

Quantum Computation

Lectures by Richard Jozsa

Contents

0	Review of Shor's Algorithm	2
0.1	Factoring Problem	2
0.2	Quantum Factoring Algorithm Summary	2
0.3	Quantum Algorithm for Periodicity Determination	3
1	The Hidden Subgroup Problem (HSP)	5
2	Quantum Phase Estimation (PE) Algorithm	12
3	Amplitude Amplification	15
4	Hamiltonian Simulation	19
4.1	Hamiltonian Simulation for local Hamiltonians	21

0 Review of Shor's Algorithm

This result is powered by the **quantum period finding algorithm**, and will lead us to the **hidden subgroup problem** (henceforce HSP).

0.1 Factoring Problem

Given an integer N , with $n = O(\log N)$ digits, we want to find a non-trivial factor in time complexity $O(\text{poly}(n))$.

The important concept here is that of **polynomial time complexity**: any computation has an input, from which we obtain an input *size* n . Then by polynomial time complexity, we mean that the number of steps/gates (either classical or quantum) grows only polynomially with n (*i.e.* is $O(\text{poly}(n))$).

When we refer to **efficient** computation, we are always referring to polynomial time complexity.

The best known *classical* factoring algorithm has complexity $e^{O(n^{\frac{1}{3}}(\log n)^{\frac{2}{3}})}$. However, the best known quantum algorithm (due to Shor) runs in $O(n^3)$, a considerable improvement.

0.2 Quantum Factoring Algorithm Summary

First, we convert factoring into period determination:

Given N , choose $a < N$ with $(a, N) = 1$ and consider $f : \mathbb{Z} \rightarrow \mathbb{Z}_N, x \mapsto a^x \bmod N$. Euler's Theorem tells us that f is periodic, and the period r is the order of a modulo N , *i.e.* the least $m > 1$ such that $a^m \equiv 1 \bmod N$ - this exists if and only if a, N are coprime. Through knowledge of r we are able to compute a factor of N .

While the process of determining r is *mathematically* very simple, it is in fact as difficult to compute from a classical perspective as factoring N itself. Instead we use the **Quantum algorithm for periodicity determination**.

The task: Given an oracle/black box for $f : \mathbb{Z}_M \rightarrow \mathbb{Z}_N$ with promises:

- f is periodic, with (unknown) period $r \in \mathbb{Z}_M$, *i.e.* $f(x+r) = f(x)$ for all $x \in \mathbb{Z}_M$.
- f is 1-1 in each period, *i.e.* $f(x_1) \neq f(x_2)$ for any $0 \leq x_1 < x_2 < r$.

We want to find r in time $O(\text{poly}(m))$, $m = \log M$ (with any prescribed success probability $1 - \varepsilon$, $\varepsilon > 0$).

Remark: Queries to the oracle count as 1 step. In the quantum context we assume the oracle is a unitary gate U_f on $\mathcal{U}_M \otimes \mathcal{U}_N$, where \mathcal{U}_M is the state space with dimension M , basis $\{|i\rangle\}_{i \in \mathbb{Z}_M}$. U_f acts on basis states as

$$U_f \underbrace{|i\rangle}_{\text{input}} \underbrace{|j\rangle}_{\text{output}} = |i\rangle |j + f(i)\rangle, \quad i \in \mathbb{Z}_M, j \in \mathbb{Z}_N$$

The **Query complexity** of an algorithm is the number of times the oracle is queried, which is also required to be $O(\text{poly}(m))$.

To solve the periodicity problem classically, it can be shown that it is both necessary and sufficient to query the oracle $O(\sqrt{N})$ times, so there is no polynomial algorithm. However, there *is* a quantum algorithm.

0.3 Quantum Algorithm for Periodicity Determination

For further details *c.f.* Part II notes pp.60-64.

Write $A = M/r = \text{\#periods}$. We work in the state space $\mathcal{U}_M \otimes \mathcal{U}_N$ with basis $\{|i\rangle|k\rangle : i \in \mathbb{Z}_M, k \in \mathbb{Z}_N\}$.

Step 1: obtain the state

$$\frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |i\rangle|0\rangle$$

Step 2: apply U_f to obtain

$$\frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |i\rangle|f(i)\rangle$$

Step 3: measure the output register, obtaining result y . By the **Born rule**, the input register collapses to all those i such that $f(i) = y$, *i.e.* $i = x_0, x_0 + r, \dots, x_0 + (A-1)r$ where $0 \leq x_0 < r$ in the first period has $f(x_0) = y$.

We discard the output register to obtain

$$|\text{per}\rangle = \frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |x_0 + jr\rangle$$

Note that each $0 \leq x_0 < r$ occurs with probability $1/r$.

If we naively measure $|\text{per}\rangle$, the Born rule implies we get $x_0 + jr$ with $j = 0, \dots, A-1$ chosen uniformly with probability $1/A$, *i.e.* a random element of a random period; this is a uniformly random integer in \mathbb{Z}_M . This is useless to us. Instead...

Step 4: apply **Quantum Fourier Transform** (QFT). Recall that

Lecture 2

$$\text{QFT}|x\rangle = \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} \omega^{xy} |y\rangle$$

Fact: QFT modulo M is unitary, and can be implemented in $O(m^2)$ time, $m = \log M$. See Part II QIC notes for circuit details of implementation.

Then

$$\begin{aligned} \text{QFT}|\text{per}\rangle &= \frac{1}{\sqrt{MA}} \sum_{j=0}^{A-1} \left(\sum_{y=0}^{M-1} \omega^{(x_0+jr)y} |y\rangle \right) \\ &= \frac{1}{\sqrt{MA}} \sum_{y=0}^{M-1} \omega^{x_0 y} \left[\sum_{j=0}^{A-1} \omega^{jry} \right] |y\rangle \end{aligned}$$

Note that $[\dots]$ is a geometric series, with ratio $\omega^{ry} = e^{2\pi i r y / M} = (e^{e\pi i / A})^y$. So the sum equals zero unless y is a multiple of $A = M/r$, in which case every term in the sum is 1 so the sum equals A . So the non-multiples of A get sifted out by QFT.

Hence, we have

$$\text{QFT}|\text{per}\rangle = \sqrt{\frac{A}{M}} \sum_{k=0}^{r-1} \omega^{x_0 k M/r} |k \frac{M}{r}\rangle$$

Then measuring $\text{QFT}|\text{per}\rangle$ we get a value $c = k_0 M/r$, with $0 \leq k_0 \leq r-1$ chosen uniformly at random. Thus we have $k_0/r = c/M$, where the values c, M are known and k_0 has been chosen at random; we want r . Note that if we are fortunate enough to have $(k_0, r) = 1$, then we can (efficiently) cancel c/M down to its lowest terms, and read off r as the denominator. But in general this will not be the case:

Theorem: (Coprimality Theorem) *The number of positive integers $< r$ that are coprime to r grows as $O(r/\log \log r)$ for large r .*

Hence the above $\mathbb{P}(k_0 \text{ coprime to } r) = O(1/\log \log r)$. So if we do it enough times, we will almost surely be successful:

Probability Lemma: If a single trial has success probability p , then we repeat k times, and for any $0 < 1 - \varepsilon < 1$, we have that

$$\begin{aligned} \text{if } k &= -\frac{\log \varepsilon}{p} \\ \text{then } \mathbb{P}(\geq 1 \text{ success in } k \text{ trials}) &> 1 - \varepsilon \end{aligned}$$

So after finding c , cancel c/M down to its lowest terms a/b (classically, in polynomial time using Euclid's algorithm). We get r as denominator b if $(k_0, r) = 1$, which happens with probability $O(1/\log \log r)$, otherwise c, M have more common factors, so $b < r$.

We don't know immediately whether that has happened or not, but we can check the b value by making two more queries to the oracle, $f(0)$ and $f(b)$; these are equal iff $b = r$.

So if we repeat this $K = O(\log \log r)$ times, then we will obtain r with any high probability we desire - and this runs in polynomial time.

Origin and utility of QFT here

Write $R = \{0, r, 2r, \dots, (A-1)r\} \subseteq \mathbb{Z}_M$, and

$$\begin{aligned} |R\rangle &= \frac{1}{\sqrt{A}} \sum_{k=0}^{A-1} |kr\rangle \\ |\text{per}\rangle &= |x_0 + R\rangle = \frac{1}{\sqrt{A}} \sum_{k=0}^{A-1} |x_0 + kr\rangle \end{aligned}$$

The problem is that the $|x_0 + kr\rangle$ terms are distributed randomly.

For each $x_0 \in \mathbb{Z}_M$, consider the map $k \mapsto k + x_0$ on \mathbb{Z}_M ; this is the 1-1 reversible map "shift by x_0 ".

This gives rise to a linear map $U(x_0)$ on \mathcal{U}_M , and $U(x_0) : |k\rangle \rightarrow |k + x_0\rangle$ is unitary, and $|x_0 + R\rangle = U(x_0)|R\rangle$.

Since $(\mathbb{Z}_M, +)$ is an *abelian* group, these shift operators all commute, i.e. $U(x_0)U(x_1) = U(x_0 + x_1) = U(x_1)U(x_0)$. So they have an orthonormal basis of common eigenvectors $\{|\chi_k\rangle\}_{k \in \mathbb{Z}_M}$, called the *shift-invariant* states. Note that they are not left entirely unchanged by the $U(x_0)$ operators, but they are shifted only by a constant phase factor, i.e. $U(x_0)|\chi_k\rangle = \omega(x_0, k)|\chi_k\rangle$ for all $x_0, k \in \mathbb{Z}_M$, and $|\omega(x_0, k)| = 1$.

Now consider $|R\rangle$ written in the χ -basis

$$|R\rangle = \sum_{k=0}^{M-1} a_k |\chi_k\rangle$$

where the amplitudes a_k depend only on r , and not on x_0 (obviously). Then $|\text{per}\rangle = U(x_0)|R\rangle = \sum a_k \omega(x_0, k) |\chi_k\rangle$, and measurement in the χ -basis has $\mathbb{P}(k) = |a_k \omega(x_0, k)|^2 = |a_k|^2$, independent of x_0 , depending only on r . So we want to measure in this basis, but aren't allowed to do that (computationally) since the basis is too complicated.

So we introduce QFT as the unitary mapping that rotates the χ_k -basis onto the standard basis $|k\rangle$, and follow this up by a standard basis measurement.

But what does this mapping look like, and where does it come from? We need the explicit form of the shift-invariant eigenstates:

$$\begin{aligned} |\chi_k\rangle &= \frac{1}{\sqrt{M}} \sum_{\ell=0}^{M-1} e^{-2\pi i k \ell / M} |\ell\rangle \\ \Rightarrow \mathcal{U}(x_0) |\chi_k\rangle &= \frac{1}{\sqrt{M}} \sum_{\ell=0}^{M-1} e^{-2\pi i k \ell / M} |\ell + x_0\rangle \\ &= \frac{1}{\sqrt{M}} \sum_{\tilde{\ell}=0}^{M-1} e^{-2\pi i k (\tilde{\ell} - x_0) / M} |\tilde{\ell}\rangle \\ &= e^{2\pi i k x_0 / M} |\chi_k\rangle \end{aligned}$$

So $\omega(x_0, k) = e^{2\pi i k x_0 / M}$.

The matrix of QFT^{-1} (mapping $|k\rangle$ to $|\chi_k\rangle$) has components of $|\chi_k\rangle$ as the k^{th} column, so $[\text{QFT}^{-1}]_{\ell k} = \frac{1}{\sqrt{M}} e^{-2\pi i \ell k / M}$. Since QFT is unitary, to find the inverse we need only take the conjugate transpose.

Hence $[\text{QFT}]_{k\ell} = \frac{1}{\sqrt{M}} e^{2\pi i k \ell / M}$, as previously defined.

This notion of QFT in fact occurs very naturally in group theory as the *discrete Fourier transform*. The fact that this QFT is unitary means that all the group theoretic results it relies on slot in perfectly, allowing us to make as much use of this as we want in a way that we are not able classically.

Lecture 3

The following algorithm was inspired by a desire to generalise the successful technique of Shor's celebrated algorithm.

1 The Hidden Subgroup Problem (HSP)

Let G be a finite group, of size $|G|$.

We are given an oracle $f : G \rightarrow X$ (where X is some set), and a promise that there is a subgroup $K < G$ such that

- f is constant on (left) cosets of K in G
- f is distinct on distinct cosets.

Problem: Determine the “hidden subgroup” K .

For instance, this might entail outputting a set of generators, or we might be happy enough with just sampling uniformly from the elements of K .

* we want to solve this in time $O(\text{poly log } |G|)$, with any constant probability $1 - \varepsilon$.

Examples of problems that can be seen as HSPs:

- (a) Periodicity. $f : \mathbb{Z}_M \rightarrow X$ periodic, period r , bijective within periods.

Then let $G = \mathbb{Z}_M$, $K = \{0, r, 2r, \dots\} < G$, and the cosets are $x_0 + K = \{x_0, x_0 + r, x_0 + 2r, \dots\}$. It is then clear that f is constant/distinct on the cosets in the desired way.

- (b) Discrete Logarithms: this was also solved by Shor in his original paper.

Take p prime, and consider \mathbb{Z}_p^* the group of units modulo $p = \{1, 2, \dots, p-1\}$. We say that $g \in \mathbb{Z}_p^*$ is a *generator*, or a *primitive root modulo p* , if the powers of g generate all of \mathbb{Z}_p^* .

Fact: generators always exist, *i.e.* \mathbb{Z}_p^* is always cyclic. For instance, 2, 3 both generate \mathbb{Z}_5^* - though 1, 4 do not.

So any $x \in \mathbb{Z}_p^*$ can be written as $x = g^y$ for $y \in \mathbb{Z}_{p-1}$. We thus write $y = \log_g x$ for the **discrete logarithm** of x , to base g .

The discrete log problem is then: given a generator g and $x \in \mathbb{Z}_p^*$, compute $y = \log_g x$. This is very difficult classically, and it underpins public key cryptography.

To express this as a hidden subgroup problem, consider $f : \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1} \rightarrow \mathbb{Z}_p^*$ by $f(a, b) = g^a x^{-b} = g^{a-yb} \pmod{p}$. Then (check) $f(a_1, b_1) = f(a_2, b_2)$ iff $(a_2, b_2) = (a_1, b_1) + \lambda(y, 1)$, $\lambda \in \mathbb{Z}_{p-1}$.

So we let $G = \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}$, and $K = \{\lambda(y, 1) : \lambda \in \mathbb{Z}_{p-1}\} < G$. Then f is constant/distinct as appropriate on the cosets of K , and the generator $(y, 1)$ of K gives $y = \log_g x$.

- (c) Graph Problems.

Consider a graph $A = (V, E)$, $|V| = n$. We stipulate that these are undirected, that there is at most one edge between any pair of vertices, and that the vertices are labelled by $[n] = \{1, 2, \dots, n\}$. We also might be interested in the adjacency matrix M_A , the $n \times n$ matrix given by $[M_A]_{ij} = \mathbb{I}[\{i, j\} \in E]$, which is always symmetric for an undirected graph.

The group that will be of interest to us is $P_n :=$ the permutation group of $[n]$. So $|P_n| = n! \sim \sqrt{2\pi n}(n/e)^n$ has $|P_n| \sim O(n \log n) < O(n^2)$, which is polynomial in the number of vertices. This is what we want for the running time of a graph algorithm.

The subgroup of interest is $\text{Aut}(A)$, the **automorphism group** of $A < P_n$ the set of permutations $\pi \in P_n$ such that for all i, j , $\{i, j\} \in E$ iff $\{\pi(i), \pi(j)\} \in E$. What this means is that after permuting the labels of the graph, we are left with the same labelled graph.

An associated HSP (non-abelian G):

Take $G = P_n$, and $X =$ the set of all labelled graphs on n vertices (equivalent to the set of all symmetric $n \times n$ 0/1-matrices).

For any $A \in X$, we consider $f_A : G \rightarrow X$, with $f_A(\pi) = \pi(A)$, *i.e.* the graph A with its vertex labels permuted by π . A little bit of thought shows that $K = \text{Aut}(A)$ is the hidden subgroup for this problem; f_A is constant on the automorphisms of A , for instance.

An important application of this is that if we can sample uniformly from K , then we can solve the **Graph Isomorphism Problem** (GI), which has received a lot of attention in complexity theory in recent years.

Two labelled graphs A, B each on n vertices are **isomorphic** if there is a bijective map (permutation) on the labels $\pi : [n] \rightarrow [n]$ such that for all $i, j \in [n]$, $\{i, j\} \in A$ iff $\{\pi(i), \pi(j)\} \in B$. In other words, A, B are the same underlying graph (*i.e.* ignoring their labels they are indistinguishable). We write $A \cong B$.

The GI Problem is then: given graphs A and B , determine whether or not they are isomorphic. This has many useful applications; *e.g.* if you can see some proteins and their structure, you may want to be able to tell which proteins are actually the same.

This can again be expressed as a non-abelian HSP, *c.f.* Sheet 1.

There is no known polynomial time classical algorithm, and in fact there is no known polynomial time quantum algorithm either. The problem is in NP, but is *not* believed to be NP-complete. A problem is NP if it is, roughly speaking, difficult to solve but easy to validate that you have the right answer once you've solved it. A problem is NP-complete if you can rephrase any other NP problem as this one, and then solve it that way - so solving an NP-complete problem solves all NP problems; it is the hardest problem in NP.

We currently do not believe that even quantum algorithms are able to solve NP-complete problems efficiently, so it is in some sense hopeless to try and work on these problems even from a quantum perspective. However, they can do NP-incomplete problems, so factoring and GI *etc.* are good candidates to attempt.

Laslo Babai (2017) found a *quasi*-polynomial time *classical* algorithm for GI; it has runtime $n^{O((\log n)^2)}$. This is slower than polynomial time, but faster than exponential time. We have the following hierarchy:

$$\begin{aligned} \text{poly}(n) &< n^{O((\log n)^2)} < \exp \\ 2^{O(\log n)} &< 2^{O((\log n)^3)} < 2^{O(n)} \end{aligned}$$

So in terms of exponents, these are linear/polynomial/exponential in $\log n$.

- (d) Another non-abelian example is the ***dihedral group***; there is a connection to the HSP 'shortest vector in a lattice'. We are given n linearly independent vectors in \mathbb{R}^n , and consider their lattice; the problem is to find the lattice point closest to the origin.

Lecture 4

What we *do* have is:

Quantum Algorithm for Finite Abelian HSP

We write the group $(G, +)$, additively. We need a couple of components:

Construction of shift invariant states & Fourier transform for G

Definition: (Representation of G) A group homomorphism $\chi : G \rightarrow \mathbb{C}^* = (\mathbb{C} \setminus \{0\}, \cdot)$ is called a ***representation*** of G .

For abelian groups, any such map is called an ***irreducible representation*** (henceforth *irrep*) of G .

These have the following properties:

Theorem: (Theorem A)

- (i) any value $\chi(g)$ is a $|G|^{th}$ root of unity (so $\chi : G \rightarrow S^1$, the unit circle in \mathbb{C})
- (ii) (Schur's Lemma/Orthogonality) If χ_i and χ_j are representations, then

$$\frac{1}{|G|} \sum_{g \in G} \chi_i(g) \overline{\chi_j(g)} = \delta_{ij}$$

(iii) There are exactly $|G|$ different representations of G .

By (iii) we can label the χ 's as $\chi_g : g \in G$.

Example: $\chi(g) = 1$ for all $g \in G$ is always an irrep, called the *trivial* irrep; label it as χ_0 for $0 \in G$.

Then for any other irrep $\chi \neq \chi_0$, orthogonality to χ_0 gives

$$\sum_{g \in G} \chi(g) = 0$$

We now introduce:

Definition: (Shift-Invariant States, Shift Operators) Consider the state sapce \mathcal{H}_G , dimension $|G|$, basis $|g\rangle : g \in G$.

The *shift operators* are

$$U(k) : |g\rangle \mapsto |g + k\rangle, \quad k, g \in G$$

These all commute, so we have a simultaneous eigenbasis:

For each $\chi_k, k \in G$, consider the state

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

By Theorem A (ii), these form an orthonormal basis, and it follow from the group homomorphism property of irreps that these are in fact the simultaneous eigenbasis, $U(g)|\chi_k\rangle = \chi_k(g)|\chi_k\rangle$. These are the *shift-invariant states*.

Proof. We have that

$$\begin{aligned} U(g)|\chi_k\rangle &= \frac{1}{\sqrt{|G|}} \sum_{h \in G} \overline{\chi_k(h)} |g + h\rangle \\ &= \frac{1}{\sqrt{|G|}} \sum_{h' \in G} \overline{\chi_k(h' - g)} |h'\rangle \\ &= \frac{1}{\sqrt{|G|}} \chi_k(g) \sum_{h' \in G} \overline{\chi_k(h')} |h'\rangle \\ &= \chi_k(g) |\chi_k\rangle \end{aligned}$$

Which follows since $\chi_k(g^{-1}) = \overline{\chi_k(g)}$. □

Definition: (Fourier Transform QFT) The unitary gate on \mathcal{H}_G mapping $|\chi_g\rangle$ basis to the $|g\rangle$ basis is known as the *Quantum Fourier Transform (QFT)* i.e.

$$\begin{aligned} \text{QFT}|\chi_g\rangle &= |g\rangle, \quad \forall g \in G \\ \text{QFT}^{-1}|g\rangle &= |\chi_g\rangle \end{aligned}$$

In particular, the k^{th} column of QFT^{-1} in $|g\rangle$ -basis is given by components of $|\chi_k\rangle$, i.e.

$$[\text{QFT}^{-1}]_{gk} = \frac{1}{\sqrt{|G|}} \overline{\chi_k(g)} \quad \forall g, k \in G$$

so QFT (as the conjugate transpose) has the matrix

$$\begin{aligned} [QFT]_{kg} &= \frac{1}{\sqrt{|G|}} \chi_k(g) \\ \implies QFT|g\rangle &= \frac{1}{\sqrt{|G|}} \sum_{k \in G} \chi_k(g) |k\rangle \end{aligned}$$

Examples: $G = \mathbb{Z}_M$ Check $\chi_a(b) = e^{2\pi i ab/M}$ for all $a, b \in \mathbb{Z}_M$ satisfies (Hom), so we have irreps naturally labelled by $a \in \mathbb{Z}_M$, $\chi_0(b) = 1$ for all b , giving the ‘usual’ QFT_M for \mathbb{Z}_M .

Similarly for $G = \mathbb{Z}_{M_1} \times \cdots \times \mathbb{Z}_{M_r}$, where we have

$$\begin{aligned} \left. \begin{aligned} (a_1, \dots, a_r) &= g_1 \\ (b_1, \dots, b_r) &= g_2 \end{aligned} \right\} \in G \\ \chi_{g_1}(g_2) &:= e^{2\pi i \left(\frac{a_1 b_1}{M_1} + \cdots + \frac{a_r b_r}{M_r} \right)} \end{aligned}$$

which satisfies (Hom), and we get

$$QFT_G = QFT_{M_1} \otimes \cdots \otimes QFT_{M_r}$$

Importantly, the above is exhaustive due to the classification of all finite abelian groups, which states that they are all isomorphic to a direct product $G \cong \mathbb{Z}_{M_1} \otimes \cdots \otimes \mathbb{Z}_{M_r}$, and furthermore we can take all the M_i ’s to be (not necessarily distinct) prime powers.

Quantum Algorithm for Finite Abelian HSP

We have $f : G \rightarrow X$, hidden subgroup K , cosets $K = 0 + K, g_2 + K, \dots, g_m + K$, where $m = |G|/|K|$ is the number of cosets.

This works on the state space $\mathcal{H}_{|G|} \otimes \mathcal{H}_{|X|}$, with basis $|g\rangle|x\rangle$ for $g \in G, x \in X$.

- Make the initial state

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

- Apply \mathcal{U}_f , to obtain

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle|f(g_0)\rangle$$

- Measure the second register to see a value $f(g_0)$. Then the first register will give a coset state:

$$|g_0 + K\rangle = \frac{1}{\sqrt{|K|}} \sum_{k \in K} |g_0 + k\rangle = U(g_0)|K\rangle$$

Where we are writing $|K\rangle = \frac{1}{\sqrt{|K|}} \sum_{k \in K} |k\rangle$.

Here the coset has been chosen uniformly at random from all $|G|/|K|$ cosets.

- Apply QFT and measure, to obtain a result $g \in G$.

How does the output g relate to K ? Observe that

- The output distribution of g is independent of g_0 , so it is the same as that obtained from $\text{QFT}|K\rangle$, (i.e. $g_0 = 0$).

This is since if we write $|K\rangle$ in the shift-invariant basis $|\chi_g\rangle$, $|K\rangle = \sum_{g \in G} a_g |\chi_g\rangle$, then $|g_0 + K\rangle = U(g_0)|K\rangle = \sum a_g \chi_g(g_0) |\chi_g\rangle$. But $\text{QFT}|\chi_g\rangle = |g\rangle$, so after QFT we have that $\mathbb{P}[g] = |a_g \chi_g(g_0)|^2 = |a_g|^2$ as $|\chi_g(g_0)| = 1$. Hence we have independence of g_0 .

But what is the distribution?

- How does it relate to K ?

Lecture 5

Recall that

$$\text{QFT}|k\rangle = \frac{1}{\sqrt{|G|}} \sum_{\ell \in G} \chi_\ell(k) |\ell\rangle$$

In particular

$$\text{QFT}|K\rangle = \frac{1}{\sqrt{|G|}} \frac{1}{\sqrt{K}} \sum_{\ell \in G} \left[\sum_{k \in K} \chi_\ell(k) \right] |\ell\rangle$$

where $[\dots]$ involves irreps χ_ℓ of G restricted to $K < G$, which are irreps of K . Hence

$$[\dots] = \begin{cases} |K| & \text{if } \chi_\ell \text{ restricts to trivial irrep on } K \\ 0 & \text{otherwise} \end{cases}$$

So the measurement gives a uniformly random choice of ℓ , such that $\chi_\ell(k) = 1$ for all $k \in K$, giving information about K .

For instance, if K has generators k_1, \dots, k_M , where $M = O(\log |K|) = O(\log |G|)$ (this is true for all finite groups), then the output has $\chi_\ell(k_i) = 1$ for all i .

It can be shown that if $O(\log |G|)$ such ℓ s are chosen uniformly at random, then with probability $> 2/3$ they suffice to determine the generating set via the equation $\chi_\ell(k) = 1$.

See Sheet 1 # 2 for an example of this problem and calculation.

A trivial but nonetheless useful fact about irreps of G is that they restrict to irreps of K , and in fact we could have a trivial irrep of G that restricts to the trivial irrep of K - as we saw above.

Example: If $G = \mathbb{Z}_{M_1} \times \dots \times \mathbb{Z}_{M_q}$, we had for $\ell = (\ell_1, \dots, \ell_q), g = (b_1, \dots, b_q) \in G$ irreps

$$\chi_\ell(g) = e^{2\pi i \sum_{j=1}^q \left(\frac{\ell_j b_j}{m_j} \right)}$$

So for $k = (k_1, \dots, k_q) \in K$, the equation $\chi_\ell(k) = 1$ becomes

$$\frac{\ell_1 k_1}{m_1} + \dots + \frac{\ell_q k_q}{m_q} \equiv 0 \pmod{1}$$

- that is to say, it is an integer. This is a homogeneous, linear equation in $k = (k_1, \dots, k_q)$ and $O(\log |K|)$ independent such equations determine K as the null space of the linear system.

Remarks on HSP for non-abelian groups G : Disclaimer: there is no known efficient algorithm for this problem.

As before, we can easily generate coset states

$$|g_0 K\rangle = \frac{1}{\sqrt{|K|}} \sum_{k \in K} |g_0 k\rangle$$

where the g_0 are chosen randomly. But there arise new problems with the QFT construction. In particular, there is no basis of shift invariant states because the shift operators do not commute.

So if we apply the Fourier transform to the above state, it no longer works. We can make partial progress though.

Construction of non-abelian FT

Suppose we have d -dimensional representations of G , which are group homomorphisms

$$\chi : G \rightarrow U(d)$$

where the $U(d)$ are $d \times d$ unitary matrices. So $\chi(g_1 g_2) = \chi(g_1) \cdot \chi(g_2)$, where \cdot denotes matrix multiplication.

We then say χ is **irreducible** (so an irrep) if no subspace of \mathbb{C}^d is left invariant by all matrices $\chi(g) : g \in G$. In other words, we cannot simultaneously block diagonalise all the $\chi(g)$ s by a basis change.

Looking back at the abelian case, we can consider the irreps as matrices also, but since all the matrices commute with each other they can all be simultaneously block diagonalised, and reduce to 1-dimensional representations.

A **complete set** of irreps is a set χ_1, \dots, χ_M such that any irrep is unitarily equivalent to one of them, by which we mean $\chi \equiv \chi'$ if $\chi' = V\chi V^{-1}$ for some $V \in U(d)$.

Theorem: (Non-Abelian Generalisation of Theorem A) *If d_1, \dots, d_M are the dimensions of a complete set of irreps χ_1, \dots, χ_M , then*

$$(i) \quad d_1^2 + \dots + d_M^2 = |G|$$

(ii) *Write $\chi_{i,jk}(g)$ for the $(j,k)^{th}$ entry of matrix $\chi_i(g)$. Then we have (Schur's orthogonality)*

$$\sum_{g \in G} \chi_{i,jk}(g) \bar{\chi}_{i',j'k'}(g) = |G| \delta_{ii'} \delta_{jj'} \delta_{kk'}$$

Hence the states

$$|\chi_{i,jk}\rangle \equiv \frac{1}{\sqrt{|G|}} \sum_{g \in G} \bar{\chi}_{i,jk}(g) |g\rangle$$

form an orthonormal basis.

QFT on G is then defined as the unitary map that rotates the $\{|\chi_{i,jk}\rangle\}$ basis into the standard basis $\{|g\rangle\}$.

But the $|\chi_{i,jk}\rangle$ are **not** shift invariant for all $U(g_0)$ s, so measurement of the coset state $|g_0 K\rangle$ in the $|\chi\rangle$ basis has output distribution that is **not** independent of g_0 .

However, a 'partial' shift invariance survives. Instead of performing a complete measurement, we can do an incomplete measurement on only the i labels and not the j .

We call this measurement M_{rep} on $|g_0 K\rangle$ that distinguishes only irrep labels (i values) and **not** all (i,j,k) s, i.e. the measurement outcome i is associated to the d_i^2 -dimensional subspace spanned by $\{|\chi_{i,jk}\rangle\}_{j,k=1,\dots,d_i}$.

Then it can be shown that $\chi_i(g_1 g_2) = \chi_i(g_1) \chi_i(g_2)$ implies that the output distribution of the i -values is independent of g_0 . This gives direct, but incomplete information about K .

For instance, conjugate subgroups K and $L = g_0 K g_0^{-1}$ for some $g_0 \in G$ give the same output distribution of i s.

For efficient HSP algorithm (if we use QFT), we need QFT to be efficiently implementable, i.e. in $\text{poly}(\log |G|)$ times - this is unusual, and a very special thing to ask for.

This is true for abelian groups, and some non-abelian groups, such as P_n - but even for this group there is still no known efficient hidden subgroup algorithm.

Consider \mathbb{Z}_{2^n} ; we have a tower of subgroups $\{0\} < \mathbb{Z}_2 < \mathbb{Z}_4 < \dots < \mathbb{Z}_{2^n}$, and the fast Fourier transform on this group works recursively by determining the transform on the next subgroup using the previous one. There is a similar construction $\{e\} < P_2 < \dots < P_n$. However, there is still no efficient HSP algorithm known.

Known partial results:

For normal subgroups (i.e. $gK = Kg$ for all $g \in G$), we have

Theorem: (Hallgren, Russell, Tashina SIAMJ. Comp. vol32 p916-934 (2003))

Suppose G has QFT efficiently implementable. Then if the hidden subgroup K is a normal subgroup, then there is an efficient quantum HSP algorithm.

In particular, we find $\{|g_0K\rangle\}$ and do M_{rep} on it, and repeat $O(\log |G|)$ times. Then K normal implies the outputs suffice to efficiently determine K .

Theorem: (Effinger, Heyer, Krull, 2004) *For general non-abelian HSP, $M = O(\text{polylog } |G|)$ random coset states $|g_1K\rangle, \dots, |g_MK\rangle$ suffice to determine K .*

In particular, there is always efficient query complexity. Unfortunately, there is no known method to efficiently determine K from the M coset states (see Sheet 1 # 7 for a proof of this theorem).

The way we do this is to use a measurement procedure on $|g_1K\rangle \otimes \dots \otimes |g_MK\rangle$ that takes exponential time in $\log |G|$ to complete.

Lecture 6

2 Quantum Phase Estimation (PE) Algorithm

This is a unifying principle for quantum algorithms, and again uses QFT. It has many applications, e.g. an alternative factoring algorithm to Shor's algorithm, due to A. Kitaev (Sheet 2 #2).

Scenario: we are given a unitary operator in d dimensions, and an eigenstate $|v_\varphi\rangle : U|v_\varphi\rangle = e^{2\pi i\varphi}|v_\varphi\rangle$. We want to estimate φ , with $0 < \varphi < 1$, to n binary bits of precision. In particular we have $\varphi = 0.i_1i_2\dots i_n\dots$ in binary, and we want to determine i_1, \dots, i_n for any given n .

We will need the **controlled- U^k** for integers k :

$$\begin{aligned} CU^k|0\rangle|\xi\rangle &= |0\rangle|\xi\rangle \\ CU^k|1\rangle|\xi\rangle &= |1\rangle(U|\xi\rangle) \end{aligned}$$

Note also that $U^k|v_\varphi\rangle = e^{2\pi ik\varphi}|v_\varphi\rangle$, and $C(U^k) = (CU)^k$.

Remark: Given U as a formula or circuit description, we can readily implement CU e.g. by controlling each gate of the circuit.

But if U is given as a black box (i.e. physical operation) then we need further information in order to implement CU .

We can see this since U as a black box is equivalent to $e^{i\theta}U$ as a black box, but the controlled versions of these gates are different, since the $C\text{-(}e^{i\theta}U\text{)}$ gate will switch global phase on $|0\rangle + |1\rangle$ to a local phase.

It in fact suffices to have an eigenstate $|\alpha\rangle$ with known/specified eigenvalue $e^{i\alpha} : U|\alpha\rangle = e^{i\alpha}|\alpha\rangle$. Then $e^{i\theta}U$ has $\alpha \mapsto \alpha + \theta$.

We will actually want a “generalised controlled- U ”, given for $x \in \mathbb{Z}_n$:

$$|x\rangle|\chi\rangle \mapsto U^x|\chi\rangle$$

We can make it from cU^k as follows. For $x = x_{n-1} \dots x_1 x_0$ in binary... [diagram].

If input $|\chi\rangle = |v_\varphi\rangle$, then we get $e^{2\pi i \varphi x} |x\rangle |v_\varphi\rangle$.

Theorem: (PE) *If the measurements give output $\theta = 0.y_0 y_1 \dots y_{n-1}$, with $\varphi = 0.z_0, \dots, z_{n-1} z_n \dots$ then*

(a) $\mathbb{P}(\theta \text{ is closest } n \text{ binary digit approx to } \varphi) \geq 4/\pi^2 \approx 0.4$

(b) $\mathbb{P}(|\theta - \varphi| \geq \varepsilon)$ is at most $O(1/2^n \varepsilon)$, and we will show it is in fact $\leq 1/2^{n+1} \varepsilon$.

Remark: In Theorem PE (a), we have probability $4/\pi^2$ that all n lines of n -line PE process give correct bits.

But: if we want φ accurate to m bits with probability $1 - \eta$, then use Theorem PE (b) with $\varepsilon = 1/2^m$. Then we'll need $n > m$ lines with $\eta = 1/2^{n+1} \varepsilon$ and $\varepsilon = 1/2^m$, i.e. $n = m + \log(1/\eta) - 1$, where n is the number of lines in the PE algorithm, m is the number of bits we want with high probability, and $\log(1/\eta) - 1$ is the number of additional bits we require to ensure this accuracy.

Lecture 7

Proof. Recall that we have

$$\text{QFT}^{-1}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{-2\pi i y x / 2^n} |y\rangle$$

So for the state $|A\rangle$ used earlier, we have

$$\text{QFT}^{-1}|A\rangle = \frac{1}{2^n} \sum_y \left[e^{2\pi i (\varphi - y/2^n)x} \right] |y\rangle$$

So for the measurement, the probability that we see the n -bit integer $y = y_0 y_1 \dots y_{n-1}$ is

$$\frac{1}{2^{2n}} \left| \sum_{x=0}^{2^n-1} e^{2\pi i (\varphi - y/2^n)x} \right|^2$$

Letting $\delta(y) := \varphi - y/2^n$, we see that we have a geometric series with ratio $e^{2\pi i \delta(y)}$, solve

$$\mathbb{P}[\text{see } y] = \frac{1}{2^{2n}} \left| \frac{1 - e^{2^n \cdot 2\pi i \delta(y)}}{1 - e^{2\pi i \delta(y)}} \right|^2$$

We now aim to bound/estimate this expression, which we do by bounding the numerator/denominator appropriately. Note that:

- (i) $|1 - e^{i\alpha}| = |2 \sin(\alpha/2)| \geq \frac{2}{\pi} |\alpha|$ for $|\alpha| < \pi$
- (ii) $|1 - e^{i\beta}| \leq \beta$, by comparing the length of the chord and the arc between 1 and $e^{i\beta}$ on the unit circle

So for (a), in the probability equation we use (i) with $\alpha = 2^n \cdot 2\pi\delta(a) < 2^n \cdot 2\pi/2^{n+1} < \pi$, and so $|1 - e^{i\alpha}| \geq \frac{2}{\pi}|\alpha| = 2^{n+2}\delta(a)$, where a is the closest n -bit string, so its distance from φ is small.

Then we use (ii) with $\beta = 2\pi\delta(a)$ to upper bound the bottom line, and hence

$$\mathbb{P}[\text{see } a] \geq \frac{1}{2^{2n}} \left(\frac{2^{n+2}\delta(a)}{2\pi\delta(a)} \right) = \frac{4}{\pi^2}$$

For (b), we upper bound the probability equation and sum over all y such that $|\delta(y)| > \varepsilon$.

For the top line, we use that $|1 - e^{i\alpha}| \leq 2$ for any α (trivially), and for the bottom line we use (i), so $|1 - e^{2\pi i\delta(y)}| \geq 4\delta(y)$. Hence

$$\mathbb{P}[y] \leq \frac{1}{2^{2n}} \left(\frac{2}{4\delta(y)} \right)^2 = \frac{1}{2^{2n+2}\delta(y)^2}$$

Now we sum over all y such that $|\delta(y)| > \varepsilon$; the $\delta(y)$ values are separated by $1/2^n$ s, so let δ_+ (resp. δ_-) be the first $\delta(y)$ with $\delta(y) \geq \varepsilon$ (resp. $\delta(y) \leq -\varepsilon$). So $|\delta_+|, |\delta_-| \geq \varepsilon$.

Then if $|\delta(y)| \geq \varepsilon$ we have $|\delta(y)| = \delta_+ + \frac{k}{2^n}$, $k = 0, 1, \dots$ or $|\delta(y)| = \delta_- - \frac{k}{2^n}$, $k = 0, 1, \dots$. So $|\delta(y)| \geq \varepsilon + \frac{k}{2^n}$, $k = 0, 1, \dots$ in each case.

So

$$\begin{aligned} \mathbb{P}[|\delta(y)| \geq \varepsilon] &\leq 2 \sum_{k=0}^{\infty} \frac{1}{2^{2n+2}} \frac{1}{(\varepsilon + \frac{k}{2^n})^2} \\ &\leq \frac{1}{2} \int_0^{\infty} \frac{1}{2^n \varepsilon + k)^2} dk \\ &= \frac{1}{2} \int_{2^n \varepsilon}^{\infty} \frac{dk}{k^2} = \frac{1}{2^{n+1}\varepsilon} \end{aligned}$$

□

This is an extremely useful theorem, and we will see it in various places.

Further Remarks on use of PE:

1. If $C - U^{2^k}$ is implemented as $(C - U)^{2^k}$ then the PE algorithm needs exponential time in n , via $1 + 2 + \dots + 2^{n-1} = 2^n - 1$ $C - U$ gates.

But for some special U s, U^{2^k} and $C - U^{2^k}$ can be implemented in $\text{poly}(k)$ time, so we get the polynomial time PE algorithm. For instance, we can use repeated squaring to calculate exponents efficiently.

This leads to an alternative factoring algorithm (more generally, order-finding or periodicity algorithm) due to Kitaev using PE.

2. If instead of $|v_\varphi\rangle$ we use general d -dimensional input state $|\xi\rangle$, what happens? Since we know what happens on eigenstates, and these form a basis, we have that

$$\begin{aligned} |xi\rangle &= \sum_j c_j |v_{\varphi_j}\rangle \\ U|v_{\varphi_j}\rangle &= e^{2\pi i\varphi_j} |v_{\varphi_j}\rangle \end{aligned}$$

We then get (before the final measurement) a unitary process:

$$|0 \dots 0\rangle |\xi\rangle \xrightarrow{U_{\text{PE}}} \sum_j c_j |\varphi_j\rangle |v_{\varphi_j}\rangle$$

and the Born rule implies the final measurement gives a choice of one of the φ_j s, chosen with probability $|c_j|^2$. In particular, you don't get an arbitrary average of the φ_j s, you can precisely one of them (at random), which is still useful information.

This works perfectly if φ_j s have at most n digits, the number of lines in the PE algorithm. Otherwise with m lines and general φ_j s we get a precision issue (usually swept under the rug); more precisely:

Using Theorem PE(b), if we want m bits correct with probability $1 - \eta$, we use $n = m + o(\log 1/\eta)$ lines, and U_{PE} acts as above with $|\varphi_j\rangle$ registers replaced by

$$(\sqrt{1 - \eta})|\tilde{\varphi}_j\rangle + \sqrt{\eta}|\text{perp}\rangle$$

with $|\text{perp}\rangle \perp |\tilde{\varphi}_j\rangle$, where $|\tilde{\varphi}_j\rangle$ are all n bit strings with the first m bits correct for φ , and $|\text{perp}\rangle$ all n bit strings with the first m not correct. Measurement will then give $\tilde{\varphi}_j$ value with m correct bits with probability $|c_j|^2(1 - \eta)$.

We often ignore this precision issue, since we can make η very, very small with modest $o(\log 1/\eta)$ overhead. The final state is η -close to state giving φ perfectly.

3 Amplitude Amplification

This is the apotheosis of technique in Grover's algorithm.

Some background:

Reflection operation: Recall:

- state $|\alpha\rangle$ in \mathcal{H}_d gives a 1-dim subspace L_α , and $(d - 1)$ -dime orthogonal complement L_α^\perp

$$\begin{aligned} I_{|\alpha\rangle} &:= I - 2|\alpha\rangle\langle\alpha| \\ \therefore I_{|\alpha\rangle}|\alpha\rangle &= -|\alpha\rangle \\ I_{|\alpha\rangle}|\beta\rangle &= |\beta\rangle \text{ for any } |\beta\rangle \perp |\alpha\rangle \end{aligned}$$

i.e. $I_{|\alpha\rangle}$ is reflection in $(d - 1)$ -dim mirror L_α^\perp .

Note for any unitary U , $UI_{|\alpha\rangle}U^\dagger = I_{U|\alpha\rangle}$

- For k -dim subspace $A \subseteq \mathcal{H}_d$ with any orthonormal basis $|a_1\rangle, \dots, |a_k\rangle$, $P_A = \sum_{i=1}^k |a_i\rangle\langle a_i|$ is the projection operator into A (note that this is independent of the choice of orthonormal basis). Then we have

$$\begin{aligned} I_A &:= I - 2P_A \\ \implies I_A|\xi\rangle &= |\xi\rangle \text{ if } |\xi\rangle \in A^\perp \\ I_A|\xi\rangle &= -|\xi\rangle \text{ if } |\xi\rangle \in A \end{aligned}$$

So I_A is reflection in the $(d - k)$ -dim mirror A^\perp .

Recap of Grover's Algorithm

- search for unique 'good' item in an unstructured database of $N = 2^n$ items. This is formalised as:

Given an oracle for $f: n\text{-bits} \rightarrow 1\text{-bit}$.

Promise: there is a unique $x_0 \in n\text{-bits}$ with $f(x_0) = 1$.

Problem: find x_0 .

- closely related to NP & Boolean satisfiability problem
- Using one query to the $(n + 1)$ -qubit U_f , we can implement the reflection operator:

$$I_{|x_0\rangle} : |x_0\rangle \rightarrow \begin{cases} |x\rangle & x \neq x_0 \\ -|x\rangle & x = x_0 \end{cases}$$

Visually, we apply U_f to $|x\rangle \left(\frac{|0\rangle - |1\rangle}{2} \right)$ and discard the last qubit.

- Then consider the Grover iteration operator on n qubits

$$Q := -H_n I_{|0\rangle} H_n I_{|x_0\rangle} = -I_{|\psi_0\rangle} I_{|x_0\rangle}$$

where $H_n = H \otimes \dots \otimes H$, $|\psi_0\rangle = H_n |0 \dots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle$. So one application of Q uses 1 query to U_f .

Theorem: (Grover 1996) *In the 2-dimensional span of $|\psi_0\rangle$ and (unknown) $|x_0\rangle$ the action of Q is rotation by angle 2α where $\sin \alpha = \frac{1}{\sqrt{N}}$.*

Hence to find x_0 given U_f :

1. Make $|\psi_0\rangle$
2. Apply Q m times, where

$$m = \frac{\arccos \frac{1}{\sqrt{N}}}{2 \arcsin \frac{1}{\sqrt{N}}} = \frac{\beta}{2\alpha}$$

with $\cos \beta = \frac{1}{\sqrt{N}}$, β being the angle between $|\psi_0\rangle$ and $|x_0\rangle$ (nearly orthogonal).

This rotates $|\psi_0\rangle$ very close to $|x_0\rangle$ (within $\pm\alpha$)

3. Measure, to see x_0 with high probability $\approx \cos^2 \alpha = 1 - \sin^2 \alpha = 1 - \frac{1}{N}$.

For large N , $\arccos \frac{1}{\sqrt{N}} \approx \frac{\pi}{2}$ and $\arcsin \frac{1}{\sqrt{N}} \approx \frac{1}{\sqrt{N}}$. So $m = \frac{\pi}{4} \sqrt{N}$ iterations/queries to U_f are sufficient (it can be shown this is also necessary).

Classically, we need $O(N)$ queries to see x_0 with any constant probability (independent of N), so we have obtained a square root time speed up quantumly.

Amplitude Amplification Problem

Let G be any subspace (“good subspace”) of state space \mathcal{H} and let G^\perp be its orthogonal complement, such that $\mathcal{H} = G \oplus G^\perp$ (the latter is the “bad subspace”).

Given any $|\psi\rangle \in \mathcal{H}$, we have unique decomposition with real non-negative coefficients:

$$|\psi\rangle = \sin \theta |g\rangle + \cos \theta |b\rangle$$

with $|g\rangle \in G$ and $|b\rangle \in G^\perp$.

We now introduce reflections that flip $|\psi\rangle$ and good vectors:

$$\begin{aligned} I_{|\psi\rangle} &= I - 2|\psi\rangle\langle\psi| \\ I_G &= I - 2P_G \end{aligned}$$

So $\sin \theta = ||P_G |\psi\rangle|| = \text{length of good projection of } |\psi\rangle$.

Then introduce $Q := -I_{|\psi\rangle} I_G$. This is a generalisation of Grover’s Algorithm to any subspace.

Theorem: (Amplitude Amplification) *In the 2-dimensional subspace spanned by $|\psi\rangle$ and $|g\rangle$ (or equivalently by orthonormal vectors $|g\rangle$ and $|b\rangle$), Q is rotation by 2θ where $\sin\theta = \text{length of good projection of } |\psi\rangle$.*

Proof. We have $I_G|g\rangle = -|g\rangle$, $I_G|b\rangle = |b\rangle$ so

$$\begin{aligned} Q|g\rangle &= -I_{|\psi\rangle}I_G|g\rangle = I_{|\psi\rangle}|g\rangle \\ Q|b\rangle &= -I_{|\psi\rangle}|b\rangle \end{aligned}$$

Now

$$\begin{aligned} I_{|\psi\rangle} &= I - 2(\sin\theta|g\rangle + \cos\theta|b\rangle)(\sin\theta\langle g| + \cos\theta\langle b|) \\ &= I - 2[\sin^2\theta|g\rangle\langle g| + \sin\theta\cos\theta|g\rangle\langle b| + \sin\theta\cos\theta|b\rangle\langle g| + \cos^2\theta|b\rangle\langle b|] \end{aligned}$$

And direct calculation (using $\langle g|b\rangle = 0$, $\langle g|g\rangle = \langle b|b\rangle = 1$) gives:

$$\begin{aligned} Q|b\rangle &= \cos 2\theta|b\rangle + \sin 2\theta|g\rangle \\ Q|g\rangle &= -\sin 2\theta|b\rangle + \cos 2\theta|g\rangle \\ &= I|g\rangle - 2\sin^2\theta|g\rangle - 2\sin\theta\cos\theta|b\rangle \\ &= (1 - 2\sin^2\theta)|g\rangle - 2\sin\theta\cos\theta|b\rangle \\ &= \cos 2\theta|g\rangle - \sin 2\theta|b\rangle \end{aligned}$$

So in the $\{|b\rangle, |g\rangle\}$ basis, the matrix of Q is

$$Q = \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix}$$

Which is a rotation through 2θ . □

We can then very simply observe that $Q^n|\psi\rangle = \sin((2n+1)\theta)|g\rangle + \cos((2n+1)\theta)|b\rangle$, and if we measure $Q^n|\psi\rangle$ for good vs bad, then $\mathbb{P}[\text{good}] = \sin^2(2n+1)\theta$. This is maximised when $(2n+1)\theta = \pi/2$, *i.e.* $n = \pi/4\theta - \frac{1}{2}$ (taken to the nearest integer).

Example: If we had $\theta = \pi/6$, then $n = \pi/4\theta - \frac{1}{2} = 1$ (exactly), and Q^1 rotates $|\psi\rangle$ exactly onto $|g\rangle$, hence we obtain a good result with certainty.

Generally, for given θ , n is not an integer so we instead take the nearest integer to $\pi/4\theta - \frac{1}{2} \approx \pi/4\theta$ for small θ , and $Q^n|\psi\rangle$ will be within angle $\pm\theta$ of $|g\rangle$. So $\mathbb{P}[\text{good}] \geq \cos^2\theta \approx 1 - o(\theta^2)$.

All of this can be implemented if $I_{|\psi\rangle}$ and I_G can be implemented (see ES2).

For I_G , it suffices for G to be spanned by computational basis states $|x\rangle$ s and the indicator function ($f(x) = 1$ for x good and $f(x) = 0$ for x bad) is efficiently computable.

For $I_{|\psi\rangle}$: often have $\mathcal{H} = n$ -qubits and $|\psi\rangle = H_n|0\dots 0\rangle$, and then $I_{|W^\psi\rangle}$ can be implemented in $O(n)$ time.

Notes:

1. In the AA process, relative amplitudes of good labels in $|g\rangle$ stay the same as they were in $|\psi\rangle = \sin\theta|g\rangle + \cos\theta|b\rangle$, and AA amplifies the overall $|g\rangle$ amplitude at the expense of the $|b\rangle$ amplitude.

This is useful for state preparations. For instance:

$$\sum_{x \in (\mathbb{Z}/N\mathbb{Z})^\times} |x\rangle$$

2. Final state is generally not exactly $|g\rangle$, but if $\sin \theta$ is known, then we can modify the AA process to make it exact, with only a very modest increase in queries to the oracle - so we can obtain $|g\rangle$ with certainty (ES2).

Applications of AA

1. Grover search with one or more (k) good items in $N = 2^n$

$$\begin{aligned} |\psi\rangle &= |\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{Z}_2^n} |x\rangle \\ &= \sqrt{\frac{k}{N}} \left(\frac{1}{\sqrt{k}} \sum_{\text{good}} |x\rangle \right) + \sqrt{\frac{N-k}{N}} \left(\frac{1}{\sqrt{N-k}} \sum_{\text{bad}} |x\rangle \right) \end{aligned}$$

G spanned by the good $|x\rangle$ s. $\sin \theta = \sqrt{\frac{k}{N}}$ so Q is rotation through 2θ , where $\theta := \arcsin \sqrt{\frac{k}{N}} \approx \sqrt{\frac{k}{N}}$ (since N is often much larger than k - if not then this whole problem is roughly a non-issue, just pick randomly).

So $O(\sqrt{N/k})$ queries needed.

Note that if $N = 4$, $n = 2$, $k = 1$ then $\theta = \pi/6$, $2\theta = \pi/3$, so one query gives the good item of one in four with certainty - despite being very simple, this is rather remarkable as classically this is clearly impossible.

2. Square-root speedup of general quantum algorithms

Let A be a quantum algorithm/circuit (*i.e.* is the unitary operator for the whole circuit), let the input state be $|0 \dots 0\rangle$; so the final state is $A|0 \dots 0\rangle$. The good labels are the desired computational outcomes.

Write $A|0 \dots 0\rangle = \alpha|a\rangle + \beta|b\rangle$, $\alpha = \sin \theta$ where $|a\rangle$ is a normalised, generally unequal supervision of good labels $\sum_{\text{good } x} c_x |x\rangle$.

So $\mathbb{P}[\text{success in 1 run}] = |\alpha|^2$, so we need $o(1/|\alpha|^2)$ repetitions of A to succeed with any given constant (high) probability $1 - \varepsilon$.

However, we can instead use AA!

Note that we assume that we can easily check if the answer is good or bad (this is characteristic of NP, where it is hard to find an answer but easy to verify one). This assumption is so that we can then implement $I_G : |x\rangle \rightarrow -|x\rangle$ if x good, and $|x\rangle$ if x bad (reflection in the good labels).

Consider $|\psi\rangle = A|0 \dots 0\rangle$ and

$$Q = -I_{A|0 \dots 0\rangle} I_G = -A I_{|0 \dots 0\rangle} A^\dagger I_G$$

so all parts are implementable: it is easy to invert A as we have the circuit, I_G is as above, and $I_{|\psi_0\rangle} = I - 2|0 \dots 0\rangle\langle 0 \dots 0|$ (*c.f.* ES2).

So by the Amplitude Amplification theorem Q is rotation through 2θ with $\sin \theta = |\alpha|$. So after $n \approx \pi/4\theta = o(1/\theta) = o(1/\sin \theta) = o(1/|\alpha|)$ (small α) repetitions, $A|0 \dots 0\rangle$ will be rotated very near to $|g\rangle$ and the final measurement will succeed with high probability.

So we get a square root speedup over the direct method.

Moreover, if the success probability of A (*i.e.* $|\alpha|^2$) is known then the exact “improved modified” AA process converts the probabilistic algorithm A into a deterministic one (get the outcome with certainty - this isn’t too important in practice, but is theoretically interesting).

3. Quantum Counting 1998 (uses AA & PE together)

Given $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ with k good x s, we want to estimate k (rather than just find some good x). These counting problems in complexity theory are known to be significantly more difficult than more simple existence problems.

Recall that the Grover operator Q_G for f is rotation through 2θ in 2-dimensional plane of $|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{Z}_2^n} |x\rangle$ and its good projection $|g\rangle = \frac{1}{\sqrt{k}} \sum_{x \text{ good}} |x\rangle$ with $\sin \theta = \sqrt{k/N} \approx \theta$ for $k \ll N$.

Also recall facts about 2D rotation through 2θ in the $\{|b\rangle, |g\rangle\}$ (o.n.b) plane. We have eigenvectors $|e_{\pm}\rangle = \frac{1}{\sqrt{2}}(|b\rangle \pm i|g\rangle)$, with eigenvalues $\lambda_{\pm} = e^{\pm 2i\theta}$ (this is true for any o.n.b replacing $|b\rangle, |g\rangle$).

Then check $|\psi_0\rangle := \sin \theta |g\rangle + \cos \theta |b\rangle = \frac{1}{\sqrt{2}}(e^{i\theta}|e_+\rangle + e^{-i\theta}|e_-\rangle)$ - that is to say this is an equally weighted superposition of $|e_{\pm}\rangle$ s.

For PE application we write eigenvalues $e^{\pm 2i\theta}$ as $e^{2\pi i \varphi_{\pm}}$ with $0 \leq \varphi_{\pm} < 1$. We have $\varphi_+ = (2\theta)/(2\pi) = \theta/\pi$, and $2\pi\varphi_- = -2\theta + 2\pi$ so $\varphi_- = 1 - \theta/\pi$ (note $\theta/\pi = \frac{1}{\pi} \sqrt{k/N}$).

Hence: running the PE algorithm with $U = \text{Grover } Q$ and eigenstate register set to $|\varphi_0\rangle$ will output (an approximation to) θ/π or $1 - \theta/\pi$ with probability $1/2$. For small θ these are easily distinguishable. Hence in either case we get an approximation to $\theta = \sqrt{k/N}$.

More precisely (with n as above), for any m , running PE with m qubit lines gives an m -bit approximation to $\sqrt{k/N}$ and uses $2^m C - Q$ gates, *i.e.* 2^m queries to f . So by theorem PE(a) we learn $\sqrt{k/n}$ to additive error $O(1/2^m)$ with constant probability $4/\pi^2$ using $O(2^m)$ queries to f .

Write $\frac{1}{2^m}$ as $\frac{\delta}{\sqrt{N}}$. So δ is the error in \sqrt{k} directly. So we learn \sqrt{k} to additive error $O(\delta)$ using $O(2^m) = O(\sqrt{N}/\delta)$ queries. Therefore we learn k itself to additive error $O(\delta\sqrt{k})$ using $O(\sqrt{N}/\delta)$ queries.

But classically, the same approximation (obtained with constant probability) this requires $O(N/\delta^2)$ queries, which is quadratically more.

4 Hamiltonian Simulation

The aim here is to use a quantum computer to simulate the evolution/dynamics of a quantum system, given its Hamiltonian H . This is an extremely important and very active area of quantum research, and is expected to be one of the first useful real-world applications of quantum computing. The reason is that studying the behavior of molecules of even relatively modest length becomes extremely difficult classically, but is extremely useful for developing new drugs with the correct intended effects.

For n qubits, we generally need $O(2^n)$ time on a classical computer, but we will do it in $\text{poly}(n)$ time (for a suitable class of H s).

Hamiltonians & Quantum Evolution

We have a physical system: a state $|\psi\rangle$, with a Hamiltonian H . A Hamiltonian is a self-adjoint (Hermitian) operator/matrix, the physical interpretation of which is the “energy observable”. Time

evolution is given by the Schrodinger equation ($\hbar = 1$):

$$\frac{d}{dt}|\psi(t)\rangle = -iH|\psi(t)\rangle$$

so the solution is characterised by H . We remark that the Hamiltonian can in general be time-dependent, but we will only consider the time independent case.

Formally then, $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$ where the exponential given is the matrix exponential defined by the series $e^A = I + A + A^2/2! + \dots$ for any square matrix A . Moreover, if H is Hermitian then e^{iH} is always unitary. Thus our problem is as follows:

Given H and a time t , we want to simulate the action of a unitary $U(t) = e^{-iHt}$ to a suitably good approximation.

We already have a notion for what it is for two vectors to be ‘close’, but we need to formalise a notion of closeness between (unitary) operators; for this we use the operator norm (spectral norm) given by

$$\begin{aligned} \|A\| &= \max_{\|\psi\|=1} \|A|\psi\rangle\| \\ &= |\max \text{ eigenvalue}| \quad \text{if } A \text{ diagonalisable} \end{aligned}$$

It is easy to see that:

- $\|A + B\| \leq \|A\| + \|B\|$
- $\|AB\| \leq \|A\| \cdot \|B\|$

We then say that U approximates \tilde{U} to within ε if $\|U - \tilde{U}\| < \varepsilon$; this has the geometrical interpretation that for any normalised $|\psi\rangle$ the vectors $U|\psi\rangle$ and $\tilde{U}|\psi\rangle$ are at most ε apart.

k -local Hamiltonians

H on n qubits is a $2^n \times 2^n$ Hermitian matrix. We’ll want to simulate $U = e^{-iHt}$ with a circuit of size $\text{poly}(n, t)$ using basic unitary gates (so the operations themselves don’t grow too large, negating the speed up by circuit length) and size $\text{poly}(1/\varepsilon)$ for ε -approximation.

Note that not all H s can be efficiently simulated, but some important physical classes can be.

Definition: (k -local) H is k -local (k fixed constant) on n qubits if

$$H = \sum_{j=1}^m H_j$$

where each H_j is a Hermitian matrix acting on at most k qubits (not necessarily contiguous), *i.e.* each

$$H_j = \tilde{H}_j \otimes I$$

, where \tilde{H}_j acts on some k qubits and I is the identity on the rest.

So this is still an exponentially large matrix, but it is a description of constant size. So we must have $m \leq \binom{n}{k} = O(n^k) = \text{poly}(n)$ terms in H , each with constant size description. So the whole Hamiltonian can be specified with $\text{poly}(n)$ size; generally though this is not the case since there are exponentially many degrees of freedom.

Examples:

- 1) $H = X \otimes I \otimes I - 5Z \otimes I \otimes Y$ is 2-local on 3 qubits.

- 2) Write $M_{(k)}$ to denote the 1-qubit operator M acting on the k^{th} (and I on all the others). The **Ising model** on an $(n \times n)$ square lattice of qubits is

$$H = J \sum_{i,j=1}^{n-1} Z_{(i,j)} Z_{(i,j+1)} + Z_{(i,j)} Z_{(i+1,j)}$$

these pairs act on all nearest neighbours on the square lattice.

The **Heisenberg Model** on a line is

$$H = \sum_{i=1}^{n-1} J_x X_i X_{i+1} + J_y Y_i Y_{i+1} + J_z Z_i Z_{i+1}$$

where J, J_x, J_y, J_z are all constants.

Note: we will want to exponentiate a Hamiltonians, but in general $e^{-i \sum_j H_j t} \neq \prod_j e^{-i H_j t}$ if the H_j s don't commute.

But: $e^{-i H_j t}$ s are local unitary gates, *i.e.* acting on k qubits each, and we'll simulate $U(t_0) = e^{-i \sum_j H_j t_0}$ in terms of $e^{-i H_j t}$ s (even if they don't commute) for suitably small values of t - and we'll have a $\text{poly}(n, t_0)$ -sized circuit too, as required.

If we want to use some standard universal gate set (to further express the gates above) then we invoke:

Theorem: (Solovay-Kitaev Theorem) *Let U be a unitary operator on k (constant) qubits and S any universal set of quantum gates. Then U can be approximated to within ε using $O(\log^c(1/\varepsilon))$ gates from S with $c < 4$ (but exponential in k , if k is varying).*

Proof. Omitted. See *e.g.* Nielsen & Chuang. □

In particular, this means that all S -products of length $O(\log^c(1/\varepsilon))$ will get within ε of any element of $U(k)$.

We'll also need a lemma about accumulation of error (*c.f.* ES3):

Lemma A: *Let $\{U_i\}$ be sets of m unitary operators with $\|U_i - V_i\| \leq \varepsilon$ for all $i = 1, \dots, n$. Then $\|U_m \dots U_1 - V_m \dots V_1\| \leq m\varepsilon$, *i.e.* errors accumulate only linearly*

Proof. Easy induction. □

4.1 Hamiltonian Simulation for local Hamiltonians

Warm-up: the (easy) commuting case.

Lecture 11

Proposition: *Let $H = \sum_{j=1}^m H_j$ be any k -local Hamiltonian with H_j 's commuting. Then for any t , $e^{-i H t}$ can be approximated to within ε by a circuit with $O(m \text{poly}(\log(m/\varepsilon)))$ gates for any given universal set.*

Note: As $m = \binom{n}{k} = O(n^k)$, this is $\text{poly}(n, \log 1/\varepsilon)$ too, and $\log(1/\varepsilon)$ = number of bits of precision in the approximation.

Proof. The H_j s commute, so $e^{-i \sum_j H_j t} = \prod_{j=1}^m (e^{-i H_j t})$, and each $e^{-i H_j t}$ is a k -qubit gate. Then by Solovay-Kitaev (SK), each $e^{-i H_j t}$ can be approximated to within ε/m with $O(\text{poly}(\log m/\varepsilon))$ gates. Then by Lemma A, the full product $\prod_{j=1}^m$ is approximated to within $m(\varepsilon/m) = \varepsilon$, using a total of $O(m \text{poly}(\log m/\varepsilon))$ gates. □

Now for the full non-commuting case:

Notation: For any matrix X , write $X + O(\varepsilon)$ for any $X + E$ with $\|E\| = O(\varepsilon)$.

Lemma B (Lie-Trotter Product Formula): Let A, B be matrices with $\|A\|, \|B\| \leq K$, with $K < 1$ suitably small. Then $e^{-iA}e^{-iB} = e^{-i(A+B)} + O(K^2)$.

Proof. Recall that:

$$\begin{aligned} e^{-iA} &= I - iA + \sum_{k=2}^{\infty} \frac{(-iA)^k}{k!} \\ &= I - iA + (iA)^2 \sum_{k=0}^{\infty} \frac{(-iA)^k}{(k+2)!} \end{aligned}$$

and in fact we have, by the triangle inequality:

$$\begin{aligned} \left\| \sum_{k=0}^{\infty} \frac{(-iA)^k}{(k+2)!} \right\| &\leq \sum_{k=0}^{\infty} \frac{\|(-iA)\|^k}{k!} \\ &= e^{\|A\|} \leq e^K < e \end{aligned}$$

Therefore $e^{-iA} = I - iA + O(K^2)$. (*) So:

$$\begin{aligned} e^{-iA}e^{-iB} &= (I - iA + O(K^2))(I - iB + O(K^2)) \\ &= I - i(A+B) + O(K^2) \\ &= e^{-i(A+B)} + O(K^2) \end{aligned}$$

□

Now we apply this repeatedly to accumulate the sum of our matrices H_1, \dots, H_n in the exponent.

- Note: If each $\|H_i\| < K$ then $\|H_1 + \dots + H_\ell\| < \ell K$, and we'll want this to be < 1 for all $\ell \leq m$. So for now, we'll assume $K < 1/m$ in order to have Lie-Trotter at all stages.
- Also take $t = 1$ for now (will do general t later).

Given these, consider:

$$e^{-iH_1}e^{-iH_2} \dots e^{-iH_m} = \left[e^{-i(H_1+H_2)} + O(K^2) \right] e^{-iH_3} \dots e^{-iH_m}$$

Note now that $\|H_1 + H_2\| < 2K$, and note also that the error generated at each state says the same with further unitary U s; $\|AU\| = \|A\|$. So it can be pulled through the remainder of the product:

$$\begin{aligned} &= e^{-i(H_1+H_2)}e^{-iH_3} \dots e^{-iH_m} + O(K^2) \\ &= \left[e^{-i(H_1+H_2+H_3)} + O((2K)^2) \right] e^{-iH_4} \dots e^{-iH_m} + O(K^2) \\ &\vdots \\ &= e^{-i(H_1+H_2+\dots+H_m)} + O(K^2) + O((2K)^2) + \dots + O([(m-1)K]^2) \\ &= e^{-i(H_1+H_2+\dots+H_m)} + O(m^3K^2) \end{aligned}$$

We write this error as Cm^3K^2 , and refer to it as ERR1.

Now for general finite $\|H_j\|$ s and t values:

$\|H_j t\| < Kt$ can be *large*, so introduce N (which will be large-ish, chosen later) and note $H_j t/N$ has norm $\tilde{K} = \|H_j t/N\| < Kt/N$ (*), which can be suitably *small*. *i.e.* we divide t up into (small) $1/N$ intervals, and $U = e^{i(H_1 + \dots + H_m)t} = [e^{i(H_1 t/N + \dots + H_m t/N)}]^N$.

We want the final error for U to be $< \varepsilon$, so by Lemma A we want error for $[\dots]$ to be $< \varepsilon/N$ *i.e.* from (ERR1) and (*), we want $Cm^3 \tilde{K}^2 < \varepsilon/N$. So $Cm^3 K^2 t^2 / N^2 < \varepsilon/N$, and in particular $N \geq Cm^3 K^2 t^2 / \varepsilon$ (ERR2). If N is this large, then

$$\|e^{-iH_1 t/N} e^{-iH_2 t/N} \dots e^{-iH_m t/N} - e^{-i(H_1 + \dots + H_m)t/N}\| < \varepsilon/N$$

Then by Lemma A again,

$$\|(e^{-iH_1 t/N} \dots e^{-iH_m t/N})^N - e^{-i(H_1 + \dots + H_m)t}\| < \varepsilon$$

The product on the left is Nm gates of the form $e^{-iH_j t/N}$, where N is given by (ERR2), so the circuit size is $O(m^4 (Kt)^2 / \varepsilon)$.

Recall for n qubits and k -local Hamiltonians $m = O(n^k)$, so circuit size overall is $O(n^{4k} (Kt)^2 / \varepsilon) = O(n^{4k}, t^2, 1/\varepsilon)$.

- We have circuit \mathcal{C} of size $|\mathcal{C}| = O(m^4 (Kt)^2 / \varepsilon)$ gates of the form $e^{-iH_j t/N}$ approximating $U = e^{iHt}$ to $O(\varepsilon)$. If we want to use a standard universal set, then lemma A implies each of these $e^{-iH_j t/N}$ gates needs to be approximated to $O(\varepsilon/|\mathcal{C}|)$ to maintain overall $O(\varepsilon)$ error for U . So by SK thm we need $O(\log^c(|\mathcal{C}|/\varepsilon))$ gates from the universal set for each, *i.e.* get an extra (very modest!) multiplicative factor of $O(\log^c(m^4 (Kt)^2 / \varepsilon^2))$ in $|\mathcal{C}|$.
- For fixed n, ε and variable t , the quantum process e^{-iHt} runs for time t , but the quantum circuit simulation above runs for time $O(t^2)$. By refining the Lie-Trotter Lemma, it can be shown that this can be improved to $O(t^{1+\delta})$ for any $\delta > 0$.

Lecture 12

Harrow Hassidim Lloyd (HHL) Quantum Algorithm for Systems of Linear Equations (2009)

We'll want to solve a linear system $A\underline{x} = \underline{b}$, $\underline{b}, \underline{x} \in \mathbb{C}^N$ where dimension N is potentially very large.

- Write $N = 2^n$ (or 2^n least power of 2 greater than N)
- Rather than outputting the full solution \underline{x} itself (would take at least $O(N)$ time), instead we compute (suitably approximations to) the value of properties of the solution \underline{x} , such as quadratic expressions $\underline{x}^\dagger M \underline{x}$, *e.g.* total weight of some subset of components

Very large systems of such equations are important in applications

- data mining/machine learning on data sets of very large size (terabytes/petabytes)
- numerical solutions of PDEs - discretisation techniques (finite element methods) lead to linear systems of size far larger than the original problem description

The best known classical techniques take $\text{poly}(N)$ time to solve such problems - this is true generally even to just compute properties of the solution as described above; in fact no better method is known than first computing the whole solution itself.

Important parameters (both classical and quantum algorithms)

- system size N
- the desired approximation tolerance ε

- the condition κ of the matrix A , defined as the ratio of the largest to smallest eigenvalue size:

$$\kappa = \left| \frac{\lambda_{\max}}{\lambda_{\min}} \right|$$

Note that $\kappa = \infty$ if A is not invertible. The condition number provides a measure of how close A is to being non-invertible. If we rescale A to have $\lambda_{\max} = 1$, then $|\lambda_{\min}| = 1/\kappa$ and numerical computation of A^{-1} becomes less stable with increasing κ , needing more significant digits of computation and correspondingly longer runtime.

Preliminary requirement for the HHL algorithm

Aim: to compute ε -approximation to properties of the solution of N -dimensional system $Ax = b$ in time $\text{poly}(\log N) = \text{poly}(n)$. We'll take the property to be a quadratic expression of the form $\mu = x^\dagger M x$.

Immediate issue: how the defining ingredients A, b are actually given *e.g.* if simply written out as components, then it takes $\text{poly}(N)$ time just to write them before starting!

Thus we need a different presentation, but one that is still available in important applications.

The $\text{poly}(\log N)$ runtime will be achieved using only $O(\log N)$ qubits, and it will never be required in the algorithm to write down all of A, b or x .

List of starting requirements:

For the matrix A :

1. We will require that A is Hermitian.

Remark: If it is not, just double the size of the system (no effect on time complexity), and solve

$$\begin{pmatrix} 0 & A^\dagger \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}$$

which is $A^\dagger y = 0$ and $Ax = b$, so $y = 0$ (since A invertible).

2. We will require that the condition number κ of A is “suitably small”, in particular that κ is bounded by $\text{poly}(n)$ with increasing $N \sim 2^n$. Such matrices/linear systems are called **well-conditioned**. We will also assume for convenience that A (and RHS b) has been rescaled to have $\lambda_{\max} = 1$, so all sizes of eigenvalues lie in $[1/\kappa, 1] = [1/\text{poly}(n), 1]$.
3. We will need to implement Hamiltonian simulation for e^{iAt_0} (with suitable t_0 , see later). Correspondingly, we require that e^{iAt_0} can be implemented in time (circuit size) $\text{poly}(n, t_0)$.

Remark: Any class of (Hermitian, well-conditioned) matrices A having this property will do for HHL. This is the only place that the presentation of A features in HHL.

In the literature, a particular class (that's still broadly applicable) is often described. These are matrices that are **row-sparse** and **row-computable**, defined as follows:

- (i) A is **row-sparse** if each row contains at most $\text{poly}(n)$ non-zero entries (amongst its $N \sim 2^n$ entries). More generally, A is **row s -sparse** if each row now contains at most s non-zero entries. [Note that no s implies $s = \text{poly}(n)$.]
- (ii) An s -sparse matrix A is called **row-computable** if the entries of A can be efficiently computed in the following sense (instead of, say, being given inefficiently as a list of N^2 values):

There is a (classical) $O(s)$ -time computation which given any row index i , $1 \leq i \leq N$ and integer k , $1 \leq k \leq N$ outputs the k^{th} non-zero entry A_{ij} of row i , and its column location j .

I.e. $C(i, k) = (j, A_j)$. If k exceeds the number of non-zero entries, we can set $C(i, k) = (i, 0)$ as a flag that there are none left.

Theorem: (Hamilton Simulation Property) *Let A be row s -sparse and row computable. Then e^{iAt_0} can be implemented up to error ε by a quantum circuit of size $O(\log N \cdot s^2 \cdot t_0)$ (where we have omitted factors of t_0 and $1/\varepsilon$ that scale better than any power $\alpha > 0$, however small).*

Thus for row sparse matrices (i.e. $s = O(\text{polym})$) the circuit size is $O(\text{poly}(\log N) \cdot t_0)$.

Proof. Beyond the scope of the course. c.f. D. Berry, G. Ahokas *et al.* □

Remark: The requirements (i) and (ii) will not feature in the HHL algorithm itself; they are quoted here *only* to guarantee efficient implementability of the Hamiltonian Simulation for e^{iAt_0} .

Remark: The matrices A arising in finite element methods for numerical PDE solutions are row sparse & row computable.

For the RHS vector b :

4. we will require that $b = (b_1, \dots, b_N)$ has $|b_1|^2 + \dots + |b_N|^2 = 1$ and that the $(\log N)$ -qubit state $\sum b_i |i\rangle$ can be *efficiently* produced i.e. in time $\text{poly}(n)$ on a quantum computer.

Remark: If b is not normalised, assume that $|b|^2 = \sum_{i=1}^N |b_i|^2$ can be efficiently computed, & then we use the normalised version \hat{b} of b . Finally at the end we rescale the output value using the already-computed $|b|^2$.

5. Require that M is Hermitian & the corresponding quantum measurement on $n = \log N$ qubits is implementable in $\text{poly}(n)$ time on a quantum computer. If m not Hermitian, $M = K + iL$, $K = (M + M^\dagger)/2$, $L = (M - M^\dagger)/2$, and then just do HHL for K and L separately.

Lecture 13

HHL will produce a state $|\hat{x}\rangle = "x \text{ normalised}"$ to accuracy ε and then estimate $\mu = x^\dagger M x$, M Hermitian, also to accuracy ε (by measurement of M on it, mean value = μ).

The best known classical general purpose algorithm to achieve the above runs in time $O(N \cdot s \sqrt{\kappa} \log(1/\varepsilon))$.

HHL will output state within ε of $|\hat{x}\rangle$ in time $O(\log N \cdot s^2 \kappa^2 \cdot 1/\varepsilon)$ and similar time to get μ to accuracy within ε .

In the regime that $\varepsilon = \frac{1}{\text{poly}(\log N)}$ and all other conditions, the classical time is $O(\text{poly}(N))$, and the HHL quantum time is $O(\text{poly}(\log N))$, *an exponential speed-up*.

The HHL Algorithm: For clarity, we first assume the ingredients PE, HamSim can be executed error-free, and we will *afterwards* give a discussion of errors and runtime.

- Assume $|\lambda_{\max}| = 1$ and condition number κ is known (or an upper bound known), so all sizes of eigenvalues lie within $[1/\kappa, 1]$.

Work in N -dimensional space, $n = \log N$ qubits. Computational basis $\{|i\rangle : i = 0, \dots, N-1\}$.

Let the eigenvectors and corresponding eigenvalues of A be denoted λ_j and $|u_j\rangle$ respectively, $j = 0, 1, \dots, N-1$.

Begin by implementing RHS state $|b\rangle$, and consider it in the eigenbasis

$$|b\rangle = \sum_{i=0}^{N-1} b_i |i\rangle = \sum_{j=0}^{N-1} \beta_j |u_j\rangle$$

and then

$$|x\rangle = A^{-1}|b\rangle = \sum_{j=0}^{N-1} \beta_j \frac{1}{\lambda_j} |u_j\rangle$$

- note here that $|x\rangle$ is not necessarily normalised here.

This transformation is linear, but not unitary so cannot be implemented directly as a quantum operation. To instead achieve it probabilistically, we'll use PE for $U = e^{iA}$ with the latter exponential (& needed powers of C-U in PE) being implemented by HamSim. Then a so-called post-selected process (*cf* Sheet 1 Q6) will achieve our desired non-unitary transformation:

- PE with HamSim (all assumed to work perfectly) give:

$$|b\rangle|0\rangle \xrightarrow{U_{\text{PE}}} \sum \beta_j |u_j\rangle |\lambda_j\rangle$$

which we denote $(*)$. Here we do *not* do final measurement in PE, only the unitary part.

- Next adjoin an extra ancilla *qubit* $|0\rangle$, and apply the following controlled rotation C-rotn on it, controlled by $|\lambda_j\rangle$ register.

C-rot:

$$\begin{aligned} |\lambda_j\rangle|0\rangle &\mapsto |\lambda_j\rangle[\cos\theta_j|0\rangle + \sin\theta_j|1\rangle] \\ &= \sqrt{1 - \frac{c^2}{\lambda_j^2}} |\lambda_j\rangle|0\rangle + \frac{c}{\lambda_j} |\lambda_j\rangle|1\rangle \end{aligned}$$

Here:

- c is chosen to have $c \leq \min_j |\lambda_j|$ (so we take $c = \frac{1}{\kappa}$ for definiteness)
- the angle of rotation $\theta_j \in (-\pi/2, \pi/2)$ given by

$$\theta_j = \arcsin \frac{c}{\lambda_j}$$

determined by content of $|\lambda_j\rangle$ register.

Note: C-rot is a *fixed* operation on $n+1$ qubits that in fact can be implemented by a $\text{poly}(n)$ sized circuit of 1 and 2-qubit gates (see Sheet 3 Q3) (for any λ with $|\lambda| < \varepsilon$).

Apply C-rot to $(*)$ results in the state:

$$|\psi_0\rangle = \sum_{j=0}^{N-1} \beta_j \sqrt{1 - \frac{c^2}{\lambda_j^2}} |u_j\rangle |\lambda_j\rangle |0\rangle + \beta_j \frac{c}{\lambda_j} |u_j\rangle |\lambda_j\rangle |1\rangle$$

from which we *want* the part associated to the ancilla state $|1\rangle$.

- (Post selection step) we measure the ancilla in the computational basis, hoping to get result 1. What is the probability that this happens? We will see result 1 with prob p given by:

$$\begin{aligned} p &= \left\| \sum_j \beta_j \frac{c}{\lambda_j} |u_j\rangle |\lambda_j\rangle \right\|^2 = \sum_j \left| \beta_j \frac{c}{\lambda_j} \right|^2 \\ &= \frac{1}{\kappa^2} \sum_j \left| \frac{\beta_j}{\lambda_j} \right|^2 \geq \frac{1}{\kappa^2} \end{aligned}$$

where the last inequality comes from the fact that $\sum |\beta_j^2| = 1$ and $\frac{1}{\lambda_j} \geq 1$ since $|\lambda_j| \in [1/\kappa, 1]$.

In that case, the post-measurement state will be:

$$|\Upsilon\rangle = \frac{1}{\sqrt{p}} c \sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j} |u_j\rangle |\lambda_j\rangle$$

To mitigate the probability p , and obtain $|\Upsilon\rangle$ with any fixed (high) level of probability $1 - \eta$ ($\eta > 0$ small) we can repeat the whole process $\log(1/\eta)/p = O(\kappa^2)$ times, and outcome 1 will occur at least once with probability greater than $1 - \eta$. [This follows from the basic probability lemma: If a single trial has success probability p and we repeat the trial M times independently, then for any $0 < 1 - \eta < 1$, the probability of at least one successful trial in M exceeds $1 - \eta$ if $M \geq -\log(\eta)/p$.]

Remark (Amplitude Amplification): The $O(\kappa^2)$ repetitions above can be improved to $O(\kappa)$ by use of AA, instead of repeated measurements.

- Having obtained the state:

$$|\Upsilon\rangle = \frac{1}{\sqrt{p}} c \sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j} |u_j\rangle |\lambda_j\rangle$$

we run PE in reverse to “uncompute” or “erase” the $|\lambda_j\rangle$ register, resetting it to $|0\rangle$ (and discarding it) and obtain:

$$|\hat{x}\rangle = \frac{c}{\sqrt{p}} \sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j} |u_j\rangle = \frac{1}{\sqrt{p}} |x\rangle$$

i.e. normalised version of $|x\rangle$.

- Finally, we perform a measurement of observable M on (copies of) $|\hat{x}\rangle$ [the copies are obtained by just running the whole process multiple times] to estimate its mean value,

$$\frac{c^2}{p} \langle x | M | x \rangle = \frac{c^2}{p} \mu$$

via empirical mean of its values.

According to the Chernoff-Hoeffding bound (see notes), $O(\log(1/\eta)/\xi^2)$ measurements of M will suffice to estimate the mean $\mu c^2/p$ to any desired accuracy ξ , with any desired success probability $1 - \eta$.

We know c . For p , we can similarly estimate the probability p (in the post selection step) to any accuracy ξ by applying the Chernoff-Hoeffding bound to the ancilla measurement outcome, 0 or 1, its mean value being $0p_0 + 1p_1 = p$, our p that we want. Using these estimated values of $c^2/p \langle x | M | x \rangle$ and p (and $c = \frac{1}{\kappa}$) we obtain our value of $\mu = \langle x | M | x \rangle$.

Remarks about run time & approximation errors: [Non-examinable; messy to do rigorously. For a full analysis: original HHL paper ‘Phys. Rev. Lett. vol 103, 150502 (2009)’.]

- we assume we can make the RHS vector state $|b\rangle$ exactly
- similarly the controlled rotation C-rot is a fixed unitary & can be implemented efficiently - we assume it is exact too (ES3 Q3)

- Consider now the phase estimation required. The PE algorithm for a unitary U & n qubit lines gives estimate of λ up to n bits of precision. We denote it by λ' and assume for now that it is the closest n -bit approximation to λ . *I.e.* we get λ up to additive error $\eta = 1/2^n$. The PE process uses Hamiltonian Simulation that requires execution of controlled $U, U^2, \dots, U^{2^{n-1}}$. If $U = e^{iA}$, then $C - U = e^{i\tilde{A}}$, where

$$\tilde{A} = \left[\begin{array}{ccc|c} 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 \\ \hline & & 0 & A \end{array} \right]$$

is double the dimension of A , and is s -sparse/row-computable if A was. note that $C - U = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & U \end{array} \right]$. Thus we need to implement $e^{i\tilde{A}t}$ for $t = 1, 2, 4, \dots, 2^{n-1}$, which is done using HamSim for s -sparse matrices with circuit sizes $O(\log N \cdot t \cdot s^2)$, so total circuit size is $O(\log N \cdot t_0 \cdot s^2)$, with $t_0 = 1 + 2 + \dots + 2^{n-1} \approx 2^n = 1/\eta$.

After the controlled rotation C-rot using the λ 's produced and measurement in post-selection step, we denote the final state produced by:

$$|\hat{x}'\rangle = \frac{1}{D'} \sum \beta_j \frac{c}{\lambda_j'} |u_j\rangle$$

where $D' = \sqrt{p'} = \sqrt{\sum_j |\beta_j|^2 \frac{c^2}{\lambda_j'^2}}$

and we write $|\hat{x}\rangle, X, p$ for the corresponding expressions with (unexact) λ 's replaced by the true value λ s.

Our requirement is that $|\hat{x}'\rangle$ be within ε of the exact state $|\hat{x}\rangle$ and need to choose a correspondingly suitably large $t_0 = \frac{1}{\eta}$ *i.e.* suitably good accuracy of λ 's vs. λ s.

To establish dependence of t_0 (or accuracy $\eta = 1/t_0$ in λ) on a desired final error tolerance ε , we use two facts:

- If λ has additive error $\delta(\lambda) = \eta$ then $\frac{1}{\lambda}$ has $\delta(1/\lambda) \sim \frac{1}{\lambda^2} \delta(\lambda)$ so $1/\lambda$ has relative error $\delta(1/\lambda)/(1/\lambda) = \eta/\lambda$.
- If A', B' approximate A, B to relative error ξ then A'/B' approximates A/B to relative error ξ too:

$$\frac{A'}{B'} = \frac{A(1 + O(\xi))}{B(1 + O(\xi))} = \frac{A}{B}(1 + O(\xi))$$

Now by (a), $1/\lambda_j'$ approximates $1/\lambda_j$ to relative error $\delta(1/\lambda)/(1/\lambda) = O(\eta/\lambda_j) = O(\kappa\eta)$ as $1/\lambda_j \leq \kappa$.

Also similarly D' approximates D to relative error $O(\kappa\eta)$ too as D is homogeneous of degree 1 in $1/\lambda_j$ s. Then by (b) with $A' = \frac{\beta_j c}{\lambda_j'}$ and $B' = D'$ we see that amplitudes of $|\hat{x}'\rangle$ approximate those of $|\hat{x}\rangle$ up to multiplicative error $O(\kappa\eta)$ so:

$$\begin{aligned} ||\hat{x}'\rangle - |\hat{x}\rangle|| &= ||(1 + O(\kappa\eta))|\hat{x}\rangle - |\hat{x}\rangle|| \\ &= O(\kappa\eta) \text{ (want } \varepsilon) \end{aligned}$$

Thus for given ε we choose $\kappa\eta = \varepsilon$ *i.e.* $\eta = \varepsilon/\kappa$, $t_0 = 1/\eta$ so we get $t_0 = \kappa/\varepsilon$, and the PE step (run once) has run time $O(\log N t t_0 s^2) = O(\log N (\kappa/\varepsilon) s^2)$.

In HHL algorithm, to obtain \hat{x} with high probability we use amplitude amplification which requires repeating the PE process $O(\kappa)$ times.

So the final total run time is $O(\log N \cdot \kappa^2 s^2 \varepsilon^{-1})$ *i.e.* $O(\text{poly}(\log N), \varepsilon^{-1})$ for $s, \kappa = O(\text{poly}(\log N))$.

Further comments on error & run time:

- In above, have assumed the PE process (with n qubit lines) outputs the closest n -bit approximation λ' to λ , whereas in fact it outputs a superposition of all $\lambda_k = \lambda' \pm k/2^n$ (peaked around $k = 0$), which introduces a further source of error that has so far been completely ignored.

To mitigate this, we use Thm. PE(b) *i.e.* use more qubit lines in PE, to get the best n -bit λ' with high probability.

Analysis is omitted; see original paper for details. But can show that this does not increase the above complexity/runtime limit of the HHL algorithm.

In HHL paper, an “improved PE algorithm” is used, with a non-uniform initial superposition on qubit lines, giving a better concentration around the best λ' above.

- In the HHL algorithm, the PE step is the source of the worst accuracy (ε)-dependence being $O(1/\varepsilon)$. This was subsequently improved (in 2013) in D. Berry, A. Childs, R. Cleve, ..., R. Somma arxiv: 1312.1414, improving $O(1/\varepsilon)$ exponentially to $O(\log(1/\varepsilon))$. The technique involves replacing the PE process by a different method altogether, the so-called method of “linear combination of unitaries”.

See ES3 Q5 for basic idea involved.

It is in fact quite a general method, and can be used to improve the ε -dependence of HamSim too, again in a dramatic, exponential way.

Clifford Computations & Classical Simulation of Quantum Computing

Question: What is the “key quantum effect or resource” giving quantum computing its benefit over classical computing?

It's worth noting that it is currently in fact unproven whether QC is in fact faster than CC; *c.f.* P vs NP. But as things stand it is faster.

To formalise the comparison of classical vs quantum computing we introduce the precise mathematical notion of **classical simulation of quantum computation**:

- description of a quantum circuit C *i.e.* list of 1 and 2-qubit gates on n qubits, of size $N = \text{poly}(n)$
- description of input, say $|i_1 \dots i_n\rangle$ or $|\alpha_1\rangle \dots |\alpha_n\rangle$ - general *product* state
- nominated output line(s) [say just one (1st line) for yes/no answer]

Note that the full description is of size $\text{poly}(n)$ classically.

Want: by classical means only:

Weak simulation: sample the output distribution (using classical computer, that can make classical random choices too - *e.g.* “toss a fair coin” will suffice)

Strong simulation: *calculate* output probabilities p_0 and p_1 [see ES3 for multiline case]

Moreover, we want to do it *efficiently i.e.* in *classical* $\text{poly}(n)$ time *i.e.* “efficient classical simulation”.

In that case: we would have no quantum computational benefits over classical (up to some polynomial-time overhead, which we don't really care about).

Note: the quantum process itself gives only a weak simulation (“self-simulation”), *i.e.* sample of output distribution. So a strong simulation is asking for much more.

A “direct” strong classical simulation is always possible, but not generally efficient: the action of successive gates is just linear algebra, and clearly can calculate all evolving amplitudes *etc.*, but: linear algebra develops in 2^n -dimensional space, so it takes exponential time to do matrix multiplications *etc.*

However: *if* it is promised that the state is a product state $|\alpha_1\rangle \dots |\alpha_n\rangle$ at *every* stage, then we get an *efficient* strong simulation, since each step involves one or two qubits in a known state *eg.* $|\alpha_i\rangle, |\alpha_j\rangle$ separate from all others.

Example: 2 qubit gate U on qubits 1 & 2 of

$$|\psi\rangle = \sum c_{i_1 \dots i_n} |i_1 \dots i_n\rangle$$

with amplitudes of $U|\psi\rangle \sim \tilde{c}_{i_1 \dots i_n}$ given by

$$\tilde{c}_{i_1 \dots i_n} = \underbrace{\sum_{k_1, k_2=0}^1 U_{i_1 i_2}^{k_1 k_2} c_{k_1 k_2}}_{4 \times 4 \text{ product}} \underbrace{i_3 \dots i_n}_{\text{exp. choices}}$$

But if $|\psi\rangle$ a product, then $c_{i_1 \dots i_n} = a_{i_1} b_{i_2} \dots d_{i_n}$ factorises and

$$\tilde{c}_{i_1 \dots i_n} = \left(\sum_0^1 U_{i_1 i_2}^{k_1 k_2} a_{k_1} b_{k_2} \right) c_{i_3} \dots d_{i_n}$$

, so just update the 2 qubits and factorise the answer back into $\tilde{a}_{i_1} \tilde{b}_{i_2}$ (by product state promise).

Note also that “product state” = “absence of entanglement”, and it is sometimes thus claimed that entanglement is the key resource for quantum computational benefits over classical. BUT: the above shows only that no entanglement \implies no qcomp benefit, but it says nothing of the converse. So entanglement is necessary, but *not* sufficient and there is more to do here.

Notably, consider:

Clifford Computation

Preliminary Definitions:

The *n-qubit Pauli group* $\mathcal{P}_n = \{\pm 1, \pm i p_1 \otimes \dots \otimes p_n : \text{each } p_i \text{ is } I, X, Y, Z\}$ is a finite subgroup of $U(2^n)$.

A **Clifford operation** C on n qubits is any unitary that conjugates Paulis to Paulis. *i.e.* for all $P \in \mathcal{P}_n$ $CPC^\dagger = \tilde{P}$ is in \mathcal{P}_n too. The collection of all such operations forms the **Clifford group**, which is the normaliser group of the subgroup \mathcal{P}_n in $U(2^n)$.

These two groups are important in many applications:

- 1) theory of quantum error correction (“stabiliser codes”) & fault-tolerant quantum computation
- 2) insights into quantum computing power
- 3) have very elegant mathematical properties

Examples:

1. all Paulis are Cliffords (clearly)
2. H is Clifford
3. $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \pi/2$ phase gate is Clifford
4. CX (2 qubit gate) is Clifford
5. also have $Z = S^2, X = HZH$ and so $\text{SWAP}_{12} = CX_{12}CX_{21}CX_{12}, CZ_{12} = H_x(CX)_{12}H_2$ are Cliffords

In fact there is a full explicit characterisation of Cliffords (proof omitted):

Theorem: C on n qubits is Clifford if and only if C is a circuit of H, S and CX gates.

Clifford computations: circuits that involve only Clifford gates. They have very special classical simulation properties (*cf.* below).

Note: we can generate much entanglement using Clifford operations. *eg.*

$$|0\rangle_1|0\rangle_2 \xrightarrow{H_1} (|0\rangle + |1\rangle)|0\rangle \xrightarrow{CX_{12}} |0\rangle|0\rangle + |1\rangle|1\rangle$$

and similarly we can make

$$\frac{1}{\sqrt{2}}(|0\dots 0\rangle + |1\dots 1\rangle)$$

the “cat state”.

Theorem: (Gottesman-Knill Theorem (variant)) Let C be any Clifford circuit on n qubits, size $N = \text{poly}(n)$ (say given as a circuit of H, S and CX gates) with input any product state $|a_1\rangle \dots |a_n\rangle$ and output a measurement on the first qubit line (or any other line).

Then the output can always be classically strongly efficiently simulated (and so weakly too).

There are many different proofs of this using a variety of techniques - we will give one here.

Remark: An alternative proof uses the so-called *stabiliser formalism* (introduced for quantum error correction theory) and this was the origin of the GK Theorem. See *e.g.* Nielsen & Chuang Ch. 10.5.

Lecture 16

Proof. Write $Z_1 = Z \otimes I \otimes \dots \otimes I$. Recall $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$.

So $Z_1 = \Pi_0 - \Pi_1$ with $\Pi_{0,1}$ being n -qubit projections to 1st qubit being 0, 1 respectively.

So $p_0 - p_1 = \langle \psi_{\text{final}} | z_1 | \psi_{\text{final}} \rangle = \langle \psi_{\text{in}} | C^\dagger Z_1 C | \psi_{\text{in}} \rangle$ since $|\psi_{\text{final}}\rangle = C|\psi_{\text{in}}\rangle$. Now if $C = C_N \dots C_2 C_1$ (each C_i is H, S, CX) we get $p_0 - p_1 = \langle \psi_{\text{in}} | C_1^\dagger \dots C_N^\dagger Z_1 C_N \dots C_1 | \psi_{\text{in}} \rangle$. By the defining property of Cliffords, we finally get that

$$p_0 - p_1 = \langle \psi_{\text{in}} | \tilde{P}_1 \otimes \dots \otimes \tilde{P}_n | \psi_{\text{in}} \rangle \quad (*)$$

where each \tilde{P} is a Pauli.

All update rules for Pauli products by H, S, CX conjugations are easy to compute (classically) and record once & for all. [Indeed, H, S, CX are only 1, 2-qubit gates, so only constant size calculation is required.]

Finally we use $|\psi_{\text{in}}\rangle = |a_1\rangle \dots |a_n\rangle$ is *product* state, so

$$p_0 - p_1 = \prod_{i=1}^n \langle a_i | \tilde{P}_i | a_i \rangle$$

[(*) is a product operator wedged between product states.] So $p_0 - p_1$ is a product of $n \times 2 \times 2$ matrix calculations, so this is a poly-time (in fact, linear time) calculation and is efficiently computable.

Then since $p_0 + p_1 = 1$, we get p_0 and p_1 with efficient classical computation. \square

Now let's extend (unitary) Clifford circuits to allow inclusion of intermediate measurement (1-qubit) and carry on using post measurement state. We distinguish two scenarios:

- **Adaptive:** Choice of later gates *can* depend on the outcome of earlier measurements
- **Non-adaptive:** Choice of later gates *cannot* depend on the outcome of earlier measurements.

Theorem: *Let C be any Clifford circuit with intermediate measurements, arbitrary product state inputs, and single line outputs.*

- (a) *if C is non-adaptive then the output is classically efficiently strongly simulatable*
- (b) *if C is adaptive then full universal quantum computation is possible*

Remark: Can show as well that if the input is only *computational basis* states, then (b) remains classically efficiently simulatable but only weakly (and strongly if $P = NP$, but we won't worry about that).

For this and more variants: R. Jozsa & M. van der Nest arxiv: 1305.6190.

Ideas. For (a), just note that in fact non-adaptive Clifford circuits are easily reducible to the fully unitary case (& then use GK Thm). In fact (ES3) the following can be seen to be equivalent:

Scenario A: C_1 , then intermediate measurement, then C_2 , where C_2 is independent of measurement outcome.

Scenario B: C_1 then C_2 , but introduce extra line with input $|0\rangle$ ancilla, and instead of measurement do a Clifford CX with the ancilla and then discard the ancilla.

The reason for the equivalence is that the CX separates the $|0\rangle$ and $|1\rangle$ that would be measured in Scenario A into $|0\rangle|0\rangle$ and $|1\rangle|1\rangle$ orthogonal states, with the same probabilistic weightings. Then they are completely separated and it is as though they have been measured.

Note this only requires us to add in a polynomial amount of stuff, so the computation remains efficient.

For (b): Use Bravyi-Kitaev idea of so-called ***magic states***, together with the following preliminary fact:

Fact: Clifford gates with the 1-qubit (non-Clifford) T gate, $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = \sqrt{S} = \text{Phase}(\pi/4)$ gate, is universal for quantum computing.

The magic states are going to be used to implement a T -gate, but without actually using a T -gate.

let $|A\rangle$ denote the 1-qubit state

$$|A\rangle = \frac{1}{\sqrt{2}} (|0\rangle + e^{i\pi/4}|1\rangle)$$

a so-called magic state. We will implement a T gate (on any line k) using a 1-qubit ancilla $|A\rangle$ and an adaptive Clifford circuit, known as the ***T-gadget***:

We have $|\psi\rangle$ as our input state across n lines, and we focus on line k and add ancilla $|A\rangle$. Perform a controlled not on line k with $|A\rangle$, and measure the ancilla to obtain result $m = 0$ or 1 . Then on line k apply S^m , which is either I or S depending on the outcome m , and then discard the ancilla.

The claim here is that we then always get $T_k|\psi\rangle$, a T gate acting on $|\psi\rangle$ on line k , as the outcome of this process. Note also that the probabilities for $m = 0, 1$ are always $1/2, 1/2$. See ES3 for further details.

So we can make T -gates, and hence generate full universal quantum computing by the Fact. \square

Finale.

We thus have: for Clifford circuits with product state inputs, single line outputs and intermediate measurements:

- (a) non-adaptive - classically (strongly) efficiently simulatable
- (b) adaptive - has full universal quantum computing power

i.e. for Cliffords C_k , $M(i, y) \equiv$ “measurement of line i , get result $y = 0$ or 1 ” we have:

- (a) $C_0 M(i, y_1) C_1 M(i_2, y_2) C_2 \dots$
- (b) $C_0 M(i, y_1) C_1(y_1) M(i_2, y_2) C_2(y_1, y_2) \dots$

Therefore a purely classical ingredient, *adaptive choices of gate* (note: measurements are done in both scenarios) is a resource that elevates classically limited power to full quantum power.

Furthermore, experimentally there is “no difference” between (a) & (b), in the sense that there are no new quantum processes in (b) that do not already occur in (a); a run of an adaptive circuit could have occurred as a run of a non-adaptive circuit with the same outcomes.