# Notes on Quantum Search Algorithm

Zhongwei Jiang, Linfeng Zhao

October 2, 2024

Project Link: https://github.com/little-little-red/NoteForQuantumSearchAlgorithm

# Contents

# Chapter 1

# Quantum Computation

*This chapter is based on the book[4].*

## 1.1 Quantum Gate

### 1.1.1 Single-qubit gate

The number of qubits in a quantum state depends on the number of classical bits in its dimension, so we usually call a vector $|\psi\rangle = a|0\rangle + b|1\rangle$ parameterized by two complex numbers $a$ and $b$ satisfying $|a|^2 + |b|^2 = 1$ a qubit. According to the constraints and phase redundancy, the Bloch sphere representation gives us a more intuitive way to observe the qubit, an isomorphism from a qubit to the three-dimensional real sphere:

$$\mathbb{CP}^1 \to S^2$$
$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle \mapsto \vec{n}_\psi = (\cos\varphi\sin\theta, \sin\varphi\sin\theta, \cos\theta)^T. \tag{1.1.1}$$

where $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$.

Operations on a qubit must preserve the norm and thus are described by $2 \times 2$ unitary matrices. By $\det|U| = e^{2i\alpha}$ Then

$$e^{-i\alpha}U \in SU(2) \tag{1.1.2}$$

They're only off by one global phase, so we usually ignore the global phase and only consider $SU(2)$. The Lie group $SU(2)$ has three generators $i\sigma_j$, $\sigma_j$ called Pauli matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.1.3}$$

From Lie group elements, any single-qubit unitary can be written as a product of exponentials of Pauli matrices by a global phase:

$$U = e^{i\alpha}\exp\left(-i\frac{\omega}{2}\vec{n}\cdot\vec{\sigma}\right), \tag{1.1.4}$$

which $\vec{n}$ is the coordinates of the rotation axis on the Bloch sphere. From the isomorphism 1.1.1, we can also induce an isomorphism of operations on two spaces:

$$SU(2)/\{\pm 1\} \to SO(3)$$
$$\exp\left(-i\frac{\omega}{2}\vec{n}\cdot\vec{\sigma}\right) \mapsto \exp\left(\omega\vec{n}\cdot\vec{J}\right) \tag{1.1.5}$$

which $\omega \in [0, \pi]$ and $J_j$ are the three generators of Lie group $SO(3)$, the former refers to single-qubit gates and the latter to transformations on the Bloch sphere. For visualization and convenience, we let $R_{\vec{n}}(\omega) \coloneqq \exp\left(-i\frac{\omega}{2}\vec{n}\cdot\vec{\sigma}\right)$.

In the following, we use $XYZ$ instead of $\sigma_x\sigma_y\sigma_z$, here are some other single-qubit gates which are frequently used, they are called Hadamard gate, phase gate, and $\pi/8$ gate:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{1.1.6}$$

### 1.1.2   Properties of single-qubit gate

Here we list some properties of single-qubit gates:

**Theorem 1.1.1.** *A single qubit gate $U$ can be decomposed into the following form*

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta). \tag{1.1.7}$$

**Corollary 1.1.2.** *For suitable parameters $\alpha$, $\beta_k$, $\gamma_k$ and nonparallel vectors $\vec{n}$ and $\vec{m}$, there is*

$$U = e^{i\alpha} \prod_{j=1}^{k} R_{\vec{n}}(\beta_j) R_{\vec{m}}(\gamma_j). \tag{1.1.8}$$

**Corollary 1.1.3.** *If $U$ is a single qubit gate, then there exists $A, B, C \in SU(2)$, $ABC = I$, and then there is*

$$U = e^{i\alpha} AXBXC \tag{1.1.9}$$

*Proof.* let

$$A = R_z(\beta) R_y(\gamma/2), \ B = R_y(-\gamma/2) R_z(-(\gamma+\beta)/2), \ C = R_z((\gamma-\beta)/2)$$

then we have

$$AXBXC = R_z(\beta) R_y(\gamma) R_z(\delta) \tag{1.1.10}$$

and $ABC = I$. $\qquad\qquad\square$

**Theorem 1.1.4.** *We have the product of Pauli matrices,*

$$\sigma_j \sigma_k = i\epsilon_{jk}^{l} \sigma_l + \delta_{jk} I. \tag{1.1.11}$$

**Corollary 1.1.5.** *The element of Lie group $SU(2)$ can be changed into the following form,*

$$R_{\vec{n}}(\omega) := \exp\left(-i\frac{\omega}{2}\vec{n}\cdot\vec{\sigma}\right) = \cos\left(\frac{\omega}{2}\right) I - i\sin\left(\frac{\omega}{2}\right)(\vec{n}\cdot\vec{\sigma}). \tag{1.1.12}$$

*Proof.* Considering the Taylor expansion

$$\begin{aligned}
\exp\left(-i\frac{\omega}{2}\vec{n}\cdot\vec{\sigma}\right) &= \sum_{k=0}^{\infty} \frac{1}{k!}\left(-i\frac{\omega}{2}\vec{n}\cdot\vec{\sigma}\right)^{k} \\
&= \sum_{k=0}^{\infty} \frac{1}{(2k)!}\left(-i\frac{\omega}{2}\right)^{2k}(\vec{n}\cdot\vec{\sigma})^{2k} + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!}\left(-i\frac{\omega}{2}\right)^{2k+1}(\vec{n}\cdot\vec{\sigma})^{2k+1} \\
&= \cos\left(\frac{\omega}{2}\right) I - i\sin\left(\frac{\omega}{2}\right)(\vec{n}\cdot\vec{\sigma}),
\end{aligned} \tag{1.1.13}$$

where we have $(\vec{n}\cdot\vec{\sigma})^2 = I$ from the theorem 1.1.4. $\qquad\qquad\square$

**Corollary 1.1.6.** *If a rotation through an angle $2\omega_1$ about an axis $\vec{n_1}$ is followed by a rotation through an angle $2\omega_2$ about an axis $\vec{n_2}$, then the overall rotation is through an angle $\omega_{12}$ about an axis $\vec{n_{12}}$ given by*

$$\cos\omega_{12} = \cos\omega_1 \cos\omega_2 - \sin\omega_1 \sin\omega_2 \vec{n_1}\cdot\vec{n_2}$$
$$\sin\omega_{12}\vec{n_{12}} = \sin\omega_1 \cos\omega_2 \vec{n_2} + \cos\omega_1 \sin\omega_2 \vec{n_2} + \sin\omega_1 \sin\omega_2 \vec{n_1}\times\vec{n_2}.$$

*Proof.* We know $R_{\vec{n_1}}(2\omega_1) = \cos\omega I - i\sin\omega(\vec{n_1}\cdot\vec{\sigma})$ and the same as index 2, so the product of two rotations is

$$\begin{aligned}
&R_{\vec{n_1}}(2\omega_1) R_{\vec{n_2}}(2\omega_2) \\
&= \cos\omega_1 \cos\omega_2 I - i\sin\omega_1 \cos\omega_2(\vec{n_1}\cdot\vec{\sigma}) - i\cos\omega_1 \sin\omega_2(\vec{n_2}\cdot\vec{\sigma}) - \sin\omega_1 \sin\omega_2(\vec{n_1}\cdot\vec{\sigma})(\vec{n_2}\cdot\vec{\sigma}) \\
&= \cos\omega_{12} I - i\sin\omega_{12}(\vec{n_{12}}\cdot\vec{\sigma}),
\end{aligned}$$

where we have $(\vec{n_1}\cdot\vec{\sigma})(\vec{n_2}\cdot\vec{\sigma}) = i(\vec{n_1}\times\vec{n_2})\cdot\vec{\sigma} + (\vec{n_1}\cdot\vec{n_2})\cdot\vec{\sigma}$ from Theorem 1.1.4. $\qquad\qquad\square$
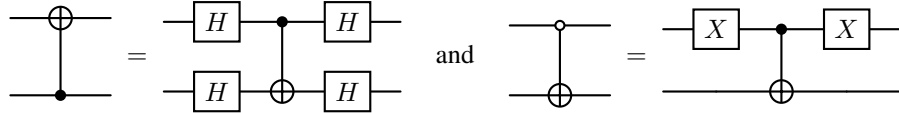
### 1.1.3 Controlled operation

Analogous to the if statement of a classical circuit, we can use the controlled-NOT gate to control the target qubit with the control qubit, its matrix form and circuit are shown as follows:

$$CNOT = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} \tag{1.1.14}$$



$$\tag{1.1.15}$$

CNOT gate makes the state of the target qubit be flipped when the control qubit is 1 and remains unchanged when the control qubit is 0. That is $|c\rangle\,|t\rangle \to |c\rangle\,X^c\,|t\rangle$. We will also use two deformations of CNOT gates:
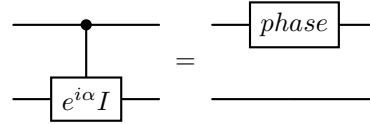


$$\tag{1.1.16}$$

We expect to extend this control from $X$ to an arbitrary single-qubit gate $U$. That is, $|c\rangle\,|t\rangle \to |c\rangle\,U^c\,|t\rangle$ which is known as a controlled-U gate. Show that:



$$\tag{1.1.17}$$
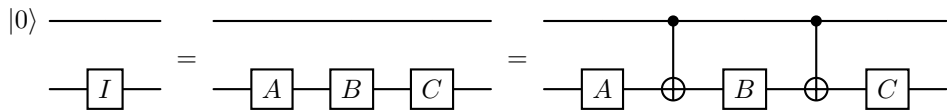
We wish to decompose this into combining single quantum bit gates and CNOT gates. With the theorem1.1.3, we can divide the controlled-U gate into several parts, looking first at the global phase part, which has:



$$\tag{1.1.18}$$

where $phase = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$.

Looking at the $AXBXC$ part again, we have the expectation that if the control qubit is $|0\rangle$, then we do nothing, and adding a CNOT gate anywhere doesn't change that, so we have.
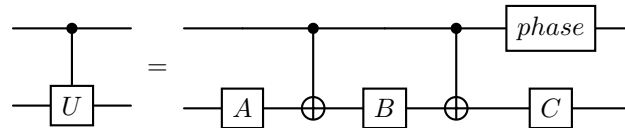


$$\tag{1.1.19}$$

This construction also satisfies: if the control qubit is $|1\rangle$, we apply $AXBXC$ to the target qubit.

So we have that:

**Theorem 1.1.7.** *The controlled-U gate can be decomposed with CNOT gates and single-qubit gates into this form:*



$$\tag{1.1.20}$$

Considering the case of multiple control qubits, first look at the case of two control qubits, you can construct: if $V$ is a single qubit gate satisfying $V^2 = U$, we have:



$$\tag{1.1.21}$$

Then we can extend this to the case of $n$ control qubits, there are many kinds of construction methods, and we use the recursive method to construct:



$$(1.1.22)$$

Ultimately we will show that any unitary operation can be composed to an arbitrarily good approximation from just H, S, T, CNOT gates. Here is the construction of the Toffoli Gate:



$$(1.1.23)$$

## 1.2 Universal Gate Set

### 1.2.1 Single-qubit and CNOT gates are universal

In the section 1.1.3, we showed that $C^n(U)$ gate can be decomposed into a combination of single-qubit gates and CNOT gates. In this section 1.2.1, we will show that any unitary gate can be combined from $C^n(U)$ gate.
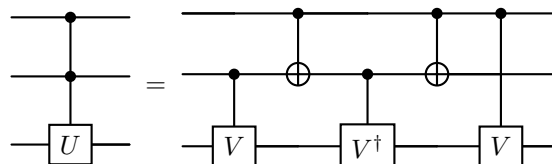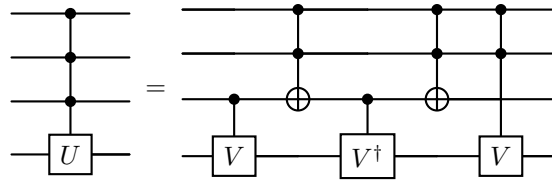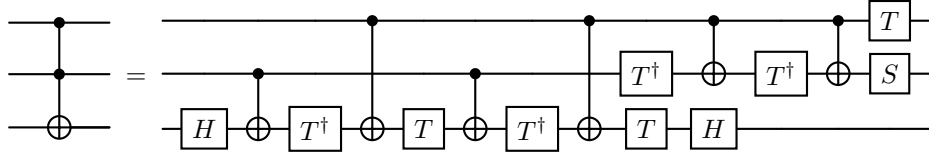
First, we introduce the lemma:

**Lemma 1.2.1.** *Any D-dimensional unitary transformation U can always be decomposed into the product of $d(d-1)/2$ two-level unitary transformations under a natural basis.*
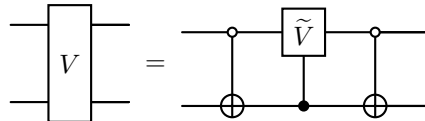
So exists two-level unitary transformations $V_j$ made $U = \prod\limits_{j=1}^{d(d-1)/2} V_j$[1]. Two-level unitary transformations are unitary matrices that act non-trivially only on two or fewer vector components. The proof idea of this theorem is simple [2], let's focus on the two-level unitary transformations $V$.

Consider the binary expansion of the two bases $|s\rangle$ and $|t\rangle$ on which $V$ operates, where $s = s_1 \ldots s_n$ and $t = t_1 \ldots t_n$. We can use Toffoli gate to turn $s_j$ into $t_j$, after $n-1$ steps $|s_1 \ldots s_{k-1} s_k s_{k+1} \ldots s_n\rangle$ into $|t_1 \ldots t_{k-1} s_k t_{k+1} \ldots t_n\rangle$. It puts $|s\rangle$ and $|t\rangle$ in the same qubit and makes $V$ into a $C^n(\widetilde{V})$ gate which $\widetilde{V}$ is the non-trivial $2 \times 2$ unitary sub-matrix of $V$.

Let's take an example to illustrate this lemma, consider the $U$ gate:

$$V = \begin{pmatrix} a & 0 & 0 & c \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ b & 0 & 0 & d \end{pmatrix} \quad \text{and} \quad \widetilde{V} = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \tag{1.2.1}$$

Notice that $V$ acts non-trivially only on the states $|00\rangle$ and $|11\rangle$, so we can use the CNOT gate to turn $|00\rangle$ into $|01\rangle$, then we can get the $C(\widetilde{V})$ gate. Here is the circuit of $V$:



$$(1.2.2)$$

So we have:

**Theorem 1.2.1.** *Any unitary gate can be combined from single-qubit gates and CNOT gates.*

---

[1]Unless otherwise specified, the conjunction sign in this document is calculated according to the composite order of the map, that is $\prod\limits_{j=1}^{n} a_j = a_n a_{n-1} \cdots a_1$.

[2]Each single-qubit gates has $d(d-1)/2$ degrees of freedom and each two-level unitary transformation turn one of single-qubit gate elements into zero in order. So exists two-level unitary transformations $V_j$ made $\left( \prod\limits_{j=1}^{d(d-1)/2} V_j \right) U = I$.

### 1.2.2 Approximate universal operators with a discrete set

It is hard to implement all single-qubit gates, so we consider approximating it with a discrete set that we can implement. We introduce the induced norm of operators to measure the degree of approximation:

$$E(U,V) = \|U - V\| := \sup_{\substack{|\psi\rangle \in \mathbb{C}^n \\ \||\psi\rangle\|=1}} \|(U-V)|\psi\rangle\| \tag{1.2.3}$$

$U$ is the target unitary operation we wish to implement, $V$ is the unitary operation implemented in practice, and $E(U,V)$ is the error when $V$ is implemented instead of $U$. It is natural that:

**Theorem 1.2.2.** *In the approximation of $m$ gates, the error is added at most linearly:*

$$E\left(\prod_{j=1}^{m} U_j, \prod_{j=1}^{m} V_j\right) \leq \sum_{j=1}^{m} E(U_j, V_j) \tag{1.2.4}$$

*Proof.* Considering $|\psi_0\rangle$ which maximizes $\left\|\left(\prod\limits_{i=1}^{m} U_i - \prod\limits_{i=1}^{m} V_i\right)|\psi\rangle\right\|$ and definition $|\psi_i\rangle = V_i|\psi_{i-1}\rangle$, $|\Delta_i\rangle = U_i|\psi_{i-1}\rangle - |\psi_i\rangle$, we can see that

$$E\left(\prod_{j=1}^{m} U_j, \prod_{j=1}^{m} V_j\right) = \left\||\Delta_m\rangle + \sum_{j=1}^{m-1}\left(\prod_{k=j+1}^{m} U_k\right)|\Delta_j\rangle\right\| \leq \sum_{j=1}^{m} \||\Delta_j\rangle\| \leq \sum_{j=1}^{m} E(U_j, V_j) \qquad \square$$

We are more concerned with the error of the approximation in the measurement. Considering $M$ is a POVM element in an arbitrary measurement POVM, and $P_U$(or $P_V$) is the probability of obtaining this outcome if $U$(or $V$) were performed with the state $|\psi\rangle$, We can prove[1] that:

$$|P_U - P_V| \leq 2E(U,V) \tag{1.2.5}$$

This makes it possible that if we want the probability difference between the approximate line and the ideal line on a certain outcome to be within a tolerance $\Delta > 0$, we only need to ensure that $E(U_j, V_j) \leq \Delta/(2m)$.

From the decomposition given by corollary 1.1.2, we can further care about the degree of approximation of $R_{\vec{n}}(\omega)$. Let's start by introducing a neat lemma :

**Lemma 1.2.2.** *Considering $\alpha$, $\omega \in \mathbb{R}/2\pi\mathbb{Z}$, if $\omega/\pi \in \mathbb{R}\backslash\mathbb{Q}$, we can find a sub-sequences $\{x_n\}$ of sequences $\{n\}$ makes $\lim\limits_{n\to\infty} x_n\omega = \alpha$.*

*Proof.* $\forall \epsilon > 0, \exists N = \frac{2\pi}{\epsilon}$, when $n > N$, we can find $|j\omega - k\omega| \leq \frac{2\pi}{n}$ which $j, k \in \{n\}, j > k$, let $x_n = (j-k)\left\lfloor\frac{\alpha}{|j\omega-k\omega|}\right\rfloor$, so we have $|x_n\omega - \alpha| < |j\omega - k\omega| \leq \frac{2\pi}{n} < \epsilon$. $\square$

Further, we have[2]:

$$\begin{aligned} &E(R_{\vec{n}}(\alpha), R_{\vec{n}}(\omega)^{x_n}) \\ &= E(R_{\vec{n}}(\alpha), R_{\vec{n}}(\alpha + (x_n\omega - \alpha))) \\ &= |1 - \exp((x_n\omega - \alpha)/2)| \leq \epsilon/2 \end{aligned} \tag{1.2.6}$$

According to the proof, we must find the gate with the angle the theorem requires. Fortunately, the $THTH$ gate has the angel we need. It is a rotation of the Bloch sphere about an axis along $\vec{n} = (\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8})$ and through an angle $\omega$ defined by $\cos\frac{\omega}{2} = \cos^2\frac{\pi}{8}$, so we just let $R_{\vec{n}}(\omega) = THTH$.

From corollary 1.1.2, to construct unitary operation $U$, we still need to approximate the rotation of the other axis, but even more fortunate is that $HR_{\vec{n}}(\omega)H$ is exactly what we need, has axis $\vec{m} = (\cos\frac{\pi}{8}, -\sin\frac{\pi}{8}, \cos\frac{\pi}{8})$ and the angle $\omega$, like $R_{\vec{n}}(\omega)$, is an irrational multiple of $\pi$. After layers and layers of preparation, let's list the final approximation with suitable positive integers $j_n$ to U:

$$E(U, \prod_{j=1}^{2k} R_{\vec{n}}(\omega)^{j_n} H) \leq k\epsilon \tag{1.2.7}$$

**Theorem 1.2.3.** *Given any single qubit unitary operation $U$ and any $\epsilon > 0$, it is possible to approximate $U$ to within $\epsilon$ [3]using a circuit composed of $H$ gates and $T$ gates alone.*

---

[1] Let $|\Delta\rangle = (U - V)|\psi\rangle$, notice that $\langle\psi|U^\dagger M U|\psi\rangle - \langle\psi|V^\dagger M V|\psi\rangle = \langle\psi|U^\dagger M|\Delta\rangle + \langle\Delta|M V|\psi\rangle$.

[2] The construction of $x_n$ in this part is only to prove the existence, without considering the complexity. There are far less complex constructs.

[3] The difference between Equation 1.2.7 and here by one coefficient can be solved by the setting of $\epsilon$ in the proof of Theorem 1.2.2, but it does not matter.

### 1.2.3   Circuit size for approximation

It makes no sense to talk about its existence without giving its size, so let's estimate the number of gates needed for the approximation in the previous section. For convenience of stating the theorem, we say that $S$ is an $\epsilon$-net in $W$, if every point in $W$ is within a distance $\epsilon$ of some point in $S$, where $S, W \in SU(2)$ and $\epsilon > 0$ and the distance is $D(U,V) := \mathrm{tr}|U - V|$. And we define $\mathcal{G}_l$ as the set of all words of length at most $l$.

**Theorem 1.2.4** (Solovay-Kitaev theorem)**.** *Let $\mathcal{G}$ be a finite set of elements in $SU(2)$ containing its own inverses, such that $\langle \mathcal{G} \rangle$ is dense in $SU(2)$. Let $\epsilon > 0$ be given. Then $\mathcal{G}_l$ is an $\epsilon$-net in $SU(2)$ for $l = O(\ln^c(1/\epsilon))$, where $c = \ln 5/\ln(3/2)$.*

According to the theorem, we can know that to approximate a circuit containing $m$ single-qubit unitary operations to an accuracy $\epsilon$ requires $O(m \ln^c(m/\epsilon))$ gates from the discrete set. The proof of this theorem is quite long, so only the main ideas are given here. Let's first introduce the lemma:

**Lemma 1.2.3.** *Let $\mathcal{G}$ be a finite set of elements in $SU(2)$ containing its own inverses, such that $\langle \mathcal{G} \rangle$ is dense in $SU(2)$. For some $\epsilon > 0$ and any $k \in \mathbb{N}$, if $\mathcal{G}_l$ is an $\epsilon^2$-net for $S_\epsilon$, then $\mathcal{G}_{5^k l}$ is an $\epsilon(k)^2$-net for $S_{\epsilon(k)}$. where $\epsilon(k) = (C\epsilon)^{(3/2)^k}/C$, $\epsilon(k)^2 < \epsilon(k+1)$ and $S_\epsilon := \{U \in SU(2) | D(U,I) \le \epsilon\}$.*

*Proof.* We know the element of $SU(2)$ can be written as $U = \exp(-i\vec{a} \cdot \vec{\sigma}/2)$, we write $U = u(\vec{a})$. And introduce the symbol $[U,V]_{gp} = UVU^\dagger V^\dagger$. With constant $d$, we have two conclusions:

$$D([u(\vec{a}), u(\vec{b})]_{gp}, u(\vec{a} \times \vec{b})) \le d\epsilon^3 \quad \text{and} \quad D(u(\vec{a}), u(\vec{b})) = \left\| \vec{a} - \vec{b} \right\| + O(\epsilon^3) \tag{1.2.8}$$

If $U = u(\vec{x}) \in S_{\epsilon^2}$, we can find $\vec{y} \times \vec{z} = \vec{x}$ which $u(\vec{y}), u(\vec{z}) \in S_\epsilon$, so that we can find $u(\vec{y_0}), u(\vec{z_0}) \in \mathcal{G}_l \cap S_\epsilon$ make $D(u(\vec{y_0}), u(\vec{y})), D(u(\vec{z_0}), u(\vec{z})) \le \epsilon$. Notice that

$$\begin{aligned} D(U, [u(\vec{y_0}), u(\vec{z_0})]_{gp}) &\le D(U, u(\vec{y_0} \times \vec{z_0})) + D(u(\vec{y_0} \times \vec{z_0}), [u(\vec{y_0}), u(\vec{z_0})]_{gp}) \\ &= \|\vec{y} \times \vec{z} - \vec{y_0} \times \vec{z_0}\| + d\epsilon^3 \\ &\le (d+2)\epsilon^3 + O(\epsilon^4) \\ &\le C\epsilon^3 = \epsilon(1)^2 \end{aligned} \tag{1.2.9}$$

Specifically, given $U \in S_{\epsilon(1)}$, we can find $V \in \mathcal{G}_l$ such that $D(U,V) \le \epsilon(0)^2$, and thus $UV^\dagger \in S_{\epsilon(0)^2}$, so

$$D([u(\vec{y_0}), u(\vec{z_0})]_{gp}, UV^\dagger) = D([u(\vec{y_0}), u(\vec{z_0})]_{gp} V, U) \le \epsilon(1)^2 \tag{1.2.10}$$

that is, $\mathcal{G}_{5l}$ is an $\epsilon(1)^2$-net for $S_{\epsilon(1)}$. Recursively, $\mathcal{G}_{5^k l}$ is an $\epsilon(k)^2$-net for $S_{\epsilon(k)}$. □

For $U \in SU(2)$ we can find $U_0$ make $D(U_0, U) < \epsilon(0)^2 < \epsilon(1)$, so we have a first order approximation $U_0$ of $U$. We can go on to approximate their difference $V = UU_0^\dagger$, With the lemma, we can find $U_1$ make $D(U_1, V) < \epsilon(1)^2 < \epsilon(2)$, so we have a second order approximation $U_1 U_0$ of $U$. This way, we can approximate the accuracy we want which $\epsilon(k+1) < \epsilon$.

Although approximating a set of single-qubit gates is polynomial, approximating arbitrary unitary gates is hard. Considering the normalization of $n$ qubit states $\|| \psi \rangle\| = 1$ gives that the state space is an unit $(2^{n+1} - 1)$-dimensional unit sphere. Given the error $\epsilon$, a state approximation gives an $(2^{n+1} - 2)$-dimensional sphere of radius $\epsilon$. So to cover the state space, we need about[1]

$$\frac{S_{2^{n+1}-1}(1)}{V_{2^{n+1}-2}(\epsilon)} = \frac{\sqrt{\pi}\Gamma(2^n - \frac{1}{2})(2^{n+1}-1)}{\Gamma(2^n)\epsilon^{2^{n+1}-1}} = \Omega(\epsilon^{-2^{n+1}+1}) \tag{1.2.11}$$

states, where[2] $S_d(r), V_d(r)$ is the surface area, volume of a $d$-dimensional sphere of radius $r$. But for a fixed initial state, $m$ gates can only compute $O(n^{km})$ different states at most, where $k$ is a constant. We must have:

$$O(n^{km}) \ge \Omega(\epsilon^{-2^{n+1}+1}) \tag{1.2.12}$$

which gives us

$$m = \Omega\left(\frac{2^n \ln(1/\epsilon)}{\ln(n)}\right). \tag{1.2.13}$$

This is exponential in $n$, so it is hard to approximate arbitrary unitary gates.

---

[1] $O$ is for upper bounds, $\Omega$ is for lower bounds and $\Theta$ is for tight bounds.
[2] $S_k(r) = 2\pi^{(k+1)/2} r^k / \Gamma((k+1)/2), V_k(r) = 2\pi^{(k+1)/2} r^{k+1}/(k+1)\Gamma((k+1)/2)$, where $\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} \mathrm{d}t$.

## 1.3   Quantum Circuit Model

Before discussing the algorithm, let's make some basic assumptions about quantum computers (quantum circuit model) clear:

- A quantum computer consists of a classical part and a quantum part: The classical part is unnecessary but can simplify many tasks.

- A quantum circuit operates on $n$ qubits, so the state space is $2^n$-dimensional complex Hilbert space.

- It is assumed that any computational basis state $|x_1, \cdots, x_n\rangle$ can be prepared in at most $n$ steps.

- Gates can be applied to any subset of qubits as desired, and a universal family of gates can be implemented.

- Measurements may be performed on the computational basis of one or more of the qubits in the computer.

# Chapter 2

# Quantum Search Algorithm

## 2.1 Grover's Algorithm

*This section is based on the book[4].*

### 2.1.1 Main idea

In the search algorithm, we have the $N = 2^n$ elements in the search space $S = \{0,1\}^n$ and the target element $T \subseteq S$ which has $M$ elements. The output of the algorithm should be the state containing all the target elements, so we let

$$|\beta\rangle := \frac{1}{\sqrt{M}} \sum_{x \in T} |x\rangle \quad \text{and} \quad |\alpha\rangle := \frac{1}{\sqrt{N-M}} \sum_{x \notin T} |x\rangle. \tag{2.1.1}$$

And the algorithm's input should be a trivial state, may as well let[1]

$$|\psi\rangle := H^{\otimes n} |0\rangle \tag{2.1.2}$$

which use $|0\rangle$ to refer to $|0\rangle^{\otimes n}$ for convenience. There is

$$\langle\alpha|\beta\rangle = 0 \quad \text{and} \quad |\psi\rangle = \sqrt{\frac{N-M}{N}} |\alpha\rangle + \sqrt{\frac{M}{N}} |\beta\rangle. \tag{2.1.3}$$

We need to find some operation that makes $|\psi\rangle$ into $|\beta\rangle$. Turning some non-trivial angles is hard, so we consider the operation reflecting across some axes. Define two operations here, $O$ is a reflection about $|\alpha\rangle$ and $D$ is a reflection about $|\psi\rangle$.
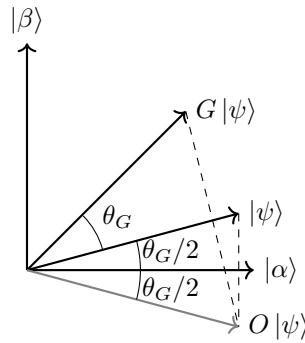


Figure 2.1

Let $\sin(\theta_G/2) = \sqrt{M/N}$, the angle between $|\psi\rangle$ and $|\alpha\rangle$ is $\theta_G/2$. As shown in Figure 2.1, operation $G = DO$ gives us a rotation of angle $\theta_G$, we call $G$ the Grover iteration. So we just need applying $R := \left\lfloor \frac{\pi - \theta_G}{2\theta_G} \right\rceil$[2] times Grover iteration on $|\psi\rangle$ to approximate $|\beta\rangle$, if $M \ll N$ we have

$$G^R |\psi\rangle = \cos(\frac{2R+1}{2}\theta_G) |\alpha\rangle + \sin(\frac{2R+1}{2}\theta_G) |\beta\rangle \approx |\beta\rangle. \tag{2.1.4}$$

The angular error is at most $\theta_G/2 \approx \sqrt{M/N}$.

---

[1] $|\psi\rangle$ is a commonly used state, called the equal superposition state.

[2] The actual function about $\lfloor x \rceil$ is to denote the integer closest to the real number $x$.

### 2.1.2  Oracle

Let's construct the reflection about $|\alpha\rangle$, we call it the oracle. We know that states in the space spanned by $|\alpha\rangle$ and $|\beta\rangle$ can be decomposed into

$$|x\rangle = |\alpha\rangle \langle\alpha|x\rangle + |\beta\rangle \langle\beta|x\rangle \tag{2.1.5}$$

and we wish

$$O|x\rangle = |\alpha\rangle \langle\alpha|x\rangle - |\beta\rangle \langle\beta|x\rangle. \tag{2.1.6}$$

We don't know the exact form of $|\alpha\rangle$ and $|\beta\rangle$, so we can not exactly construct the reflection in the whole space, but we just need it to be a reflection in the subspace spanned by $|\alpha\rangle$ and $|\beta\rangle$, may as well construct an operation that identifies each target element to flip it. To construct this, we need a special gate[1]



$$\tag{2.1.7}$$

which gives us a operation $|x\rangle \xrightarrow{U_f} (-1)^{f(x)} |x\rangle$. Then we just need to construct the mapping

$$S(x) = \begin{cases} 1 & x \in T \\ 0 & x \notin T \end{cases} \tag{2.1.8}$$

such that $U_S$ satisfies the properties we need:

$$\begin{aligned} U_S|\beta\rangle &= -|\beta\rangle \\ U_S|\alpha\rangle &= |\alpha\rangle. \end{aligned} \tag{2.1.9}$$

So the $U_S$ is the Oracle.

### 2.1.3  Grover iteration

Then, let's construct the reflection about $|\psi\rangle$. Same as the oracle, we decompose states into

$$|x\rangle = |\psi\rangle \langle\psi|x\rangle + |\psi_\perp\rangle \langle\psi_\perp|x\rangle \tag{2.1.10}$$

which $|\psi_\perp\rangle = \sqrt{\frac{M}{N}}|\alpha\rangle - \sqrt{\frac{N-M}{N}}|\beta\rangle$, and we wish

$$D|x\rangle = |\psi\rangle \langle\psi|x\rangle - |\psi_\perp\rangle \langle\psi_\perp|x\rangle. \tag{2.1.11}$$

But this time, we know $|\psi\rangle$ specifically, operation $2|\psi\rangle\langle\psi| - I$ is what we need. So we get the construction of the Grover iteration:

$$G = DO = (2|\psi\rangle\langle\psi| - I)U_S. \tag{2.1.12}$$

After constructing each sub-operation, the circuit of the algorithm is given here[2]:



$$\tag{2.1.13}$$
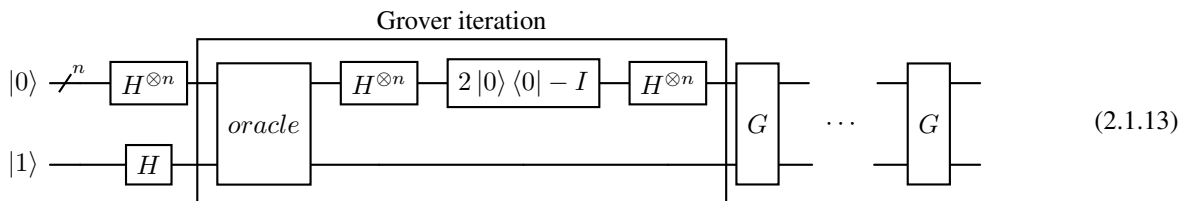
The ellipsis indicates that it is repeated $R$ times.

---

[1]Note that the mapping $|x\rangle \xrightarrow{U_f} (-1)^{f(x)} |x\rangle$ holds only if the controlled qubit is $|-\rangle$.
[2]The oracle workspace is omitted here.

### 2.1.4 Performance

Each of the operations in the Grover iteration may be efficiently implemented on a quantum computer. So we just need to focus on the number of Grover iterations.

We already know, in order to rotate $|\psi\rangle$ near $|\beta\rangle$, we need to repeat the Grover iteration

$$R = \left\lfloor \frac{2 \arccos \sqrt{M/N}}{\arcsin \sqrt{M/N}} \right\rceil \tag{2.1.14}$$

times. This is not intuitive enough, so let's sacrifice some precision to find a simpler expression. We know that $R \leq \lceil \pi/2\theta_G \rceil$, and if $M \leq N/2$ we have

$$\frac{\theta_G}{2} \geq \sin \frac{\theta_G}{2} = \sqrt{\frac{N}{M}}, \tag{2.1.15}$$

from which we obtain an elegant upper bound on the number of iterations required,

$$R \leq \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil. \tag{2.1.16}$$

That is, $R = O(\sqrt{N/M})$ Grover iterations must be performed to obtain a solution to the search problem with high probability, a quadratic improvement over the $O(N/M)$ oracle calls required classically.

If $M > N/2$, the quantum search algorithm is not necessary. We can just randomly pick an item from the search space, and then check that it is a solution using the oracle. This approach has a success probability of at least one-half and only requires one consultation with the oracle.

## 2.2 Bounds for Grover's algorithm

*This section is based on the paper[2], Here we restrict ourselves to the case $M = 1$.*

### 2.2.1 Main idea

Oracle is at the heart of Grover's algorithm, so let's focus on the oracle and ignore the other operations, we write the state that uses $R$ queries to the oracle as

$$|\Psi_x^R\rangle = \prod_{j=1}^{R} U_j O_x |\psi\rangle, \tag{2.2.1}$$

where $U_i$ is some unitary operation and $O_x$ is the oracle which marks the target element $x$. To illustrate the effect of oracles, let's replace some of them with unit operations, we write

$$|\Psi_x^{i,R}\rangle = \prod_{j=i+1}^{R} U_j O_x \prod_{j=1}^{i} U_j I |\psi\rangle, \tag{2.2.2}$$

which uses the identity for the first $i$ oracle queries and the oracle for the latter $R - i$ oracle queries. If there is no application oracle, then the state is independent of the target element and may be denoted as

$$|\Psi^R\rangle = \prod_{j=1}^{R} U_j I |\psi\rangle. \tag{2.2.3}$$

So that we can find the distance changed by oracle, here we use two kinds of metric, euclidean distance and angular distance:

$$E(|\alpha\rangle, |\beta\rangle) = \||\alpha\rangle - |\beta\rangle\| \quad \text{and} \quad A(|\alpha\rangle, |\beta\rangle) = \arccos \frac{|\langle\alpha|\beta\rangle|}{\||\alpha\rangle\| \, \||\beta\rangle\|}. \tag{2.2.4}$$

Taking the distance $E$ as an example, if $\Psi_x^R$ satisfies the properties the output should have

$$\left\|\Pi_x |\Psi_x^R\rangle\right\|^2 \geq p \tag{2.2.5}$$

which $\{\Pi_j\}$ is a finite set of orthogonal projectors that sum to the identity and $p$ is the probability we want of outputting the correct element.

We can calculate the number of queries $R$ by dividing $E(|\Psi_x^R\rangle, |\Psi^R\rangle)$ with the distance changed by oracle[1], that is

$$R = \frac{E(|\Psi_x^R\rangle, |\Psi^R\rangle)}{E(O_x |\gamma\rangle, |\gamma\rangle)}. \tag{2.2.6}$$

It is important to note that $|\gamma\rangle$ here is not some particular state, we will calculate the mean of the distance changed by oracle on each state.

---

[1]Note that this is just a loose intuition, which we'll prove in the next subsection.

### 2.2.2  Distance the oracle changes

First, let's calculate the distance changed by the oracle. Considering the distance of the state before and after $R$ oracle queries, we have

**Lemma 2.2.1.** *The average euclidean distance after $R$ oracle queries is at most*

$$\frac{1}{N}\sum_{x=1}^{N} E(|\Psi_x^R\rangle, |\Psi^R\rangle) \leq 2R\frac{1}{\sqrt{N}}. \tag{2.2.7}$$

*Proof.* With triangle inequality,

$$\frac{1}{N}\sum_{x=1}^{N} E(|\Psi_x^R\rangle, |\Psi^R\rangle) = \frac{1}{N}\sum_{x=1}^{N} E(|\Psi_x^{0,R}\rangle, |\Psi_x^{R,R}\rangle) \leq \frac{1}{N}\sum_{x=1}^{N}\sum_{i=1}^{R} E(|\Psi_x^{i-1,R}\rangle, |\Psi_x^{i,R}\rangle)$$

$$= \frac{1}{N}\sum_{i=1}^{R}\sum_{x=1}^{N} E(O_x|\Psi^i\rangle, |\Psi^i\rangle) = \frac{1}{N}\sum_{i=1}^{R}\sum_{x=1}^{N} 2\left\|\Pi_x|\Psi^i\rangle\right\|$$

$$\leq 2\sum_{i=1}^{R}\frac{1}{\sqrt{N}} = 2R\frac{1}{\sqrt{N}},$$

where the last inequality that needs to be noticed is that

$$\left(\sum_{i=1}^{N} a_i\right)^2 = \sum_{i=1}^{N} a_i^2 + 2\sum_{i=1}^{N}\sum_{j=1}^{i-1} a_i a_j \leq N\sum_{i=1}^{N} a_i^2, \tag{2.2.8}$$

that is

$$\sum_{x=1}^{N}\frac{1}{N}\left\|\Pi_x|\Psi^i\rangle\right\| \leq \sqrt{\frac{1}{N}\sum_{x=1}^{N}\left\|\Pi_x|\Psi^i\rangle\right\|^2} = \frac{1}{\sqrt{N}}. \qquad\square$$

**Lemma 2.2.2.** *The average angular distance after $R$ oracle queries is at most*

$$\frac{1}{N}\sum_{x=1}^{N} A(|\Psi_x^R\rangle, |\Psi^R\rangle) \leq 2R\arcsin(\frac{1}{\sqrt{N}}). \tag{2.2.9}$$

*Proof.* With triangle inequality,

$$\frac{1}{N}\sum_{x=1}^{N} A(|\Psi_x^R\rangle, |\Psi^R\rangle) = \frac{1}{N}\sum_{x=1}^{N} A(|\Psi_x^{0,R}\rangle, |\Psi_x^{R,R}\rangle) \leq \frac{1}{N}\sum_{x=1}^{N}\sum_{i=1}^{R} A(|\Psi_x^{i-1,R}\rangle, |\Psi_x^{i,R}\rangle)$$

$$= \frac{1}{N}\sum_{i=1}^{R}\sum_{x=1}^{N} A(O_x|\Psi^i\rangle, |\Psi^i\rangle) = \frac{1}{N}\sum_{i=1}^{R}\sum_{x=1}^{N} \arccos(\left|\cos(2\theta_x^i)\right|)$$

$$\leq 2\sum_{i=1}^{R}\arcsin(\frac{1}{\sqrt{N}}) = 2R arcsin(\frac{1}{\sqrt{N}}),$$

where has $\theta_x^i = A(|\Psi^i\rangle, |x_\perp\rangle) = \arcsin\left\|\Pi_x|\Psi^i\rangle\right\|$. And where the last inequality still needs to notice the rescaling provided by equation 2.2.8, that is

$$\sum_{x=1}^{N}\frac{1}{N}\arcsin\left\|\Pi_x|\Psi^i\rangle\right\| \leq \arcsin\sum_{x=1}^{N}\frac{1}{N}\left\|\Pi_x|\Psi^i\rangle\right\| \leq \arcsin\frac{1}{\sqrt{N}},$$

where also need notice that $\sum_{x=1}^{N}\frac{1}{N}\arcsin a_i \leq \arcsin\sum_{x=1}^{N}\frac{1}{N}a_i$ which $a_i \in \left[0, \frac{\pi}{2}\right]$. $\qquad\square$

According to the theorem, the distance changed by each oracle query can only add up linearly.

### 2.2.3 Distance when the output is we need

Then, let's calculate the distance when equation 2.2.5 is satisfied.

**Lemma 2.2.3.** *Suppose that the algorithm correctly outputs $y$ with probability at least $p$ after $R$ queries, given oracle $O_y$. Then the average Euclidean distance is at least*

$$\frac{1}{N} \sum_{x=1}^{N} E(|\Psi_x^R\rangle, |\Psi^R\rangle) \geq \frac{1}{\sqrt{2}} \left( \sqrt{p} - \sqrt{1-p} + 1 - \frac{2}{\sqrt{N}} \right). \tag{2.2.10}$$

*Proof.* For a certain $x$, we have

$$
\begin{aligned}
E(|\Psi_x^R\rangle, |\Psi^R\rangle) &\geq \frac{1}{\sqrt{2}} \left( \left\| \Pi_x \left( |\Psi_x^R\rangle - |\Psi^R\rangle \right) \right\| + \left\| \Pi_x^\perp \left( |\Psi_x^R\rangle - |\Psi^R\rangle \right) \right\| \right) \\
&\geq \frac{1}{\sqrt{2}} \left( \left\| \Pi_x |\Psi_x^R\rangle \right\| - \left\| \Pi_x |\Psi^R\rangle \right\| + \left\| \Pi_x^\perp |\Psi^R\rangle \right\| - \left\| \Pi_x^\perp |\Psi_x^R\rangle \right\| \right) \\
&\geq \frac{1}{\sqrt{2}} \left( \left\| \Pi_x |\Psi_x^R\rangle \right\| - \left\| \Pi_x^\perp |\Psi_x^R\rangle \right\| + 1 - 2 \left\| \Pi_x |\Psi^R\rangle \right\| \right) \\
&\geq \frac{1}{\sqrt{2}} \left( \sqrt{p} - \sqrt{1-p} + 1 - 2 \left\| \Pi_x |\Psi^R\rangle \right\| \right),
\end{aligned}
$$

where the last inequality from the success probability is at least $p$. To eliminate the last term, consider a traversal of $x$

$$\frac{1}{N} \sum_{x=1}^{N} E(|\Psi_x^R\rangle, |\Psi^R\rangle) \geq \frac{1}{\sqrt{2}} \left( \sqrt{p} - \sqrt{1-p} + 1 - \frac{2}{N} \sum_{x=1}^{N} \left\| \Pi_x |\Psi^R\rangle \right\| \right) \geq \frac{1}{\sqrt{2}} \left( \sqrt{p} - \sqrt{1-p} + 1 - \frac{2}{\sqrt{N}} \right),$$

where the last inequality uses the equation 2.2.8 again. $\square$

**Lemma 2.2.4.** *Suppose that the algorithm correctly outputs $y$ with probability at least $p$ after $R$ queries, given oracle $O_y$. Then the average angular distance is at least*

$$\frac{1}{N} \sum_{x=1}^{N} A(|\Psi_x^R\rangle, |\Psi^R\rangle) \geq \arcsin \sqrt{p} - \arcsin \frac{1}{\sqrt{N}}. \tag{2.2.11}$$

*Proof.* For a certain $x$, we have

$$
\begin{aligned}
A(|\Psi_x^R\rangle, |\Psi^R\rangle) &= \arccos \left( |\langle \Psi_x^R | \Psi^R \rangle| \right) \\
&= \arccos \left( \langle \Psi_x^R | (\Pi_x + \Pi_x^\perp)^\dagger (\Pi_x + \Pi_x^\perp) |\Psi^R\rangle \right) \\
&= \arccos \left( \left\| \Pi_x |\Psi_x^R\rangle \right\| \cdot \left\| \Pi_x |\Psi^R\rangle \right\| + \left\| \Pi_x^\perp |\Psi_x^R\rangle \right\| \cdot \left\| \Pi_x^\perp |\Psi^R\rangle \right\| \right) \\
&= \arccos \left( \sin \phi_x^R \sin \theta_x^R + \cos \phi_x^R \cos \theta_x^R \right) \\
&= \phi_x^R - \theta_x^R \geq \arcsin \sqrt{p} - \theta_x^R,
\end{aligned}
$$

where has $\phi_x^i = \arcsin \left\| \Pi_x |\Psi_x^i\rangle \right\|$ and still has $\theta_x^i = \arcsin \left\| \Pi_x |\Psi^i\rangle \right\|$, and where the last inequality from the success probability being at least $p$. Just like the last proof, consider a traversal of $x$

$$\frac{1}{N} \sum_{x=1}^{N} A(|\Psi_x^R\rangle, |\Psi^R\rangle) \geq \arcsin \sqrt{p} - \frac{1}{N} \sum_{x=1}^{N} \theta_x^R \geq \arcsin \sqrt{p} - \arcsin \frac{1}{\sqrt{N}},$$

where the last inequality uses the equation 2.2.8 again. $\square$

### 2.2.4 Lower bound of Grover iterations

From section 2.2.2 and section 2.2.3, we obtain the two parts required for equation 2.2.6 to compute the number of iterations: the upper bound of the distance changed by oracle and the lower bound of the distance when the output is we need. So we have the lower bound on the number of iterations for both versions

**Theorem 2.2.1.** *The unordered search problem with success probability $p > 0$ requires at least $R$ queries, where*

$$R = \frac{\sqrt{N}}{2\sqrt{2}} \left( \sqrt{p} - \sqrt{1-p} + 1 - \frac{2}{\sqrt{N}} \right). \tag{2.2.12}$$

**Theorem 2.2.2.** *The unordered search problem with success probability $p > 0$ requires at least $R$ queries, where*

$$R = \frac{\arcsin \sqrt{p} - \arcsin \frac{1}{\sqrt{N}}}{2 \arcsin \frac{1}{\sqrt{N}}}. \tag{2.2.13}$$

In the case of Euclidean distances, we conclude that Grover's algorithm is asymptotically optimal, and in the case of angular distances, Grover's algorithm is exactly optimal.

## 2.3   Long's algorithm

*This section is based on the paper[3].*

### 2.3.1   Main idea

Although Grover's algorithm is already accurate enough when the number of elements is large, for instance in deciphering the DES code where $N = 256$ the deviation is only $3 \times 10^{-9}$. But in problems where certainly is vital, especially when the dimension is not so big, using a searching algorithm with certainty becomes important. Let's make some improvements to the Grover's algorithm.

In the Grover's algorithm[1] we construct rotations in the space spanned by $|\alpha\rangle$ and $|\beta\rangle$, we wasted the degree of freedom of their relative phase, so we may as well extend[2] $G = DO = (I - 2|\psi_\perp\rangle\langle\psi_\perp|)(I - 2|\beta\rangle\langle\beta|)$ to

$$
\begin{aligned}
I_D &= I + (e^{-i\phi} - 1)|\psi_\perp\rangle\langle\psi_\perp| = e^{-i\phi}(I + (e^{i\phi} - 1)|\psi\rangle\langle\psi|) \\
I_O &= I + (e^{i\phi} - 1)|\beta\rangle\langle\beta| \\
Q &= I_D I_O = e^{-i\phi}H^{\otimes n}(I + (e^{i\phi} - 1)|0\rangle\langle0|)H^{\otimes n}(I + (e^{i\phi} - 1)|\beta\rangle\langle\beta|).
\end{aligned}
\tag{2.3.1}
$$

Intuitively, if we want $Q$ to make $|\psi\rangle$ go to $|\beta\rangle$, we should have the axis of rotation at the same angle as $|\psi\rangle$ and $|\beta\rangle$. From corollary 1.1.5, if the angle of $I_O$ is $-\phi$, the angle of $I_W$ has to be $\phi$. As a generalization of Grover's algorithm, it can be seen that if we set $\phi = \pi$, then $Q$ degenerates to $G$.
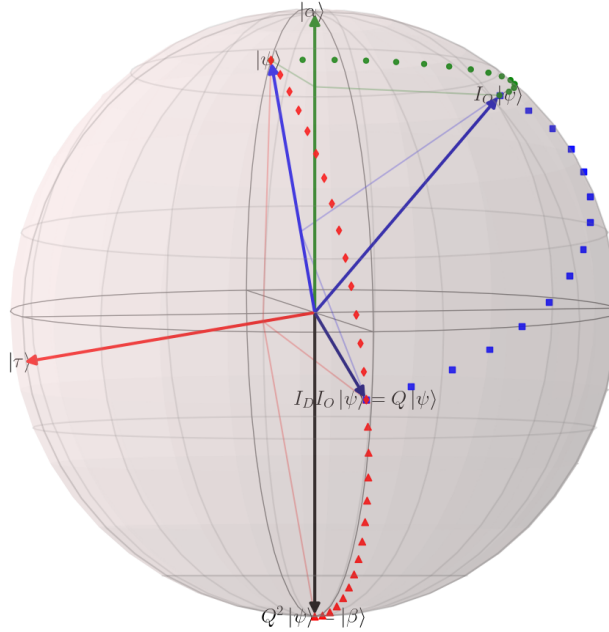


Figure 2.2

As shown in the Figure 2.2, $I_O$ gives us a rotation of angle $-\phi$ about axis $\vec{r}_\alpha = (0, 0, 1)^T$ and $I_W$ gives us a rotation of angle $\phi$ about axis $\vec{r}_\psi = (\sin\theta_G, 0, \cos\theta_G)^T$, their product $Q = I_W I_O$ gives us a rotation of angle

$$
\omega = 4\arcsin(\sin\frac{\phi}{2}\sin\frac{\theta_G}{2})
\tag{2.3.2}
$$

about axis

$$
\vec{r}_\tau = k(\cos\frac{\phi}{2}, \sin\frac{\phi}{2}, \cos\frac{\phi}{2}\tan\frac{\theta_G}{2})^T,
\tag{2.3.3}
$$

where $k$ is the normalization factor.

A naive thought is whether we can find a suitable $\phi$ such that $|\psi\rangle$ can turn to $|\beta\rangle$ exactly. The angle we need to rotate about axis $|\tau\rangle$ is

$$
e^{i\Delta\varphi} = \frac{\langle\tau_\perp|\beta\rangle}{\langle\tau_\perp|\psi\rangle}
\tag{2.3.4}
$$

where $|\tau_\perp\rangle = -|\alpha\rangle\langle\beta|\tau\rangle + |\beta\rangle\langle\alpha|\tau\rangle$, we wish $j = \frac{\Delta\varphi}{\omega}$ is an integer, so that we have $Q^j|\psi\rangle = |\beta\rangle$.

---

[1]The notation of section 2.1.1 is followed here.
[2]For convenience, the oracle can be written as $(I - 2|\beta\rangle\langle\beta|)$ when discussing only in the subspace spanned by $|\alpha\rangle$ and $|\beta\rangle$.

### 2.3.2 The right angle

Using equation 2.3.4 to calculate the angle $\Delta\varphi$ is complicated, so we may as well calculate it with Bloch vectors.
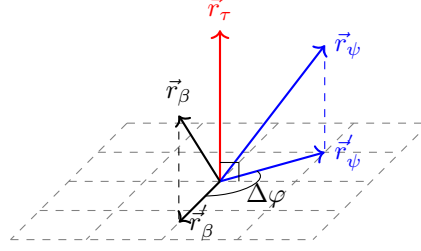


Figure 2.3

As shown in the Figure 2.3, we can calculate the angle between the projections of $\vec{r}_\psi$ and $\vec{r}_\beta$ on the plane perpendicular to $\vec{r}_\tau$, we call these projections

$$\vec{r}_\psi' = \vec{r}_\psi - (\vec{r}_\psi \cdot \vec{r}_\tau)\vec{r}_\tau$$
$$\vec{r}_\beta' = \vec{r}_\beta - (\vec{r}_\beta \cdot \vec{r}_\tau)\vec{r}_\tau,$$

(2.3.5)

so that we can express $\Delta\varphi$ in a simpler form

$$\Delta\varphi = \arccos \frac{\vec{r}_\psi' \cdot \vec{r}_\beta'}{\|\vec{r}_\psi'\|\|\vec{r}_\beta'\|}.$$

(2.3.6)

After some boring calculations, we can get

$$\Delta\varphi = 2\arccos(\sin\frac{\phi}{2}\sin\frac{\theta_G}{2}).$$

(2.3.7)

After calculating each part we need, let's find the right angle $\phi$! Though section 2.3.1, The iterations $j = \frac{\Delta\varphi}{\omega}$ must be an integer. For equation $\Delta\varphi = j\omega$, consider both ends of the equation separately, we have

$$\Delta\varphi = 2\arccos(\sin\frac{\phi}{2}\sin\frac{\theta_G}{2})$$
$$= 2\left(\frac{\pi}{2} - \arcsin(\sin\frac{\phi}{2}\sin\frac{\theta_G}{2})\right)$$
$$j\omega = 4j\arcsin(\sin\frac{\phi}{2}\sin\frac{\theta_G}{2}),$$

(2.3.8)

so that we have

$$\sin\frac{\phi}{2} = \frac{\sin\left(\frac{\pi}{4j+2}\right)}{\sin\frac{\theta_G}{2}},$$

(2.3.9)

notice that $\phi$ has a real solution only when $\theta_G/2 \geq \pi/(4j+2)$, may as well let $j := \lceil(\pi - \theta_G)/(2\theta_G)\rceil$, so we worked out the right angle $\phi$

$$\phi := 2\arcsin\left(\frac{\sin\left(\frac{\pi}{4j+2}\right)}{\sin\frac{\theta_G}{2}}\right).$$

(2.3.10)

Let's finish with some examples of $j$ and $\phi$:

| $N$ | 2 | 4 | 8 | 100 | 1000 | $10^4$ | $10^6$ | $10^8$ | $10^{10}$ | $2^{56}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $j$ | 1 | 1 | 2 | 3 | 8 | 25 | 79 | 785 | 7854 | 78540 |
| $\phi/\pi$ | 0.5 | 1 | 0.677007 | 0.698709 | 0.748018 | 0.854022 | 0.90089 | 0.989752 | 0.992688 | 0.9973 |

Table 2.1

Can be seen as $N$ increases, the angle $\phi$ approaches $\pi$. At this point, Long's algorithm is also slightly closer to Grover's algorithm.

## 2.4   Abdulrahman's work

*This section is based on the paper[1], Here we restrict ourselves to the case $M = 1$.*

### 2.4.1   Main idea

From Equation 2.1.14 we can see that when $N$ is large enough, the iterations have $R \propto 1/\theta_G$, so we can try to find a larger angle to reduce the number of iterations. To do this, we need a larger state space, which we might as well augment with state

$$|\overline{\psi}\rangle := H^{\otimes n}(|0\rangle^{\otimes n-1} \otimes |1\rangle), \tag{2.4.1}$$

and we orthogonalize it to get the orthogonal basis $(|\psi\rangle, |\psi_\perp\rangle, |\gamma\rangle)$,

$$\cos\frac{\omega}{2}|\gamma\rangle := |\overline{\psi}\rangle - \sin\frac{\omega}{2}|\psi_\perp\rangle, \tag{2.4.2}$$

which $\langle\psi|\overline{\psi}\rangle = 0$ and $\sin(\omega/2) = \langle\psi_\perp|\overline{\psi}\rangle = \pm\sqrt{\frac{1}{N-1}}$, here the signs correspond to the 0 and 1 of the last bit of the search element, and we will first discuss the case where the last bit is 0, that is, the sign is positive. In the same way, we have another orthogonal basis $(|\psi\rangle, |\tau\rangle, |\overline{\psi}\rangle)$ which

$$\sin\frac{\omega}{2}|\tau\rangle := |\psi_\perp\rangle - \cos\frac{\omega}{2}|\overline{\psi}\rangle. \tag{2.4.3}$$

So we have three orthogonal bases in this three-dimensional space,

$$(|\overline{\psi}\rangle, |\psi\rangle, |\tau\rangle)\begin{pmatrix} \cos\frac{\omega}{2} & 0 & -\sin\frac{\omega}{2} \\ 0 & 1 & 0 \\ \sin\frac{\omega}{2} & 0 & \cos\frac{\omega}{2} \end{pmatrix}^T = (|\gamma\rangle, |\psi\rangle, |\psi_\perp\rangle) = (|\gamma\rangle, |\alpha\rangle, |\beta\rangle)\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\frac{\theta_G}{2} & -\sin\frac{\theta_G}{2} \\ 0 & \sin\frac{\theta_G}{2} & \cos\frac{\theta_G}{2} \end{pmatrix}.$$



(a) orthogonal basis $(|\overline{\psi}\rangle, |\psi\rangle, |\tau\rangle)$     (b) orthogonal basis $(|\gamma\rangle, |\psi\rangle, |\psi_\perp\rangle)$     (c) orthogonal basis $(|\gamma\rangle, |\alpha\rangle, |\beta\rangle)$
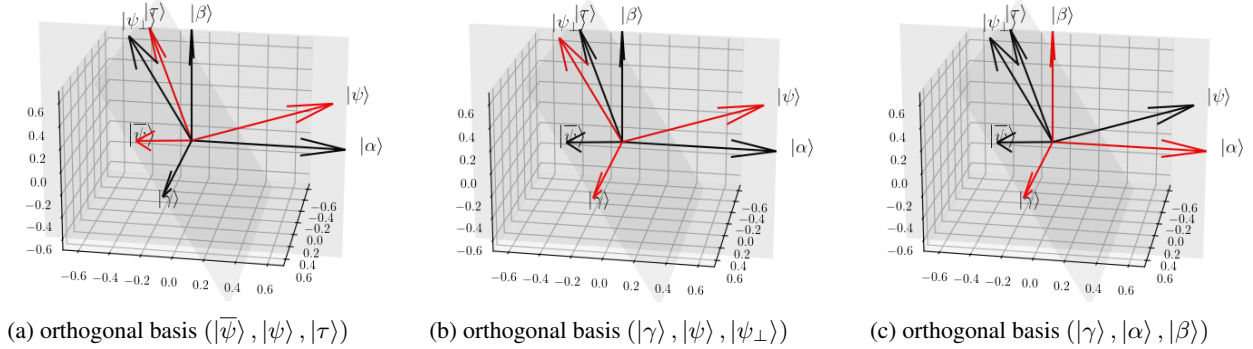
Figure 2.4: Three orthogonal bases

After a basic understanding of this space, we can start to make some improvements. We require a rotation that is independent of the target element, considering the rotation in the space spanned by $|\psi\rangle$ and $|\overline{\psi}\rangle$, in the sub-space $(|\overline{\psi}\rangle, |\psi\rangle, |\tau\rangle)$ we can write this rotation as

$$A(\phi) := H^{\otimes n}X^{\otimes n}c^{n-1}\left(R_y(\phi)\right)X^{\otimes n}H^{\otimes n} = \begin{pmatrix} \cos\frac{\phi}{2} & -\sin\frac{\phi}{2} & 0 \\ \sin\frac{\phi}{2} & \cos\frac{\phi}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.4.4}$$

Our objective is to maximize the probability of finding the target element, notice that

**Theorem 2.4.1.** *Cosidering vectors $\vec{m}, \vec{n} \in \mathbb{R}^3$ and angle $\phi \in [0, 2\pi]$, for a certainly axis defined by vector $\vec{l}$, the inner product $\vec{m} \cdot \left(\exp\left(\phi\vec{l} \cdot \vec{J}\right)\vec{n}\right)$ take the extreme only when* $\text{rank}\left(\left(\vec{m}, \vec{n}, \vec{l}\right)\right) < 3$, *which $J_j$ are the three generators of Lie group $SO(3)$.*

According to this theorem, we can see that the maximum probability is achieved when the state after the rotation is in the plane spanned by $|\gamma\rangle$ and $|\beta\rangle$. So we just need angles $\phi$ such that $A(\phi)$ keeps the states after each iteration in the plane spanned by $|\gamma\rangle$ and $|\beta\rangle$, we extend the Grover iteration $G = DO$ to

$$Q = A(\phi)G = A(\phi)DO. \tag{2.4.5}$$

## 2.4.2 Keep states in the plane

For convenience, we use basis $(|\overline{\psi}\rangle, |\psi\rangle, |\tau\rangle)$ and spherical coordinates $(x, y, z)^T \rightarrow (\sin\frac{\theta}{2}\sin\varphi, \sin\frac{\theta}{2}\cos\varphi, \cos\frac{\theta}{2})^T$ to describe this space. After some calculations, we have

$$|\beta\rangle = (\sin\frac{\theta_\beta}{2}\sin\frac{\pi}{4}, \sin\frac{\theta_\beta}{2}\cos\frac{\pi}{4}, \cos\frac{\theta_\beta}{2})^T, \tag{2.4.6}$$

where $\sin(\theta_\beta/2) = \sqrt{2/N}$, so the space spanned by $|\gamma\rangle$ and $|\beta\rangle$ can be write as $\varphi = \pi/4$. We need to get the initial state into the plane first, so we apply $A_0 := A(-2(\varphi_\beta - \varphi_\psi)) = A(-\pi/2)$ to $|\psi\rangle$ so that

$$A_0|\psi\rangle = (\sin\frac{\pi}{4}, \cos\frac{\pi}{4}, 0)^T. \tag{2.4.7}$$

Next, let's consider the effect of the Grover iteration $G$ on the angle $\varphi$, what can be computed is that the Oracle gives us the mapping

$$O: \frac{\theta}{2} \rightarrow \pi - \frac{\theta}{2} + \theta_\beta \tag{2.4.8}$$
$$\varphi \rightarrow \varphi$$

and the diffusion operation gives us the mapping

$$D: \frac{\theta}{2} \rightarrow \pi - \frac{\theta}{2} \tag{2.4.9}$$
$$\varphi \rightarrow -\varphi.$$

Since the plane gives us the angle $\varphi = \pi/4$, we need to operate $A := A(-2(\varphi - (-\varphi))) = A(-\pi)$ to pull back the angle $\varphi$ to $\pi/4$, so the operation $Q$ gives us the mapping

$$Q = AG: \frac{\theta}{2} \rightarrow \frac{\theta}{2} - \theta_\beta \tag{2.4.10}$$
$$\varphi \rightarrow \varphi.$$

Can be seen that operation $Q$ keeps the state in the plane spanned by $|\gamma\rangle$ and $|\beta\rangle$, so we visualize this plane to further observe the role of operation $Q$, with $\sqrt{1 - \langle\beta|A_0|\psi\rangle^2}|\beta_\perp\rangle = A_0|\psi\rangle - |\beta\rangle\langle\beta|A_0|\psi\rangle$.
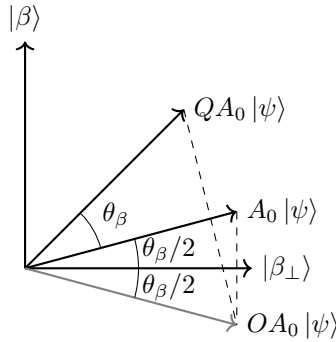


Figure 2.5

As shown in the Figure 2.5, we can see that the operation $Q = ADO$ gives us a rotation of angle $\theta_\beta$, So we just need applying $R_Q = \left\lfloor \frac{\pi - \theta_\beta/2}{\theta_\beta} \right\rfloor$ times operation $Q$ on $A_0|\psi\rangle$ to approximate $|\beta\rangle$, when the number of searches $N$ is large enough,

$$Q^{R_Q}A_0|\psi\rangle = \cos(\frac{2R_Q+1}{2}\theta_\beta)|\beta_\perp\rangle + \sin(\frac{2R_Q+1}{2}\theta_\beta)|\beta\rangle \approx |\beta\rangle, \tag{2.4.11}$$

we have the ratio of the number of iterations after and before refinement $R_Q/R \approx \theta_G/\theta_\beta \approx 1/\sqrt{2}$.

It seems that this violates the lower bound on the number of iterations we computed in section 2.2, but we did restrict the search element. If the last bit of the search element is 1, what can be calculated is $A_0 = A(\pi/2)$ and $A = A(\pi)$, such that we need to distinguish the last digit of the search element first, so we're essentially trading one bit of state space for this $\sqrt{2}$ speedup.

# Bibliography

[1] Ismael Abdulrahman. Enhancing grover's search algorithm: a modified approach to increase the probability of good states. *The Journal of Supercomputing*, 80(12):18048–18061, May 2024.

[2] Catalin Dohotaru and Peter Hoyer. Exact quantum lower bound for grover's problem. *Quantum Information & Computation*, 9(5):533–540, 2009.

[3] G. L. Long. Grover algorithm with zero theoretical failure rate. *Physical Review A*, 64(2), July 2001.

[4] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, December 2010.