}{2}\right)\right| < \sin\left(\frac{\theta}{2}\right)+\frac{|\Delta\theta|}{2}$$

Then:

$$\begin{aligned}\frac{|\Delta M|}{2N} &< \left(2\overbrace{\sin\left(\frac{\theta}{2}\right)}^{=\sqrt{M/2N}}+\overbrace{\frac{|\Delta\theta|}{2}}^{\leq 2^{-m-1}}\right)\overbrace{\frac{|\Delta\theta|}{2}}^{\leq 2^{-m-1}}\\ \frac{|\Delta M|}{2N} &< \left(2\sqrt{\frac{M}{2N}}+\frac{1}{2^{m+1}}\right)\frac{1}{2^{m+1}}\end{aligned}$$

$$\boxed{\frac{|\Delta M|}{2N} < \left(2\sqrt{2MN}+N2^{-(m+1)}\right)2^{-m}} \tag{187}$$

As an example, we choose $m=\lceil \frac{n}{2}+1\rceil$ and $\epsilon=\frac{1}{6}$. Then it follows that $t=\lceil\frac{n}{2}\rceil+3$, se we will need $\Omega(\sqrt{N})$ iterations and oracle calls. Our accuracy is $|\Delta M|<\sqrt{\frac{M}{2}}+\frac{1}{4}=O(\sqrt{M})$.

This example works also to show if there is a solution on N items, that is wheter $M=0$ or $M\neq 0$. If $M=0$ we will have $|\Delta M|<\frac{1}{4}$, so the algorithm must produce the estimate within probability at least $\frac{5}{6}$. Conversely, if $M\neq 0$ then it is easy to verify that the estimate for M is not equal to 0 at least $\frac{5}{6}$ times.

Another application is to find a solution to the search problem where M is unknown because it is needed to know M in order to known how many grover iterations are needed for the Grover algorithm. Then we first estimate θ and M with high accuracy using phase estimation and then use Grover's algorithm using M to know R. The angular error is at most $\frac{\pi}{4}\left(1+\frac{|\Delta\theta|}{\theta}\right)$ so choosing $m=\lceil n/2\rceil+1$ gives an angular error of at least $\frac{\pi}{4}\frac{3}{2}$ this corresponds to a probability of at least $\cos^2\left(\frac{3\pi}{8}\right)\approx 0.15$ for the search algorithm. If the probability is just as the former case of $\frac{5}{6}$, then the probability of obtaining a solution for the search problem is $\frac{5}{6}\cos^2\left(\frac{3\pi}{8}\right)\approx 0.12$ which can be boosted by the combined counting-search procedure.

2.13 HHL Algorithm

2.13.1 Introduction

Systems of linear equations have a lot of applications in science. So if we find a speedup on solving this problem, there is real impact on society.

The problem that we want to solve is of the form:

$$Ax = b \tag{188}$$

Where A is a matrix and b is given, we want to solve for x. Classically, there are two types of solutions:

(1) **LU Decomposition**: That finds x in $O(N^{2.376}\text{poly}(log(k/\epsilon)))$ where ϵ is a bound of x error and k is the ith singular value, $k = ||A||||A^{-1}|| = \frac{\sigma_1(A)}{\sigma_N(A)}\sigma_1(A)$.

(2) **Iterative Methods**: Needs $O(\sqrt{k}\ \log(1/\epsilon))$ matrix vector multiplications. If A is spase, then we need $O(Nd\sqrt{k}\log(1/\epsilon))$ where d is the number of non-zero entries per row.

2.13.2 Quantum Algorithm

For the implementation of this algorithm is important to understand the Quantum Phase Estimation algorithm, because this is an application of this algorithm.

For the quantum algorithm we need to convert the linear equation problem for a quantum mechanical problem, to do this we represent b as quantum state:

$$|b\rangle = \sum_{i=1}^{N} b_i |i\rangle$$

and the problem is:

$$A|x\rangle = |b\rangle$$

we assume that A is an hermitian operator (Matrix), that means if A is not Hermitian we expand A in order to be hermitian:

$$C = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$$

And change the problem to be

$$Cy = \begin{pmatrix} b \\ 0 \end{pmatrix} \quad , \quad y = \begin{pmatrix} 0 \\ x \end{pmatrix}$$

The algorithm is called HHL because of the three authors of the original [paper](https://arxiv.org/abs/0811.3171): Harrow, Hassidim and Lloyd.

2.13.3 Overview of the Quantum Algorithm

The HHL consists of 3 registers, which we denote as A' for the ancilla, W for work register, and IO for Input/Output register.

The input of the algorithm is the quantum state of b, $|b\rangle$, which is input to the IO register. All other registers start on the $|0\rangle$, so we have:

$$|\psi_0\rangle \equiv |0\rangle_{A'} \otimes |0\rangle_W \otimes |b\rangle_{IO} \tag{189}$$

In addition, we also have the matrix A as the input of the algorithm. The algorithm works on three steps:

(1) Quantum Phase Estimation with the unitary $U_A \equiv e^{iAt}$ controlled by the W register and U_A applied to the IO register.

(2) Pauli-Y rotation for a particular angle θ on the A' register controlled by the W register.

(3) Do the first step in reverse, that is, doing the Inverse Quantum Phase Estimation for U_A, on the W register.

If the register A' is measured and one post-select (That is, we select only the state with the given outcome) on the $|1\rangle_{A'}$ outcome, then the state of the IO register will be close to $|x\rangle$.

2.13.4 Algorithm's Mathematical Walk-through

Since we assume that the matrix A is hermitian, then because of the spectral theorem A can be written in the eigenbasis:

$$A = \sum_j \lambda_j |u_j\rangle\langle u_j|$$

Also because A is hermitian, the operator U_A is unitary and has eigenvalues $e^{i\lambda_j t}$ and eigenstates $|u_j\rangle$. Then we apply the QPE:

$$|\psi_1\rangle = |0\rangle_{A'} \otimes \sum_j \beta_j |\overline{\lambda_j}\rangle_W \otimes |u_j\rangle_{IO}$$

Where $|\overline{\lambda_j}\rangle$ is a binary representation of λ_j up to a set precision and

$$\sum_j \beta_j |u_j\rangle_{IO}$$

is the expansion of $|b\rangle$ in the eigenbasis $|u_j\rangle$.

Applying the second step, which is the controlled-Y rotation $e^{i\theta Y}$ for the angle:

$$\theta = \arccos\frac{C}{\lambda_j}$$

Where C is an hyperparameter of the algorithm and is set by the user. After this rotation on the A' register, we have:

$$|\psi_2\rangle = \sum_j \beta_j \left(\sqrt{1 - \frac{C^2}{\overline{\lambda_j}^2}} |0\rangle_{A'} + + \frac{C}{\overline{\lambda_j}} |0\rangle_{A'} \right) \otimes |\overline{\lambda_j}\rangle_W \otimes |u_j\rangle_{IO}$$

The third step is to do de reverse QPE, this gives:

$$|\psi_3\rangle = \sum_j \beta_j \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle_{A'} + + \frac{C}{\lambda_j} |0\rangle_{A'} \right) \otimes |0\rangle_W \otimes |u_j\rangle_{IO}$$

This state is already on the inverse form, because:

$$A^{-1}|b\rangle = \sum_{j=1}^{N} \frac{\beta_j}{\lambda_j} |u_j\rangle$$

Therefore if we measure the A' register and post-select $|1\rangle_{A'}$, we have the solution:

$$|\psi_4\rangle = \sum_{j=1}^{N} \frac{\beta_j}{\lambda_j} |u_j\rangle_{IO} \approx |x\rangle$$

Thus the IO register contains the approximation of $|x\rangle$.

This solves the linear equation problem exponentially faster than a classical computer, but there's a caveat (as always), the solution vector is a quantum state, therefore for getting the classical description of x we need to do quantum state tomography (QST) but QST scales exponentially with the number of qubits loosing the exponential speedup. There are some applications that we only need an expectation value, this leads to an exponential speedup.

Note: A good value for the hyperparameter C is to be the smallest eigenvalue that can be represented by the circuit, that is

$$C = \frac{2\pi}{2^n t}$$

2.13.5 Example

Let's solve the following problem:

$$A = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix} \quad \text{and} \quad |b\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

We only need 1 qubit to represent $|b\rangle$, 2 qubits to represent the binary representation of eigenvalues and 1 qubit for the conditioned rotation, therefore we need 4 qubits total.

For demonstrations purposes let's calculate the eigenvalues of A:

$$\lambda_1 = \frac{2}{3} \quad \lambda_2 = \frac{4}{3}$$

Our QPE has only 2-bit precision for $\frac{\lambda_j t}{2\pi}$. Therefore we can set t:

$$t = 2\pi \frac{3}{8}$$

This will give a 2-bit approximation to:

$$\frac{\lambda_1 t}{2\pi} = \underbrace{\frac{1}{4}}_{|01\rangle} \qquad \frac{\lambda_2 t}{2\pi} = \underbrace{\frac{1}{2}}_{|10\rangle}$$

The eigenvectors are:

$$|u_1\rangle = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \qquad |u_2\rangle = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Writing $|b\rangle$ in the eigenbasis of A:

$$|b\rangle_{\text{IO}} = \sum_{j=0}^{N-1} \frac{1}{\sqrt{2}} |10\rangle |u_j\rangle$$

Let's go through the steps of the algorithm:

1. The state preparation in this example is $|b\rangle = |0\rangle$.

2. Applying QPE will yield:

$$\frac{1}{\sqrt{2}} |01\rangle |u_1\rangle + \frac{1}{\sqrt{2}} |10\rangle |u_2\rangle$$

3. Conditioned rotation with $C = \frac{3}{8}$:

$$\begin{aligned} &\frac{1}{\sqrt{2}} |01\rangle |u_1\rangle \left(\sqrt{1 - \frac{(3/8)^2}{(1/4)^2}} |0\rangle + \frac{3/8}{1/4} |1\rangle \right) + \frac{1}{\sqrt{2}} |10\rangle |u_2\rangle \left(\sqrt{1 - \frac{(3/8)^2}{(1/2)^2}} |0\rangle + \frac{3/8}{1/2} |1\rangle \right) \\ &= \frac{1}{\sqrt{2}} |01\rangle |u_1\rangle \left(\sqrt{1 - \frac{9}{4}} |0\rangle + \frac{3}{2} |1\rangle \right) + \frac{1}{\sqrt{2}} |10\rangle |u_2\rangle \left(\sqrt{1 - \frac{9}{16}} |0\rangle + \frac{3}{4} |1\rangle \right) \end{aligned}$$

4. After applying QPE†:

$$\frac{1}{\sqrt{2}} |00\rangle |u_1\rangle \left(\sqrt{1 - \frac{9}{4}} |0\rangle + \frac{3}{2} |1\rangle \right) + \frac{1}{\sqrt{2}} |00\rangle |u_2\rangle \left(\sqrt{1 - \frac{9}{16}} |0\rangle + \frac{3}{4} |1\rangle \right)$$

5. Post-selecting 1 on the auxiliary qubit, we have:

$$\frac{\frac{3}{2\sqrt{2}} |00\rangle |u_1\rangle |1\rangle + \frac{3}{4\sqrt{2}} |00\rangle |u_2\rangle |1\rangle}{\sqrt{\frac{45}{32}}} = \frac{|x\rangle}{||x||}$$

2.14 Variational Quantum Eigensolver

Variational Quantum Eigensolver (VQE) is a hybrid Quantum/Classical algorithm that can be used to find the minimum of any objective function that we can express as a quantum circuit, that is unitary. This method was proposed by Peruzo et al. [15] in 2013 and there's a good article explaining the use of VQE by Moll et al [12], the main idea of the algorithm is to find the minimum eigenvalue of a given matrix using the variational principle of quantum mechanics by calculating the expected values of Pauli strings of a Hamiltonian that can be written as a polynomial number of terms.

In variational methods we start with an ansatz for the ground state (state associated with the minimum eigenvalue). We parametrized the initial ansatz by a set of parameters θ, then we have our initial quantum state: $|\psi(\theta)\rangle$. We want to solve the following problem:

Given an Hamiltonian $\mathcal{H}$ we want to approximate the ground state energy by solving the following optimization problem:

$$\boxed{\min_{\theta} \langle\psi(\theta)| \mathcal{H} |\psi(\theta)\rangle} \tag{190}$$

Using the variational principle, we have that:

$$\lambda_\theta = \langle\psi(\theta)| \mathcal{H} |\psi(\theta)\rangle \geq \lambda_{\min} = E_{gs} \tag{191}$$

So by minimizing this value we get an approximation of the minimum eigenvalue of the Hamiltonian.

2.14.1 Variational Method

Consider the eigenvector $|\psi_i\rangle$ of a matrix $\mathcal{H}$ which is invariant under the transformation $\mathcal{H}$ up to a constant (the eigenvalue of $|\psi_i\rangle$). Therefore: $\mathcal{H} |\psi_i\rangle = \lambda_i |\psi_i\rangle$.

Now let's consider that the matrix $\mathcal{H}$ is Hermitian ($\mathcal{H} = \mathcal{H}^\dagger$), then all eigenvalues are real and $\mathcal{H}$ can be written as:

$$H = \sum_{i=1}^{N} \lambda_i |\psi_i\rangle\langle\psi_i|$$

The expectation value of $\mathcal{H}$ on an arbitrary state $|\psi\rangle$ is given by:

$$\langle H\rangle_\psi \equiv \langle\psi| \mathcal{H} |\psi\rangle \tag{192}$$

Thus we may write:

$$\begin{aligned}\langle H\rangle_\psi &= \langle\psi| \mathcal{H} |\psi\rangle = \langle\psi| \sum_{i=1}^{N} \lambda_i |\psi_i\rangle\langle\psi_i| |\psi\rangle \\ &= \sum_{i=1}^{N} \lambda_i \langle\psi|\psi_i\rangle \langle\psi_i|\psi\rangle = \sum_{i=1}^{N} \lambda_i |\langle\psi|\psi_i\rangle|^2\end{aligned}$$

This equation shows that the expectation value of an observable on any state can be written as a weighted sum of the eigenvalues associated with $\mathcal{H}$. Since $| \langle\psi|\psi_i\rangle |^2 \geq 0$:

$$\boxed{\lambda_{\min} \leq \langle H\rangle_\psi = \langle\psi| \mathcal{H} |\psi\rangle = \sum_{i=1}^{N} \lambda_i | \langle\psi|\psi_i\rangle |^2} \tag{193}$$

Thus the expectation value of any wave function will always be at least the minimum eigenvalue associated with $\mathcal{H}$. This method thus is usually used to estimate the ground state of an Hamiltonian.

2.14.2 Algorithm

The algorithm consists in 4 steps, which will be explained in length in the following subsections:

1. Construct a variational circuit with a desired ansatz, thus creating $|\psi(\theta)\rangle$.

2. Decompose the problem Hamiltonian into Pauli strings of polynomial size: $\mathcal{H} = \sum_\alpha h_\alpha P_\alpha$, where: $P_\alpha = \sigma^{\alpha_1} \otimes \ldots \sigma^{\alpha_N}$.

3. Evaluate the problem Hamiltonian using the chosen ansatz: $\langle H\rangle(\theta) = \langle\psi(\theta)| \mathcal{H} |\psi(\theta)\rangle = \sum_\alpha h_\alpha \langle\psi(\theta)| P_\alpha |\psi(\theta)\rangle$

4. Using the value of the first step, minimize the expectation value with respect to the parameter θ using a classical optimizer.

2.14.3 Variational Forms

It is needed to construct a quantum circuit for the ansatz, for this we will use parametrized quantum circuits with a fixed form which will help us to explore a part of the Hilbert space that (hopefully) is useful for our problem. Such a circuit is often called *Variational Form* and is represented by the unitary operator $U(\theta)$. A variational form is applied to the initial state $|\psi\rangle$ and generates the output state: $U(\theta)|\psi\rangle = |\psi(\theta)\rangle$. We try to optimize this ansatz to yield $\langle\psi(\theta)| \mathcal{H} |\psi(\theta)\rangle \approx E_{gs} = \lambda_{min}$.

A fixed variational form with a polynomial number of parameters (i.e. efficient quantum circuit) can only generate transformations on to a polynomially sized subspace of all the states in a exponentially sized Hilbert space. Thus we need to create variational forms depending on our problem at hand and the variational form has to satisfy two conflicting goals:

- We want to generate any state $|\psi\rangle \in \mathbb{C}^{2^n}$.

- We want to use as few parameters as possible and of polynomial size.

In order to see how this problem can conflict, let's consider the case for 1 qubit, thus we want to generate any state $|\psi\rangle \in \mathbb{C}^2$. This can be done by the U3 gate which takes exactly 3 parameters, up to a phase:

$$U3(\theta, \phi, \lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i\lambda+i\phi}\cos\frac{\theta}{2} \end{pmatrix}$$

Since this variational form can generate any state, the evaluation of the expected value of the Hamiltonian is limited only by the capabilities of the classical optimizer. This example illustrates that for only one qubit, we need 3 parameters to generate an arbitrary state, the problem of generating a arbitrary state is indeed NP-Hard and the number of parameters grows exponentially for the numbers of qubits, thus we need to restrain our state creation for sub-spaces that are polynomially generated.

Since we need variational forms that are parametrized polynomially and such forms can't create any state, thus we need to construct categories of variational forms that are described by a specific application. There are two categories:

- Domain Specific: Uses domain knowledge in order to construct variational circuits.
- Hardware Efficient: Uses ansatze that the connections are efficient for a given hardware.

Domain Specific For these kinds of ansatze we exploit the characteristics of the problem which we have prior knowledge. For example, suppose we want to calculate the ground state of a molecule, then we know a priori how many particles are on the circuit. Thus we limit the variational for that only produces particle preserving transformations reducing the number of parameters needed. One of those ansatze is the Unitary Coupled-Cluster (UCC) approach, which is an adaptation of the commonly used Coupled-Cluster method. We thus use a unitary operator of the form $U(\theta)$ into a trial state $|\phi\rangle$:

$$\boxed{|\psi(\theta)\rangle = U(\theta)\,|\phi\rangle = e^{T(\theta)-T^{\dagger}(\theta)}\,|\phi\rangle} \tag{194}$$

This state is constructed by the hamiltonian simulation of the cluster operator $T(\theta)$ defined as $T(\theta) = \sum_k T_k(\theta)$, where:

$$T_1(\theta) = \sum_{\substack{i\in\text{occ}\\ j\in\text{unocc}}} \theta_{(i)}^{(j)} a_j^{\dagger} a_i$$

$$T_2(\theta) = \sum_{\substack{i,j\in\text{occ}\\ k,l\in\text{unocc}}} \theta_{(i,j)}^{(k,l)} a_k^{\dagger} a_l^{\dagger} a_i a_j$$

And so on, it is important to observe that T_k is given by the number of k-body terms of your approximation. For the trial state, it is common to choose the ground-state of the solution of the Hartree-Fock equation, which can be obtained efficiently. The expansion is commonly truncated to second order, giving the UCCSD ansatz. Unfortunately, this ansatz doesn't satisfy the depth problem of current devices, but could be good for fault-tolerant quantum computers in the future.

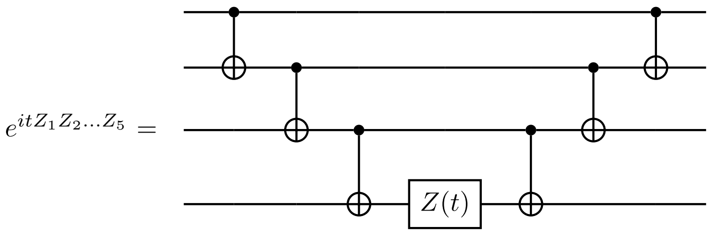


Figure 23: Example of UCCSD circuit.

Hardware Efficient For this kind of ansatze we choose gates and coupling strategies that are efficient for the device that we are working on, this fixes the depth problem from the domain specific ansatze, making us control the depth that we are working on. Two common ones are $R_y R_z$ and R_y followed by an entangling block that is commonly used by CNOT gates. Thus the ansatz is of the form:

$$\boxed{\big|\psi(\theta) = U^N(\theta)\; U_{\text{ENT}} \ldots\; U^1(\theta)\; U_{\text{ENT}}\; U^0(\theta)\,|0\rangle\big\rangle} \tag{195}$$

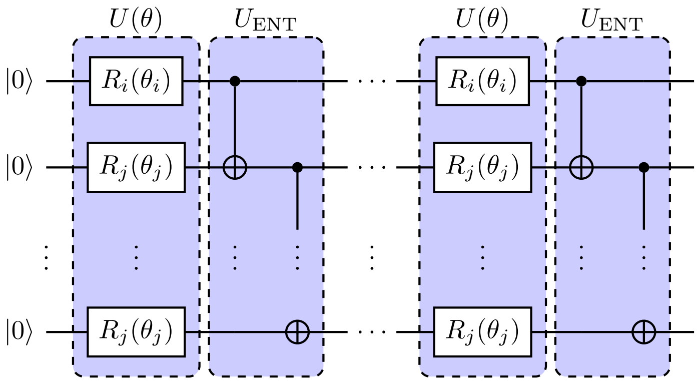


Figure 24: Example of Hardware Efficient variational circuits.

2.14.4 Parameter Optimization

After selecting the ansatz we should optimize it according to a cost function that is a sum o Pauli strings. This rise all sort of problems as the quantum hardware has inherent noise and may not reflect the actual cost function [19], another problem is to choose a suited optimizer for the specific application. It is empirically stated that the Simultaneous Pertubation Stochastic Approximation (SPSA) optimizer is a good optimizer for circuits with noise, but this is still an area of active research. SPSA approximates the gradient of the objective function with only two measurements. This is recommended on a noisy simulation or on real hardware because we do not need many measurements samples which the cost is high.

2.14.5 Creating the Objective Function

It is possible to decompose a Hermitian matrix into Pauli components (or Pauli strings, as sometimes is called), for instance for a 2^Nx2^N matrix A, we have:

$$A = \frac{1}{N}\sum_{i_1,\ldots,i_n} h_{i_1,\ldots,i_n}\sigma_{i_1}\otimes\cdots\otimes\sigma_{i_n} \qquad \text{where :}\quad h_{i_1,\ldots,i_n} = \text{Tr}\big((\sigma_{i_1}\otimes\cdots\otimes\sigma_{i_n})\cdot A\big)$$

After decomposing the matrix into Pauli components we need to measure them on a real device, but one limitation of current quantum computers is that we need to measure in the Z basis, thus in order to measure in the X or Y basis we need to change the basis using H or $HS^\dagger$ gates. Let's consider how to convert counts into expectation values, consider $|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$:

$$\begin{aligned} &Z_1Z_2\,|\psi\rangle = \alpha\,|00\rangle - \beta\,|01\rangle + \gamma\,|10\rangle - \delta\,|11\rangle \\ &\qquad \Rightarrow \langle\psi|\,Z_1Z_2\,|\psi\rangle = \alpha^2 - \beta^2 - \gamma^2 + \delta^2 = P_{00} - P_{01} - P_{10} + P_{11} \end{aligned} \tag{196}$$

What about if we have X (or Y) expectations?

$$\langle\psi| Z_1X_2 |\psi\rangle = \langle\psi| Z_1U_2^\dagger Z_2U_2 |\psi\rangle = \big(\langle\psi| U_2^\dagger\big)Z_1Z_2\big(U_2 |\psi\rangle\big)$$

For X, we have: $X = HZH$ and for Y, we have: $Y = (S^\dagger H)^\dagger Z(S^\dagger H)$, thus we only need to add those gates at the end of each circuit and measure the Z expectation.

Another thing you can do to calculate Pauli strings more efficiently is to use Pauli grouping, that is group Pauli strings with the same basis, for instance consider the Hamiltonian $H = Z_1Z_2 + I_1Z_2$, we only need one measurement:

$$\langle\psi| I_1Z_2 |\psi\rangle = P_{00} - P_{01} + P_{10} - P_{11}$$

which P_{ij} we already know from the Z_1Z_2 measurement, thus only need to rearrange the counts to get the I_1Z_2 or Z_1I_2 measurements.

2.14.6 Chemistry with qubits

To simulate systems with continuous variables in quantum computers we need to truncate them. The most direct route is to define a finite set of basis functions and then to project the exact many-body Hamiltonian onto the chosen basis. The resulting discretized system is then expressed in terms of qubits. Finally, depending on the particles involved it may also be necessary to account for their bosonic or fermionic nature. A choice of a good representation is vital as it may affect the simulation cost drastically.

Electronic Structure qubit representations The main objective of electronic structure is to understand the low-energy properties of the electronic structure Hamiltonians that describes a system of interacting electrons moving in the potential created by atomic nuclei, consider the Hamiltonian after the Born-Oppenhaimer Approximation, which considers the nuclei fixed:

$$\mathcal{H} = \overbrace{\sum_{i=1}^{K} -\frac{1}{2}\nabla_i^2 + V(r_i)}^{\mathcal{H}_1} + \overbrace{\sum_{1\le i\le j\le K} \frac{1}{|r_i - r_j|}}^{\mathcal{H}_2} \tag{197}$$

Here K is the number of electrons, r_i is the position operator, $\mathcal{H}_1$ is the kinetic energy of non-interacting electron and the potential energy of non-interacting electrons, while $\mathcal{H}_2$ is the Coulomb repulsion between two electrons.

Each electron is described by the position $r_i \in \mathbb{R}^3$ and spin $\omega_i \in \{\uparrow, \downarrow\}$. The wavefunction of K electrons is $\psi(x_1, \ldots, x_K)$, $x_i = (r_i, \omega_i)$ and must be antisymmetric under exchange of x_i. Electronic structure simulation algorithms based on the first quantization method describe a system of K lectrons using the configuration interaction. It is convenient to choose the Slater determinant as a basis for this space:

$$\Psi(x_1,\dots x_K) \propto \det \begin{pmatrix} \psi_1(x_1) & \psi_1(x_2) & \dots & \psi_1(x_k) \\ \vdots & \vdots & \vdots & \vdots \\ \psi_K(x_1) & \psi_K(x_2) & \dots & \psi_K(x_k) \end{pmatrix}$$

Where $\{\psi_i(x_i)\}_{i=1}^N$ is a set of orthonormal spin-orbitals. The set of Slater determinants spanning the Configuration Interaction space $H_{K,N}$ is formed by distributing K electrons over N one electron spin-orbitals in all possible ways, thus the dimension is $\binom{N}{K}$ and can be identified with the anti-symmetric subspace of $\left(\mathbb{C}^N\right)^{\otimes K}$.

The projection of the full electronic hamiltonian onto Configuration Interaction space has the form:

$$\mathcal{H} = \sum_{i=1}^{K} \sum_{p,q=1}^{N} t_{pq} \langle p|q\rangle_i + \sum_{1\leq i\neq j\leq K} \sum_{p,q,r,s=1}^{N} u_{pqrs} \langle p|r\rangle_i \otimes \langle q|s\rangle_j \tag{198}$$

The coefficients t_{pq} and u_{pqrs} are known as one- and two-body integrals:

$$t_{pq} = \langle p| \left(-\frac{\nabla^2}{2} + V \right) |q\rangle$$

$$u_{pqrs} = \langle p, q| \left(\frac{1}{|r_i - r_j|} \right) |r, s\rangle$$

Which can be calculated efficiently on classical computers.

Each copy of the single-electron Hilbert space $\mathbb{C}^N$ is then encoded by a register of $\log_2 N$ qubits. This requires $n = K\log_2 N$ qubits in total. The CI Hamiltonian H includes multi-qubit interactions among subset of $2\log_2 N$ qubits.

Second-Quantization The second-quantization approach often results in a simpler simulation Hamiltonian and require fewer qubits, especially in the case where the filling fraction $\frac{K}{N}$ is not small. Given a set of N orbitals $\psi_1,\dots,\psi_N$ the second-quantized simulation Hamiltonian is:

$$\boxed{\mathcal{H} = \sum_{p,q=1}^{N} t_{pq} c_p^\dagger c_q + \sum_{p,q,r,s=1}^{N} u_{pqrs} c_p^\dagger c_q^\dagger c_r c_s} \tag{199}$$

Where $c_p^\dagger$ and c_p are creation and annihilation operators for the orbital ψ_p. Now the Hamiltonian acts on the Fock space $\mathcal{F} = \bigoplus_{k=0}^N \mathcal{H}_{k,N}$ of an arbitrary number of fermions in N spin-orbitals.

The Fock space is spanned by 2^N basis vectors $|n_1 \dots n_N\rangle$, where $n_p \in \{0,1\}$ is the occupation number of the orbital ψ_p. The definition of creation/annihilation operators is the following:

$$\begin{aligned} c_i^\dagger |\dots n_i\ \dots\rangle &= (1-n_i)\ (-1)^{\sum_{j<i} n_j} |\dots\ n_i + 1\ \dots\rangle \\ c_i |\dots n_i\ \dots\rangle &= n_i\ (-1)^{\sum_{j<i} n_j} |\dots\ n_i - 1\ \dots\rangle \\ &\Rightarrow\ \{c_i^\dagger, c_j\} = \delta_{ij}\ ,\ [c_i, c_j] = [c_i^\dagger, c_j^\dagger] = 0 \end{aligned} \tag{200}$$

The advantage of second-quantization is that the Fermi statistics is automatically enforced at the operator level and the simulation has to be restricted to the subspace with exactly K occupied orbitals. The second quantization can be written in terms of qubits using one of the Fermion-to-qubit mappings.

Fermion-to-qubit mapping Since we are working with qubits and not fermions, we should have a 1-on-1 mapping between fermions and qubits. Let's begin with the case for only one fermion:

$$c^\dagger |0\rangle = |1\rangle \qquad c^\dagger |1\rangle = 0$$
$$c |0\rangle = 0 \qquad c |1\rangle = |0\rangle$$

Thus, we can do the mapping:

$$c^\dagger \rightarrow \sigma^+ \qquad\qquad c \rightarrow \sigma^-$$

Now we can generalize for N fermions, using equation 200 we know that to change from n_i to $n_i + 1$ we use σ_i^+, how we transform $(-1)^{\sum_{j<i} n_j}$ into gates?

It's rather easy, because this is just a σ^Z gate on each qubit that precedes n_i, because: $\sigma^Z |n_i\rangle = (-1)^{n_i} |n_i\rangle$. Thus we have the following mapping:

$$\boxed{c_i^\dagger \rightarrow \prod_{j<i} \sigma_j^Z \sigma_i^+ \qquad\qquad c_i \rightarrow \prod_{j<i} \sigma_j^Z \sigma_i^-} \tag{201}$$

This is called Jordan-Wigner Mapping and was discovered in the 20s, and it recovers fermionic algebra. There's a inherent problem that for increasing i, this operator becomes increasingly non-local with O(N)-local interactions, this leads to larger circuits and more measurements.

Another mapping is the parity mapping, which is still O(N)-local but simplify the number of qubits needed, let's see an example:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a+b+c+d \\ b+c+d \\ c+d \\ d \end{pmatrix}$$

Another one is the Bravyi-Kitaev [8] map which is O(log N)-local and uses a binary tree rule to encode fermionic modes:

$$\beta_1 = (1) \qquad \rightarrow \qquad \beta_{2^{x+1}} = \left(\begin{array}{c|ccc} & 1 & 1 & \dots \\ \beta_{2^x} & 0 & 0 & \dots \\ & 0 & 0 & \dots \\ \hline 0 & & \beta_{2^x} & \end{array}\right)$$

Let's see some examples:

- 1 qubit:

$$\begin{pmatrix} 1 \end{pmatrix}$$

- 2 qubits:

$$\left(\begin{array}{c|c} 1 & 1 \\ \hline 0 & 1 \end{array}\right)$$

- 4 qubits:

$$\left(\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

The creation/annihilation operators mappings are:

$$\begin{aligned} c_j^\dagger &\equiv X_{U(j)} \otimes \Pi_j^+ \otimes Z_{P(j)} = \frac{1}{2}\Big(X_{U(j)} \otimes X_j \otimes Z_{P(j)} - iX_{U(j)} \otimes Y_j \otimes Z_{P(j)}\Big) \\ c_j &\equiv X_{U(j)} \otimes \Pi_j^- \otimes Z_{P(j)} = \frac{1}{2}\Big(X_{U(j)} \otimes X_j \otimes Z_{P(j)} + iX_{U(j)} \otimes Y_j \otimes Z_{P(j)}\Big) \end{aligned} \tag{202}$$

Where we have the following set of states:

- Update Set $[U(j)]$: Qubits that are flipped when node j change occupation;
- Parity Set $[P(j)]$: Set of qubits that encodes the parity of the fermionic modes with index less than j.

It is important to note that we can calculate these encodings efficiently using a classical computer.

Reducing further the number of qubits Another way to reduce the number of qubits is to taper symmetries [7]. Suppose we have the Hamiltonian:

$$H = h_1 \ IXIY \ + \ h_2 \ XZZX \ + \ h_3 XZYZ$$

We call the number of non-identity Pauli strings on a single qubit operator as the weight of the Pauli string. The larger the weight larger the error that is associated to that Pauli string. Now consider the following operator:

$$P_0 = XIII \qquad\qquad \Rightarrow [H, P_0] = 0$$

and we also know the eigenvalues of P_0 (which are ±1), since this is a symmetry of the Hamiltonian we can substitute by the eigenvalues and drop this qubit on the operator:

$$H = h_1 \; XIY \;\pm\; h_2 \; ZZX \;\pm\; h_3 ZYZ$$

2.15 Quantum Approximate Optimization Algorithm

2.16 Variational Quantum Thermalizer

2.16.1 Introduction

In this section we will focus on Quantum Hamiltonian-Based models, which is a generalization of Energy-Based models in Deep Learning to the Quantum realm. This is based on the paper [18]. The general algorithm is depicted in 25 The variational latent distribution ρ_θ have parameters θ and is constructed in a way that is easy to make in a Quantum Computer. Quantum correlations are incorporated using $U(\phi)$ with parameters ϕ.

Thus the mixed state is given by:

$$\rho_{\theta\phi} = U(\phi)\rho_\theta U^\dagger(\phi)$$

Which we will call the visible state.

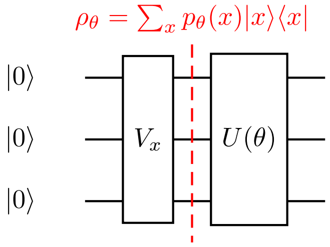


Figure 25: Variational Quantum Thermalizer circuit.

Without loss of generality, one can consider the altent space to be a thermal state of a given parametrized Hamiltonian:

$$\rho_\theta = \frac{1}{Z_\theta}e^{-K_\theta} \qquad Z_\theta = tr(e^{-K_\theta})$$

We will call K_θ the *latent modular Hamiltonian*, and Z_θ is *the model partition function*. For such a latent space, we have that the visible state is given by:

$$\rho_{\theta\phi} = \frac{1}{Z_\theta}e^{-U(\phi)K_\theta U^\dagger(\phi)} \equiv \frac{1}{Z_\theta}e^{-K_{\theta\phi}}$$

Where the *model modular Hamiltonian* is defined as $K_{\theta\phi}$.

In order to train this model, we have to define the Von Neumann Entropy, which is invariant under unitary action:

$$S(\rho_\theta) = -tr(\rho_\theta \log \rho_\theta)$$

Using the invariance, we have:

$$S(\rho_\theta) = S(U(\phi)\rho_\theta U^\dagger(\phi)) = S(\rho_{\theta\phi})$$

Thus estimating the entropy in the visible state is the same as the latent space!

2.16.2 Algorithm

Problem: Given a Hamiltonian H and a target inverse temperature $\beta \equiv \frac{1}{T}$, we wish to generate an approximation to the Thermal State:

$$\sigma_\beta = \frac{1}{Z_\beta} e^{-\beta H} \quad Z_\beta = tr(e^{-\beta H})$$

Where Z_β is the *Thermal partition function.* The strategy is to convert this problem to a quantum-probabilistic variational learning task. Let's consider an ansatz for the thermal state, $\rho_{\theta\phi}$, let's analyse the relative entropy between this unknown thermal state and our ansatz:

$$\begin{aligned} D(\rho_{\theta\phi}||\sigma_\beta) &= -S(\rho_{\theta\phi}) - tr(\rho_{\theta\phi} \log \sigma_\beta) \\ &= -S(\rho_{\theta\phi}) - \beta tr(\rho_{\theta\phi} H) + \log Z_\beta \end{aligned}$$

This is the *quantum relative free energy* of our model with respect to the target Hamiltonian H. This is called *relative free energy*, because it is the difference of the free energy $F(\nu) = tr(\nu H) - \frac{1}{\beta} S(\nu)$, up to a factor β, of the thermal state and the ansatz:

$$D(\rho_{\theta\phi}||\sigma_\beta) = \beta F(\rho_{\theta\phi}) - \beta F(\sigma_\beta)$$

For the variational quantum algorithm is important to note that the positivity of the free energy implies that the minimum of the free energy is achieved by the Thermal state:

$$D(\rho_{\theta\phi}||\sigma_\beta) = 0 \quad \Rightarrow F(\rho_{\theta\phi}) = F(\sigma_\beta) \text{ and } \rho_{\theta\phi} = \sigma_\beta$$

Therefore, our goal is to minimize the free energy as the loss function:

$$\mathcal{L}_{VQT} = \beta F(\rho_{\theta\phi}) = \beta tr(\rho_{\theta\phi} H) - S(\rho_{\theta\phi})$$

and find optimal parameters $\{\theta^*, \phi^*\}$, such that $\rho_{\theta\phi} = \sigma_\beta$.

The great advantage of the QHBM structure (fig 25) is that the entropy of the variational model distribution is equal to the latent distribution: $S(\rho_{\theta\phi}) = S(\rho_\theta)$. Thus the only evaluation that needs to be done in a quantum computer is the **energy expectation**, therefore having the same cost as the Variational Quantum Eigensolver.

Furthermore, the VQE is a limiting case of the VQT when $t \to 0$:

$$\tilde{\mathcal{L}} = \mathcal{L}_{VQT} = F(\rho_{\theta\phi}) = tr(\rho_{\theta\phi} H) - \frac{1}{\beta} S(\rho_{\theta\phi})$$

$$\boxed{\lim_{\beta \to \infty} \tilde{\mathcal{L}} = tr(\rho_{\theta\phi} H) = \langle H \rangle_{\theta\phi}} \tag{203}$$

The VQT is the natural generalization of the VQE for non-zero temperature states.

2.16.3 Structure of the Latent Space

In [18] they consider a good choice of latent space when it is "quantum simple", i.e. low complexity for quantum computers to prepare. Let's consider the most simple latent space model which is the *Factorized latent Space.*

Factorized Latent Space Model: Let's choose a quantum system that is separated into N smaller dimensional subsystems:

$$\rho_\theta = \bigotimes_{j=1}^{N} \rho_j(\theta_j)$$

Using this latent space we have that the latent modular Hamiltonian becomes a sum of the subsystems' Hamiltonians:

$$K_\theta = \sum_j K_j(\theta_j) \quad , \quad \rho_\theta = \bigotimes_{j=1}^{N} \frac{1}{\theta_j} e^{-K_j(\theta_j)} \quad , \quad Z_{\theta_j} = tr[e^{-K_j(\theta_j)}]$$

When estimating the expectation value of the modular Hamiltonian it becomes the sum of the subsystems' modular Hamiltonians:

$$\langle K_\theta \rangle = \sum_j \langle K_j(\theta_j) \rangle$$

The partition functions is the product of the subsystems' partition functions $Z_\theta = \prod_j Z_{\theta_j}$ and the log becomes $\log Z_\theta = \sum_j \log Z_{\theta_j}$ so does the entropy $S(\rho_\theta) = \sum_j S(\rho_j(\theta_j))$ which is much easier to calculate than a non-factorizable state.

Another feature is that the number of parameters scales linearly with the number of the subsystems N. By learning a completely decorrelated representation in the latent space, we are learning a representation which has a natural orthogonal basis for its latent space.

Examples

2.1) Qudit Operators: The first class are qudits operators diagonal in the computational basis. The more general qudit representation is given by:

$$K_j(\theta_j) = \sum_{k=1}^{d_j} \theta_{jk} |k\rangle\langle k|_j$$

Where the eigenvalues of this Hamiltonian form a set of variational parameters of the distribution. The latent distribution is similar to a softmax function.

2.2) Continuous-Variable quantum node (qunode): It is a Harmonic Oscillator, which the exponential distribution becomes a single-mode thermal state:

$$K_j(\theta_j) = \frac{\theta_j}{2}(x^2 + p^2) = \theta_j(a_j^\dagger a_j + \frac{1}{2})$$

This is the closest thing that one can have of a latent product of Gaussians which are of common use in Deep Learning.

2.3) Particle number of fermions: it is similar to qumodes but for Fermions:

$$K_j(\theta_j) = \theta_j c_j^\dagger c_j$$

3 Quantum Hardware

> “Quantum phenomena do not occur in a Hilbert space. They occur in a laboratory.”
>
> Asher Peres, Quantum Theory: Concepts and Methods

Now we will focus on the physics of superconducting qubits.

3.1 Transmon Physics

In order to have a qubit we need a two-level system. The first type of superconducting qubits discovered was the Cooper Pair Box. But the better suited superconducting qubit is the Transmon, because it fixes the charge noise problem on the cooper pair box, this is done by an the anharmonicity of a Josephson Junction.

To understand the physics of Transmon qubits firstly we need to find the Hamiltonian to quantize a classical circuit, by comparing classical poisson brackets to quantum commutators.

3.1.1 Hamiltonian of a Quantum Circuit

Let's consider the simple example of a LC circuit:

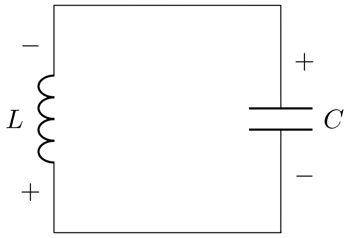


Figure 26: LC Circuit.

We will first find the Lagrangean and then do a Legendre transform to obtain the Hamiltonian. For finding the Lagrangean, we use the Branch-Flux Method. We the define the flux and charge as follows:

$$\Phi(t) = \int_{-\infty}^{t} V(t')dt' \quad Q(t) = \int_{-\infty}^{t} I(t')dt' \tag{204}$$

We will use the flux Φ as our generalized coordinate. The instantaneous energy across the circuit at a time t is:

$$E(t) = \int_{-\infty}^{t} V(t')I(t')dt' \tag{205}$$

We know that the voltage across a capacitor with capacitance C is $I = C\frac{dV}{dt}$ and for an inductor with inductance L is: $V = L\frac{dI}{dt}$. The potential energy in the inductor is:

$$U_L(t) = \int_{-\infty}^{t} L\frac{dI(t')}{dt}I(t')dt' = \int_{-\infty}^{t} LdI(t')I(t') = \frac{1}{2}L[I(t)]^2 \tag{206}$$

Since $\Phi(t) = \int_{-\infty}^{t} V(t')dt' = \int_{-\infty}^{t} \frac{dI(t')}{dt'}dt' = LI(t)$, we have that:

$$U_L(t) = \frac{1}{2L}\Phi^2 \tag{207}$$

Similarly, voltage is the rate of change of flux, so it corresponds to the kinectic energy:

$$K_C(t) = \int_{-\infty}^{t} C\frac{dV(t')}{dt'}V(t') = \frac{1}{2}C[V(t)]^2 = \frac{1}{2}C\dot{\Phi}^2 \tag{208}$$

Then we can construct the Lagrangean since it is the diference between Kinectic and Potential energy:

$$\mathcal{L} = K - U = \frac{1}{2L}\Phi^2 - \frac{1}{2}C\dot{\Phi}^2 \tag{209}$$

The dynamics is given by the Euler-Lagrange Equation:

$$\frac{\partial \mathcal{L}}{\partial \Phi} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\Phi}}\right) = 0 \Rightarrow \frac{\Phi}{L} + C\ddot{\Phi} = 0 \tag{210}$$

Thus the dynamical equation is:

$$\ddot{\Phi} = \omega^2 \Phi \tag{211}$$

This is a harmonic oscillator in Φ with frequency $\omega = \frac{1}{\sqrt{LC}}$. Since we want to get the Hamiltonian, we get the conjulgate variable of the flux, Q:

$$Q \equiv \frac{\partial \mathcal{L}}{\partial \Phi} = C\dot{\Phi} = CV \tag{212}$$

Now we obtain the Hamiltonian by using a Legendre transform:

$$\mathcal{H} = Q\Phi - \mathcal{L} = \frac{Q^2}{2C} + \frac{\Phi^2}{2L} \tag{213}$$

3.1.2 Quantizing a circuit Hamiltonian

We will get the quantum harmonic oscillator when we quantize the LC circuit by changing conjulgate variables to operators $Q \to \hat{Q}$, $\Phi \to \hat{\Phi}$:

$$\hat{\mathcal{H}} = \frac{\hat{Q}}{2C} + \frac{\hat{\Phi}}{2L} \tag{214}$$

After this we associate Poisson brackets into commutation relations:

$$\{A, B\} = \frac{\delta A}{\delta \Phi}\frac{\delta B}{\delta Q} - \frac{\delta A}{\delta Q}\frac{\delta B}{\delta \Phi} \;\Leftrightarrow\; \frac{1}{i\hbar}[\hat{A}, \hat{B}] = \hat{A}\hat{B} - \hat{B}\hat{A} \tag{215}$$

For our variables:

$$\{\Phi, Q\} = 1 \Rightarrow [\hat{\Phi}, \hat{Q}] = i\hbar \tag{216}$$

We can rewerite the hamiltonian using the reduced charge $\hat{n} = \frac{\hat{Q}}{2e}$ and phase $\hat{\phi} = \frac{2\pi\Phi}{\Phi_0}$, were $\Phi_0 = \frac{h}{2e}$:

$$\hat{H} = 4E_C\hat{n}^2 + \frac{1}{2}E_L\hat{\phi}^2 \tag{217}$$

Where $E_C = \frac{e^2}{2C}$ is the charging energy and $E_L = \left(\frac{\Phi_0}{2\pi}\right)^2 \frac{1}{L}$ is the inductive energy. Now we can define creation and annihilation operators in terms of zero-point fluctuations of the charge and phase:

$$\hat{n} = i\ n_{ZPF}(\hat{a} + \hat{a}^\dagger) \qquad , \quad n_{ZPF} = \left(\frac{E_L}{32E_C}\right)^{\frac{1}{4}} \tag{218}$$

$$\hat{\phi} = \phi_{ZPF}(\hat{a} - \hat{a}^\dagger) \qquad , \quad \phi_{ZPF} = \left(\frac{2E_C}{E_L}\right)^{\frac{1}{4}} \tag{219}$$

Thus we have the quantum harmonic oscillator hamiltonian:

$$\mathcal{H} = \hbar\omega\left(a^\dagger a + \frac{1}{2}\right) \tag{220}$$

with $\omega = \frac{\sqrt{8E_L E_C}}{\hbar} = \frac{1}{\sqrt{LC}}$.

3.1.3 Transmon Hamiltonian

All this work was done for the quantization of LC circuits, for the Transmon circuit we need to change the inductor for a josephson junction which has the following current-flux relation:

$$I = I_0 \sin\left(\frac{2\pi\Phi}{\Phi_0}\right) \tag{221}$$

Using Kirchoff's law, just as before, we have the following equation of motion:

$$I_0 \sin\left(\frac{2\pi\Phi}{\Phi_0}\right) + C\ddot{\Phi} = 0 \tag{222}$$

Now we need to convert this equation of motion into a Lagrangian, which is:

$$\mathcal{L} = \frac{I_0\Phi_0}{2\pi}\cos\left(\frac{2\pi\Phi}{\Phi_0}\right) + \frac{C}{2}\dot{\Phi}^2 \tag{223}$$

Using the Euler-Lagrange equations we have the same equations of motion. We have the same conjugate variable $Q \equiv \frac{\partial\mathcal{L}}{\partial\dot{\Phi}} = C\dot{\Phi}$, thus the Hamiltonian is:

$$\mathcal{H}_T = Q\dot{\Phi} - \mathcal{L} = \frac{Q^2}{2C} - \frac{I_0\Phi_0}{2\pi}\cos\left(\frac{2\pi\Phi}{\Phi_0}\right) \tag{224}$$

Now let's quantize it as we did for the LC Circuit:

$$\mathcal{H}_T = 4\ E_C\ \hat{n}^2 - E_J\cos\hat{\phi} \quad , \quad E_J = \frac{I_0\Phi_0}{2\pi} \tag{225}$$

Where E_J is the josephson junction energy. We can see that the functional form of the phase is now non-linear. Now let's quantize using creation and annihilation operators:

$$\hat{n} = i\; n_{ZPF}(\hat{c}+\hat{c}^{\dagger}) \quad n_{ZPF} = \left(\frac{E_J}{32E_C}\right) \tag{226}$$

$$\hat{\phi} = \phi_{ZPF}(\hat{c}-\hat{c}^{\dagger}) \quad \phi_{ZPF} = \left(\frac{2E_C}{E_J}\right) \tag{227}$$

Noting that $\phi << 1$ in the transmon regime $\frac{E_J}{E_C} >> 1$, we take the taylor expansion:

$$\mathcal{H}_T = 4E_C n_{ZPF}^2 (c+c^{\dagger})^2 - E_J\left[1-\frac{1}{2}E_J\phi_{ZPF}^2(c-c^{\dagger})^2+\frac{1}{24}E_J\phi_{ZPF}^4(c-c^{\dagger})^4+\dots\right] \tag{228}$$

$$\boxed{\mathcal{H}_T \approx \sqrt{8E_CE_J}\left[c^{\dagger}c+\frac{1}{2}\right] - E_J - \frac{E_C}{12}(c^{\dagger}+c)^4} \tag{229}$$

We can expand it on $c, c^{\dagger}$ and drop fast-rotating terms (with uneven number of c and $c^{\dagger}$) and dropping constants $\omega_0 = \sqrt{8E_CE_J}$ and $\delta = -E_J$:

$$\mathcal{H}_T \approx \omega_0 c^{\dagger}c + \frac{\delta}{2}\left(((c^{\dagger}c)^2 + c^{\dagger}c\right) \tag{230}$$

$$\boxed{\mathcal{H}_T = \left(\omega_0 + \frac{\delta}{2}\right)c^{\dagger}c + \frac{\delta}{2}(c^{\dagger}c)^2} \tag{231}$$

Which is a Hamiltonian of a *Duffing Oscillator*. From the definition of $c = \sum_j \sqrt{j+1}\,|j\rangle\langle j+1|$ we have $c^{\dagger}c = \sum_j j\,|j\rangle\langle j|$, thus:

$$\mathcal{H}_T = \omega_0 c^{\dagger}c + \frac{\delta}{2}c^{\dagger}c(c^{\dagger}c-1) = \sum_j \left[\left(\omega_0 - \frac{\delta}{2}\right)j + \frac{\delta}{2}j^2\right]|j\rangle\langle j| = \sum_j \omega_j\,|j\rangle\langle j| \tag{232}$$

where $\omega_j = \left(\omega_0 - \frac{\delta}{2}\right)j + \frac{\delta}{2}j^2$ are the energy levels of the transmon.

3.1.4 Qubit Drive and Rotating Wave Approximation

Now that we know about the transmon qubit we need a way to control it. This is done by applying an electric field $\vec{E}(t) = \vec{E}_0 e^{-i\omega_d t} + \vec{E}_0^{*} e^{i\omega_d t}$ which induces a dipole interaction between the transmon and microwave field. Now we have the qubit Hamiltonian and a drive Hamiltonian:

$$H = H_0 + H_d \quad , \quad H_0 = -\frac{1}{2}\hbar\omega_q\sigma^Z \tag{233}$$

Since we assume that the qubit hamiltonian is only a 2 level system we can use Pauli raising/lowering operators $\sigma^{\pm} = \frac{1}{2}\left(\sigma^X \mp \sigma^Y\right)$ which acts as annihilation/creation operators:

$$\sigma^+ \left|0\right\rangle = \left|1\right\rangle \qquad \sigma^- \left|1\right\rangle = \left|0\right\rangle \tag{234}$$

Since the electric field will excite and de-excite the qubit, we define the dipole operator as $\vec{d} = \vec{d}_0\sigma^+\vec{d}_0^*\sigma^-$. The drive hamiltonian is:

$$\begin{aligned} H_d = -\vec{d}\cdot\vec{E}(t) &= -(\vec{d}_0\sigma^+ + \vec{d}_0^*\sigma^-)\cdot(\vec{E}_0 e^{-i\omega_d t} + \vec{E}_0^* e^{i\omega_d t}) \\ &= -(\vec{d}_0\cdot\vec{E}_0 e^{-i\omega_d t} + \vec{d}_0\cdot\vec{E}_0^* e^{i\omega_d t})\sigma^+ - (\vec{d}_0^*\cdot\vec{E}_0 e^{-i\omega_d t} + \vec{d}_0^*\cdot\vec{E}_0^* e^{i\omega_d t})\sigma^- \end{aligned}$$

Defining $\Omega = \vec{d}_0\cdot\vec{E}_0$ and $\tilde{\Omega} = \vec{d}_0\cdot\vec{E}_0^*$, we have the Drive Hamiltonian:

$$\boxed{H_d = -\hbar\Big[\Omega e^{-i\omega_d t} + \tilde{\Omega}e^{i\omega_d t}\Big]\sigma^+ - \hbar\Big[\tilde{\Omega}^* e^{-i\omega_d t} + \Omega^* e^{i\omega_d t}\Big]\sigma^+} \tag{235}$$

We have to move to the interaction picture using the transformation $H_{d,I} = UH_dU^\dagger$, where:

$$\boxed{U = e^{i\frac{H_0 t}{\hbar}} = \mathbb{1}\cos\left(\frac{\omega_q t}{2}\right) - i\ \sigma^Z\sin\left(\frac{\omega_q t}{2}\right)}$$

Let's compute the operator terms separately:

$$\begin{aligned} &\left(\mathbb{1}\cos\left(\frac{\omega_q t}{2}\right) - i\ \sigma^Z\sin\left(\frac{\omega_q t}{2}\right)\right)\sigma^+\left(\mathbb{1}\cos\left(\frac{\omega_q t}{2}\right) + i\ \sigma^Z\sin\left(\frac{\omega_q t}{2}\right)\right) \\ &= \sigma^+\cos^2\left(\frac{\omega_q t}{2}\right) + i\sigma^+\sigma^Z\cos\left(\frac{\omega_q t}{2}\right)\sin\left(\frac{\omega_q t}{2}\right) - i\sigma^Z\sigma^+\cos\left(\frac{\omega_q t}{2}\right)\sin\left(\frac{\omega_q t}{2}\right) + \sigma^Z\sigma^+\sigma^Z\sin^2\left(\frac{\omega_q t}{2}\right) \\ &= \left(\cos^2\left(\frac{\omega_q t}{2}\right) - \sin^2\left(\frac{\omega_q t}{2}\right)\right) + i\left(\cos\left(\frac{\omega_q t}{2}\right)\sin\left(\frac{\omega_q t}{2}\right) + \cos\left(\frac{\omega_q t}{2}\right)\sin\left(\frac{\omega_q t}{2}\right)\right)\sigma^+ \\ &= \big[\cos\omega_q t + i\sin\omega_q t\big]\sigma^+ = e^{i\omega_q t}\sigma^+ \end{aligned} \tag{236}$$

We can do the same for σ^- : $U\sigma^-U^\dagger = e^{-i\omega_q t}\sigma^-$. Thus:

$$H_{d,I} = -\hbar\Big[\Omega e^{-i\omega_d t} + \tilde{\Omega}e^{i\omega_d t}\Big]e^{i\omega_q t}\sigma^+ - \hbar\Big[\tilde{\Omega}^* e^{-i\omega_d t} + \Omega^* e^{i\omega_d t}\Big]e^{-i\omega_q t}\sigma^+$$

$$\boxed{H_{d,I} = -\hbar\Big[\Omega e^{-i\Delta_q t} + \tilde{\Omega}e^{i(\omega_d+\omega_q)t}\Big]e^{i\omega_q t}\sigma^+ - \hbar\Big[\tilde{\Omega}^* e^{-i(\omega_d+\omega_q)t} + \Omega^* e^{i\omega_d t}\Big]e^{-i\Delta_q t}\sigma^+} \tag{237}$$

Where $\Delta_q = \omega_q - \omega_d$. Now we can use the **rotating wave approximation** : $\omega_q + \omega_d >> \Delta_q$, the terms with $\omega_q + \omega_d$ will oscillate fast, therefore we can drop them because they will average out. Now Our Hamiltonian is:

$$H_{d,I}^{\text{RWA}} = -\hbar\Omega e^{-i\Delta_q t}\sigma^+ - \hbar\Omega^* e^{i\Delta_q t}\sigma^- \tag{238}$$

Moving back to the Schrödinger picture:

$$\boxed{H_d^{\text{RWA}} = -\hbar\Omega e^{-i\omega_d t}\sigma^+ - \hbar\Omega^* e^{i\omega_d t}\sigma^-} \tag{239}$$

Adding the drive hamiltonian and the qubit hamiltonian we have:

$$H^{\text{RWA}} = -\frac{1}{2}\hbar\omega_q\sigma^Z - \hbar\Omega e^{-i\omega_d t}\sigma^+ - \hbar\Omega^* e^{i\omega_d t}\sigma^- \tag{240}$$

We can go to the frame of the drive by doing the following transformation $U_d = e^{-i\omega_d t\frac{\sigma}{2}}$, then in the frame of the drive using the Schrödinger Equation, the effective Hamiltonian is:

$$H_{\text{eff}} = U_d H^{\text{RWA}} U_d^\dagger - i\hbar U_d \dot{U_d}^\dagger$$

After a bit of algebra...

$$H_{\text{eff}} = -\frac{1}{2}\hbar\omega_q\sigma^Z - \hbar\Omega\sigma^+ - \hbar\Omega^*\sigma^- + \frac{1}{2}\hbar\Omega_d\sigma^Z = -\frac{1}{2}\hbar\Delta_q\sigma^Z - \hbar\Omega\sigma^+ - \hbar\Omega^*\sigma^-$$

Assuming that the drive is real, $\Omega = \Omega^*$, we have the desired hamiltonian:

$$\boxed{H_{\text{eff}} = -\frac{1}{2}\hbar\Delta_q\sigma^Z - \hbar\Omega\sigma^X} \tag{241}$$

This shows that when the drive is resonant with the qubit ($\Delta_q = 0$) the drive causes an X rotation in the Bloch sphere. On the other hand, an off-resonant qubit drive has additional Z rotations generated by the σ^Z contribution, and those manifest themselves as oscillations in a Ramsey experiment.

References

[1] Scott Aaronson. *Quantum Computing Since Democritus.* Cambridge University Press, New York, NY, USA, 2013.

[2] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal. Qiskit: An Open-source Framework for Quantum Computing, January 2019.

[3] Abraham Asfaw, Luciano Bello, Yael Ben-Haim, Sergey Bravyi, Lauren Capelluto, Almudena Carrera Vazquez, Jack Ceroni, Jay Gambetta, Shelly Garion, Leron Gil, Salvador De La Puente Gonzalez, David McKay, Zlatko Minev, Paul Nation, Anna Phan, Arthur Rattew, Javad Shabani, John Smolin, Kristan Temme, Madeleine Tod, and James Wootton. Learn quantum computation using qiskit, 2019.

[4] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin Chan. Quantum algorithms for quantum chemistry and quantum materials science. *arXiv preprint arXiv:2001.03685*, 2020.

[5] Giuliano Benenti, Giulio Casati, and Giuliano Strini. *Principles of Quantum Computation and Information.* WORLD SCIENTIFIC, 2004.

[6] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, December 2014.

[7] Sergey Bravyi, Jay M Gambetta, Antonio Mezzacapo, and Kristan Temme. Tapering off qubits to simulate fermionic hamiltonians. *arXiv preprint arXiv:1701.08213*, 2017.

[8] Sergey B Bravyi and Alexei Yu Kitaev. Fermionic quantum computation. *Annals of Physics*, 298(1):210–226, 2002.

[9] Ronald De Wolf. Quantum computing: Lecture notes. *arXiv preprint arXiv:1907.09415*, 2019.

[10] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, 2019.

[11] Jarrod McClean, Nicholas Rubin, Kevin Sung, Ian David Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai, Eric Schuyler Fried, Craig Gidney, Brendan Gimby, et al. Openfermion: the electronic structure package for quantum computers. *Quantum Science and Technology*, 2020.

[12] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3):030503, 2018.

[13] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3):030503, 2018.

[14] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[15] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[16] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc. Press, 1994.

[17] Vlatko Vedral. The role of relative entropy in quantum information theory. *Reviews of Modern Physics*, 74(1):197, 2002.

[18] Guillaume Verdon, Jacob Marks, Sasha Nanda, Stefan Leichenauer, and Jack Hidary. Quantum hamiltonian-based models and the variational quantum thermalizer algorithm. 2019.

[19] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *arXiv preprint arXiv:2007.14384*, 2020.