An Introduction to Quantum Computers and Hidden Subgroup Problems

Postquantum Cryptography Reading Group

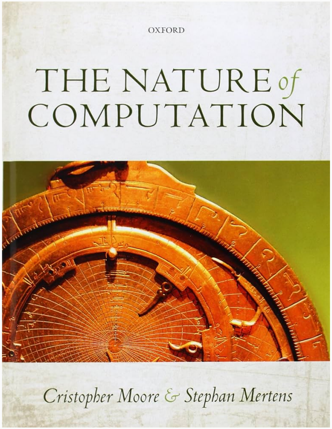
Luke Mader

The plan

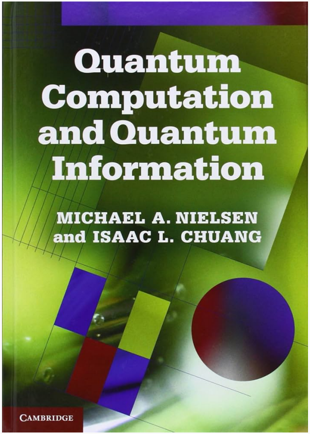
1. An overview of quantum for quantum computing
2. Some quantum algorithms:
    - The Deutsch problem and phase kickback
    - Quantum Fourier transforms and the Deutsch-Jozsa problem
    - Simon's problem
    - Shor's algorithm
3. Hidden subgroup problems on finite Abelian groups

Some good resources

This presentation is heavily taken from:

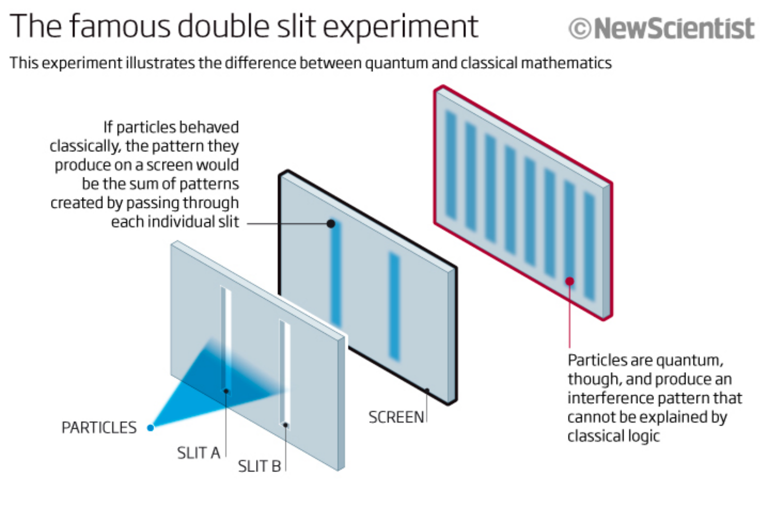


(a) Chapter 15



(b) Section I and II

The Double Slit Experiment



Bits and qubits

In classical computing, data is represented through *bits*.

A bit has two states: 0 or 1, true or false, on or off, etc. Physically, this is voltage on or off in a circuit.

Bits and qubits

In classical computing, data is represented through *bits*.

A bit has two states: 0 or 1, true or false, on or off, etc. Physically, this is voltage on or off in a circuit.

The quantum analogue is a *qubit*, and could physically be the spin of an electron (up or down) or the polarization of a photon (horizontal or vertical).

A qubit has a *continuum of states*.

Bits and qubits

A classical computer with m bits has 2^m states.

We can view each *state as a basis vector for a 2^m-dimensional vector space*, and *each computation as a $2^m \times 2^m$ matrix* acting on the state.

A program is the composition of all the matrices describing the computations.

Bits and qubits

Suppose we have two bits x_1 and x_2. Our *computational basis states* are

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

The operation $x_2 \mapsto x_1$ (e.g. $|10\rangle \mapsto |11\rangle$) is described by

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Bits and qubits

If now our operation is

$$\begin{cases} x_2 \mapsto x_1 & \text{with probability } \frac{1}{2} \\ x_2 \mapsto x_2 \text{ (do nothing)} & \text{with probability } \frac{1}{2} \end{cases}$$

we have a corresponding *stochastic matrix*

$$\begin{pmatrix} 1 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 1 \end{pmatrix}$$

Vectors in the state space are now probability distributions, e.g.

$$U|10\rangle = \frac{1}{2}(|10\rangle + |11\rangle)$$

and the computer has an equal chance of being in either $|10\rangle$ or $|11\rangle$.

Bits and qubits

A qubit has states in $\mathbb{C}^2$, such as the *computational basis states*

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Quantum particles can also be in complex linear combinations

(*superpositions*) of states:

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

α and β are known as the *amplitudes*. As they are complex, they can

interfere *constructively and destructively*.

Bits and qubits

If we measure a qubit, we either measure 0 or 1; *nothing else!*

If the qubit has the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then

$$\mathbb{P}(\text{Measuring } 0) = |\alpha|^2$$

$$\mathbb{P}(\text{Measuring } 1) = |\beta|^2$$

$$\text{Total Probability} = |\alpha|^2 + |\beta|^2 = 1 = \|\psi\|^2$$

The state of a qubit is unobservable; when measuring, we only get information about the state.

Bits and qubits

We describe operations on our qubits through matrices acting on our state.

We need our matrices to preserve total probability. Computations are therefore described by *unitary matrices*, as these preserve the inner product.

Unitary matrices are invertible, which is interpreted as *reversible processes* and they *cannot create or destroy information*.

Some example of unitary operators

Pauli matrices:

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

σ_x acts as a classical NOT-gate, e.g $\sigma_x|0\rangle = |1\rangle$, $\sigma_x|1\rangle = |0\rangle$.

σ_y and σ_z are more quantum as they introduce phase changes:

$$\sigma_y|0\rangle = i|1\rangle,\ \sigma_y|1\rangle = -i|0\rangle$$

$$\sigma_z|0\rangle = |0\rangle,\ \sigma_z|1\rangle = -|1\rangle$$

Some example of unitary operators

Hadamard matrix:

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

Maps $|0\rangle$ and $|1\rangle$ to superpositions of the two where each state is equally likely (*uniform superposition*);

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) =: |+\rangle$$
$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =: |-\rangle$$

$|\pm\rangle$ are the eigenvectors of σ_x and are called the X-basis.

Multiple qubits

If we have N qubits, their states live in $(\mathbb{C}^2)^{\otimes N}$.

Our N-qubit basis vectors are the tensor products of single qubit each basis vector; e.g.

$$|10\rangle := \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |1\rangle \otimes |0\rangle =: |1, 0\rangle$$

As

$$\begin{pmatrix} u_0 \\ u_1 \end{pmatrix} \otimes \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} = \begin{pmatrix} u_0 v_0 \\ u_0 v_1 \\ u_1 v_0 \\ u_1 v_1 \end{pmatrix}$$

the amplitudes of the joint state $|u, v\rangle$ are the products of the amplitudes

Quantum computers can be faster than classical computers due to

- *Parallelism* – e.g. qubits being in a superposition of states
- *Interference* – amplitudes are complex, so can combine constructively and destructively

Deutsch's algorithm

Let $f\colon \{0,1\} \to \{0,1\}$. Does $f(0) = f(1)$?

Classically, can determine through two *queries*: calculating $f(0)$ and $f(1)$ separately.

Deutsch's algorithm

Consider a two qubit computer in state $|x, y\rangle$. Define the unitary map

$$U_f |x, y\rangle = |x, y \oplus f(x)\rangle$$

where $\oplus$ is addition mod 2.

If x is in the uniform superposition $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and y in the state $|0\rangle$, then

$$U_f |+, |0\rangle\rangle = \frac{1}{\sqrt{2}} (|0, f(0)\rangle + |1, f(1)\rangle)$$

We have found information about $f(0)$ and $f(1)$ with only one computation; this is *parallelism*.

Deutsch's algorithm

We can improve this by using *interference* to be better than classical computation. Notice that

$$U_f|x, y\rangle = |x\rangle \otimes \sigma_x^{f(x)}|y\rangle$$

$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ is an *eigenvector for σ_x* with eigenvalue -1, so

$$U_f|x, -\rangle = (-1)^{f(x)}|x, -\rangle.$$

Now preparing x in the state $|+\rangle$ gives

$$\begin{aligned} U_f|+, -\rangle &= \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \otimes |-\rangle \\ &\equiv \frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}} \otimes |-\rangle. \end{aligned}$$

If we now measure x as $|+\rangle$ then $f(0) = f(1)$.

This is known as *phase kickback*: we prepare our 'output qubit' y to be an eigenvector whose eigenvalue affects the phase of the 'input qubit'.

By then measuring the 'input' qubit x and not caring about the 'output' qubit y, we learn about f.

Deutsch's algorithm

Now consider $f\colon \{0,1\}^n \to \{0,1\}$. We use a $(n+1)$-qubit computer in the state $|\vec{x}, y\rangle$ where $\vec{x} \in \{0,1\}^n$.

Consider again

$$U_f|\vec{x}, y\rangle = |\vec{x}, y \oplus f(\vec{x})\rangle = |\vec{x}\rangle \otimes \sigma_x^{f(x)}|y\rangle.$$

Let's try phase kickback again. Prepare the input qubits $\vec{x}$ in a uniform superposition

$$\frac{1}{\sqrt{2^n}}\sum_{\vec{x}}|\vec{x}\rangle$$

and the output qubit y in $|-\rangle$ to get

$$U_f\frac{1}{\sqrt{2^n}}\sum_{\vec{x}}|\vec{x}, -\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{\vec{x}}(-1)^{f(\vec{x})}|\vec{x}\rangle\right) \otimes |-\rangle.$$

Deutsch's algorithm

The state

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\vec{x}} (-1)^{f(\vec{x})} |\vec{x}\rangle$$

again contains the values of $f(\vec{x})$ in the phases of $\vec{x}$'s amplitudes.

Questions:

1. What basis should we measure $|\psi\rangle$ in?
2. What can we learn when measuring $|\psi\rangle$ about f?

The Quantum Fourier Transform

Discrete Fourier transform: For $x_0, \cdots, x_{N-1} \in \mathbb{C}$,

$$x_j \mapsto y_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}} x_k$$

Quantum Fourier transform: Defined by mapping the computational basis

states

$$|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}} |k\rangle$$

where $0 \leq j \leq N-1$.

QFT on $\mathbb{Z}_2^n$

Going back to $\mathbb{Z}_2^n$, let $|\phi\rangle = \sum_{\vec{x}} a_x |x\rangle$. We want to describe a_x as a linear combination of basis vectors oscillating at specific frequencies.

Each frequency $\vec{k} \in \mathbb{Z}_2^n$ has the basis function $(-1)^{\vec{k}\cdot\vec{x}}$ and basis state

$$|\vec{k}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\vec{x}} (-1)^{\vec{k}\cdot\vec{x}} |x\rangle$$

giving us that

$$|\phi\rangle = \sum_{\vec{k}} \tilde{a}_{\vec{k}} |k\rangle, \qquad \tilde{a}_{\vec{k}} = \frac{1}{\sqrt{2^n}} \sum_{|x\rangle} (-1)^{\vec{k}\cdot\vec{x}} a_{\vec{x}}$$

This is *equivalent to applying $H^{\otimes n}$*.

For our state

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\vec{x}} (-1)^{f(\vec{x})} |\vec{x}\rangle$$

We apply the QFT $H^{\otimes n}$ to $|\psi, 1\rangle$ to get the frequency bits $k_1 k_2 \cdots k_n$ written in the computational basis.

Measuring a given $\vec{k}$ has the probability

$$\mathbb{P}(\vec{k}) = \left|\tilde{a}_{\vec{k}}\right|^2 = \left| \frac{1}{2^n} \sum_{\vec{x}} (-1)^{\vec{k}\cdot\vec{x}+f(\vec{x})} \right|^2$$

If $n = 1$, we have the Deutsch problem. We have two frequencies $k = 0$ and $k = 1$.

If $f(0) = f(1)$ then $\mathbb{P}(0) = 1$, $\mathbb{P}(1) = 0$.

If $f(0) \neq f(1)$ then $\mathbb{P}(0) = 0$, $\mathbb{P}(1) = 1$.

The Deutsch-Jozsa Problem

For $f : \{0,1\}^n \to \{0,1\}$ suppose either is true:

1. f is constant
2. f is balanced: $f(\vec{x}) = 0$ for half of all $\vec{x}$, $f(\vec{x}) = 1$ for the other half.

Which is true?

The probability for observing the frequency of $(0, 0, \cdots, 0)$ is now encoded in

$$\tilde{a}_{\vec{0}} = \frac{1}{2^n} \sum_{\vec{x}} (-1)^{f(\vec{x})}$$

If f is constant, this is ± 1. If f is balanced, this is 0, so

$$\mathbb{P}(\vec{0}) = \left|\tilde{a}_{\vec{0}}\right|^2 = \begin{cases} 0 & \text{if } f \text{ balanced,} \\ 1 & \text{if } f \text{ constant} \end{cases}$$

so we answer the Deutsch-Jozsa problem with just one observation.

The plan for next time

- Simon's problem
- Shor's algorithm
- Generalising our problems: the hidden subspace problem

An Introduction to Quantum Computers and Hidden Subgroup Problems

Postquantum Cryptography Reading Group

Luke Mader

What we saw last time

Quantum particles behave very differently to the world we are used to.

They are *wave-like* – constructive and destructive interference.

They are inherently *probabilistic* – repeated measurements of identically prepared system may give different observations.

What we saw last time

A qubit is our quantum version of a bit.

Physically, could be spin up/down of an electron, horizontal/vertical polarisation of light, etc.

What we saw last time

Mathematically, a qubit lives in $\mathbb{C}^2$ with Euclidean norm, and states (vectors with norm 1) we can observe (in the *computational basis*) are

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Qubits could be in *superpositions* of these states, e.g. $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

We *can't observe* $|\psi\rangle$; instead,

$$\mathbb{P}(\text{observing } 0) = |\alpha|^2 \qquad \mathbb{P}(\text{observing } 1) = |\beta|^2$$

$$\mathbb{P}(\text{observing}) = 1 = \||\psi\rangle\|^2$$

N qubits live in $(\mathbb{C}^2)^{\otimes N}$.

What we saw last time

Abusing quantum properties can give us probabilistic methods to work out problems with less computations.

A method that we saw before for slightly constructed problems:

1. Set up input and output qubits to be in convenient states
2. Apply a unitary operator to cleverly separate information about a given function into the input qubits
3. Change our basis into one which is more convenient to work in
4. Take measurements to get information of our function to solve a question

What we saw last time

The *quantum Fourier transform* takes us to the computational basis to one where state amplitudes are in frequencies:

$$|j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}} |k\rangle$$

This proved useful to us previously as *measuring certain frequencies could give us probabilistic methods* to give us *information about a given function* based on what we can measure

Simon's problem

Simon's Problem

Let $f\colon \{0,1\}^n \to \{0,1\}^n$ such that

$$f(x) = f(x') \iff x' = x \oplus r$$

for some fixed r. How many computations on f to determine r?

1. Define $U_f|x, y\rangle = |x, y \oplus f(x)\rangle$.
2. Prepare the input qubits x in a uniform superposition, prepare the output qubits y in $|0\rangle$.
3. $U_f \frac{1}{\sqrt{2^n}} \sum_x |x, 0\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x, f(x)\rangle$.
4. Observe $f(x)$ to be some f_0. Our state collapses to the x with $f(x) = f_0$, $|\psi, f_0\rangle$ where for some x_0

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus r\rangle)$$

5. Measure ψ in Fourier basis. We observe a given frequency k with chance

$$\mathbb{P}(k) = \begin{cases} \frac{1}{2^{n-1}} & \text{if } k \text{ and } r \text{ are orthogonal} \\ 0 & \text{otherwise} \end{cases}$$

6. The vectors perpendicular to r form a $(n-1)$-dimensional subspace, so observe enough k's that span this subspace. This happens by computing f roughly n times.

Shor's factoring algorithm

In the first talk, we saw that RSA is a cryptosystem we have faith in because to break it, you need to be able to factor large numbers.

Factoring for classical computers is difficult and takes a long time. For quantum computers, it is a lot easier.

Suppose we want to find a non-trivial factor of N.

1. If N even, return 2.
2. Use classical algorithm to determine if $N = a^b$ for $a \geq 1$, $b \geq 2$. If so, return a.
3. Guess $x \in \{3, \cdots, N-1\}$ to be a factor. If $\gcd(x, N) > 1$, return $\gcd(x, N)$.
4. Use quantum algorithm to find the smallest r such that $x^r \equiv 1 \mod N$.
5. If r even and $x^{\frac{r}{2}} \not\equiv -1 \mod N$, check if $\gcd(x^{\frac{r}{2}} - 1, N)$ or $\gcd(x^{\frac{r}{2}} + 1, N)$ is a factor.
6. If none of the above was successful, the algorithm fails.

The quantum order finding algorithm

With 'enough' input qubits t and output qubits:

1. Prepare input qubits in a uniform superposition $\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j, 1\rangle$
2. Apply unitary operator $U|j, k\rangle = |j, x^j \mod N\rangle$ for our guess of a factor x
3. $|1\rangle$ can be written as a sum of eigenstates u_s of U with Fourier coefficients, so applying inverse Fourier transform gives us the state
$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\frac{s}{r}, u_s\rangle.$$
4. Take measurement to get $\frac{s}{r}$, apply the *continued fractions algorithm* to get r.

The factoring problem falls into the class of *bounded-error quantum polynomial time* (BQP) problems.

These problems can be solved in *polynomial-time* by quantum algorithms which *give the correct answer at least $\frac{2}{3}$ of the time*.

Order finding is a special case of the period finding problem, and we can express it as the following:

Order finding as periodicity

Fix a such that $a^r \equiv 1 \mod N$ and $f : \mathbb{Z} \to \left\{a^j : j \in \mathbb{Z}_r\right\}$ with

$$f(x) = a^x, \qquad f(x+r) = f(x)$$

How do we find r?

This is the same sort of problem as we've seen previously.

Diffie-Hellmann

As we saw previously, Diffie-Hellmann key exchange is a way to share symmetric public keys securely in public channels, and can be attacked by quantum computers via the *discrete logarithm problem*:

Discrete Logarithm Problem

Fix $a, N \in \mathbb{Z}$ and let r be smallest number with $a^r \equiv 1 \mod N$. Define $f : \mathbb{Z}_r \times \mathbb{Z}_r \to \{a^j : j \in \mathbb{Z}_r\}$,

$$f(x_1, x_2) = a^{sx_1 + x_2}$$

which has the period $(l, -sl)$ for some choice of l. What is s?

The Hidden Subgroup Problem

Hidden Subspace Problem

Let G be a group, $f\colon G \to S$ a function such that there is a subgroup

$H \subseteq G$ with

$$f(x) = f(x') \iff x' = xh \text{ for some } h \in H.$$

What is H?

Solving HSP for Finite Abelian Groups

Quantum computers can easily solve the hidden subgroup problem for finite Abelian groups:

- Prepare qubits in uniform superposition $\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g, f(g)\rangle$.
- Convert $|f(g)\rangle$ into Fourier basis:

$$|f(g)\rangle = \frac{1}{\sqrt{|G|}} \sum_{j=0}^{|G|-1} \exp(\frac{2\pi ijg}{|G|}) |\hat{f}(j)\rangle$$

which, as f is constant and different on each coset of G, has nearly zero amplitude for all values of j except those satisfying

$$\sum_{h \in H} \exp(\frac{-2\pi ijh}{|G|}) = |H|$$

- As G is finite Abelian, there exists primes $p_1, \cdots, p_N$ such that $G \cong Z_{p_1} \times \cdots \times Z_{p_N}$, so for $g_k \in Z_{p_k}$ we get

$$\exp(\frac{2\pi ijg}{|G|}) = \prod_{j=k}^{N} \exp(\frac{2\pi ij'_k g_k}{p_k})$$

- An algorithm known as *phase estimation* gets us j'_k, which lets us find the j for which the amplitudes of $|\hat{f}(j)\rangle$ are not nearly 0, which lets us find H.

An interesting problem to look into

Things aren't as simple in the non-Abelian case – this would take multiple talks by itself!

A nice natural non-Abelian problem which can be formed as a Hidden Subgroup Problem is the *graph isomorphism problem*:

Graph Isomorphism Problem

If G_1, G_2 are two graphs, is there a permutation π on the edges with $\pi(G_1) = G_2$? I.e. are the graphs topologically equivalent?

- "Quantum algorithms for algebraic problems" by A. Childs and W. van Dam
- "The Nature of Computing" by C. Moore and S. Mertens, Ch 15.6