Advanced Computational Complexity

George Voutsadakis[1]

[1]Mathematics and Computer Science
Lake Superior State University

LSSU Math 600

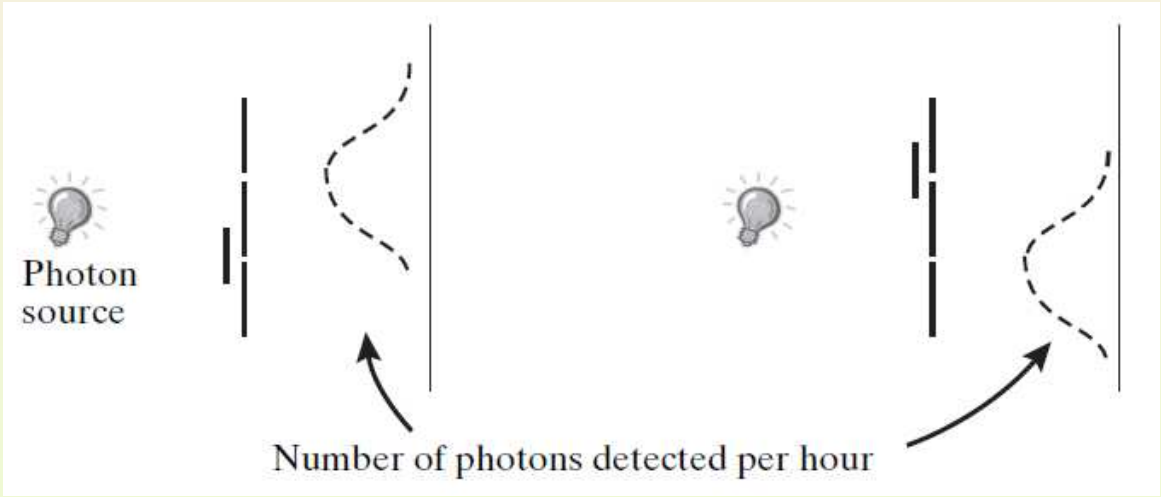
1. Quantum Computation
   - Quantum Weirdness: The Two-Slit Experiment
   - Quantum Superposition and Qubits
   - Definition of Quantum Computation and BQP
   - Grover's Search Algorithm
   - Simon's Algorithm
   - Shor's Algorithm: Integer Factorization
   - BQP and Classical Complexity Classes

Subsection 1

Quantum Weirdness: The Two-Slit Experiment

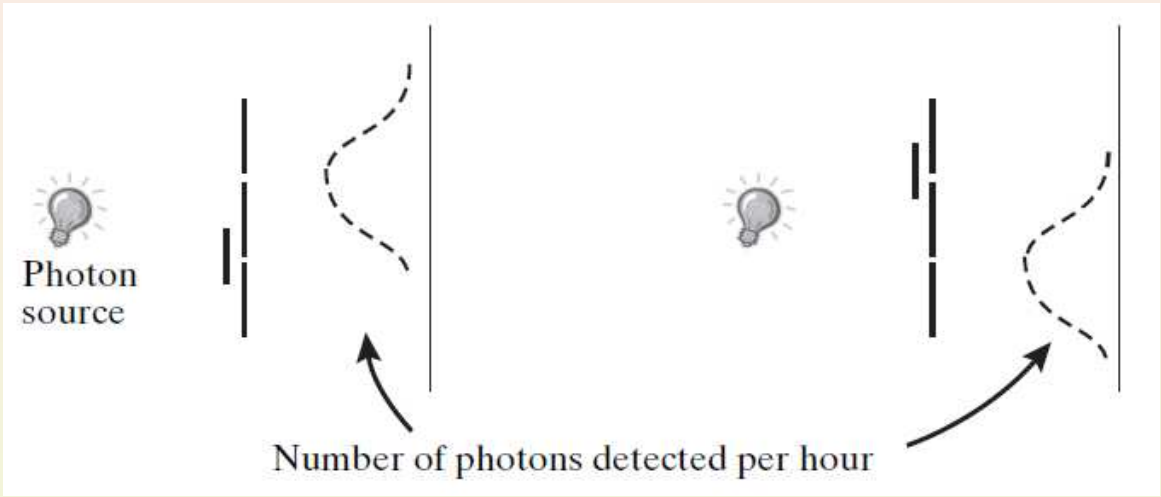
The Two-Slit Experiment

- Now we describe the two-slit experiment, that illustrates the fact that basic physical properties of an elementary particle are "smeared".
- Suppose that a source that fires photons one by one (say, at the rate of one photon per second) is placed in front of a wall containing two tiny slits.



The Two-Slit Experiment (Cont'd)

- Beyond the wall, we place an array of detectors that light up whenever a photon hits them.



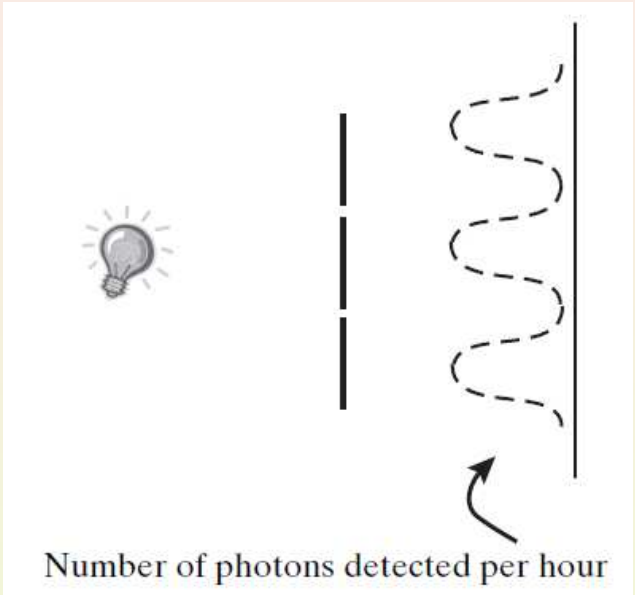
- We measure the number of times each detector lights up in an hour.
- When we cover one of the slits, we expect that the detectors directly behind the open slit will receive the largest number of hits.
- This is indeed the case.

The Two-Slit Experiment (Cont'd)

- When both slits are uncovered, we expect that the number of times each detector is hit is the sum of:
  - The number of times it is hit when the first slit is open;
  - The number of times it is hit when the second slit is open.
- In particular, uncovering both slits should only increase the number of times each location is hit.
- Surprisingly, this is not what happens.

The Two-Slit Experiment (Cont'd)

- The pattern of hits exhibits the "interference" phenomena.



- In particular, at several detectors the total hit rate is lower when both slits are open as compared to when a single slit is open.
- This defies explanation if photons behave as particles or "little balls".

Quantum Mechanics Explanation

- According to quantum mechanics, it is wrong to think of a photon as a little ball that can either go through the first slit or the second.
- Rather, somehow the photon instantaneously explores all possible paths to the detectors through all open slits.
  - Some paths are taken with positive "amplitude" and some with negative "amplitude";
  - Two paths arriving at a detector with opposite signs will cancel each other.
- The end result is that the distribution of hit rates depends upon the number of open slits, since the photon "finds out" how many slits are open via this exploration of all possible paths.

Verification Experiment

- To check whether the path exploration asserted by quantum mechanics is actually happening, we place a detector at each slit.
- If a photon is really going through both slits simultaneously, we hope to detect it at both slits.
- However, when we try to make the photon reveal its quantum nature this way, the quantum nature i.e., interference pattern, disappears!
- The hit rates observed exactly correspond to the little balls model.

Consequence for Quantum Computation

- The explanation is that observing an object "collapses" its distribution of possibilities and so changes the result of the experiment.
- One moral to draw from this is that quantum computers, if they are ever built, will have to be carefully isolated from external influences and noise, since noise may be viewed as a "measurement" performed by the environment on the system.
- Of course, we can never completely isolate the system.
- This means we have to make quantum computation tolerant of a little noise.
- This seems to be possible under some noise models.

Subsection 2

Quantum Superposition and Qubits

Qubits and Superpositions

- Now we describe quantum superposition.
- We use a very simple quantum system called a qubit.
- Classical computation involves manipulation of bits or, more generally, storage elements with finite memory.
- The analogous unit of storage in quantum computing is a **qubit**.
- It can be in two basic states, which we denote by zero and one.
- Unlike a classical bit, it can be simultaneously in both basic states.
- Thus, the state of a qubit at any time is called a **superposition** of these basic states.

Qubits

- We denote the basic states by

$$|0\rangle \quad \text{and} \quad |1\rangle .$$

- We allow a qubit to be in any state of the form

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

where α_0, α_1 are complex numbers satisfying

$$|\alpha_0|^2 + |\alpha_1|^2 = 1.$$

- The numbers α_0, α_1 are called **amplitudes**.

Observing Qubits

- A qubit can be in any state of the form $\alpha_0 |0\rangle + \alpha_1 |1\rangle$, where α_0, α_1 are complex numbers satisfying $|\alpha_0|^2 + |\alpha_1|^2 = 1$.
- If isolated from outside influences, the qubit stays in this superposition, until it is observed by an observer.
- When the qubit is observed:
  - It is revealed to be in state zero, i.e., $|0\rangle$, with probability $|\alpha_0|^2$;
  - It is revealed to be in state one, i.e., $|1\rangle$, with probability $|\alpha_1|^2$.
- After observation the amplitude wave collapses, and the values of the amplitudes are irretrievably lost.
- We restrict attention to the case where the amplitudes are real (possibly negative) numbers.

Pair of Qubits

- A system of two qubits can exist in four basic states

$$|00\rangle\,,\ |01\rangle\,,\ |10\rangle\,,\ |11\rangle\,.$$

- The state of a two-qubit system at any time is described by a superposition of the type

$$\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle\,,$$

where

$$\sum_{b_1,b_2} |\alpha_{b_1 b_2}|^2 = 1.$$

- When this system is observed, its state is revealed to be $|b_1 b_2\rangle$ with probability $|\alpha_{b_1 b_2}|^2$.
- We will sometimes denote the state $|xy\rangle$ as $|x\rangle\,|y\rangle$.

The Geometry of Quantum States

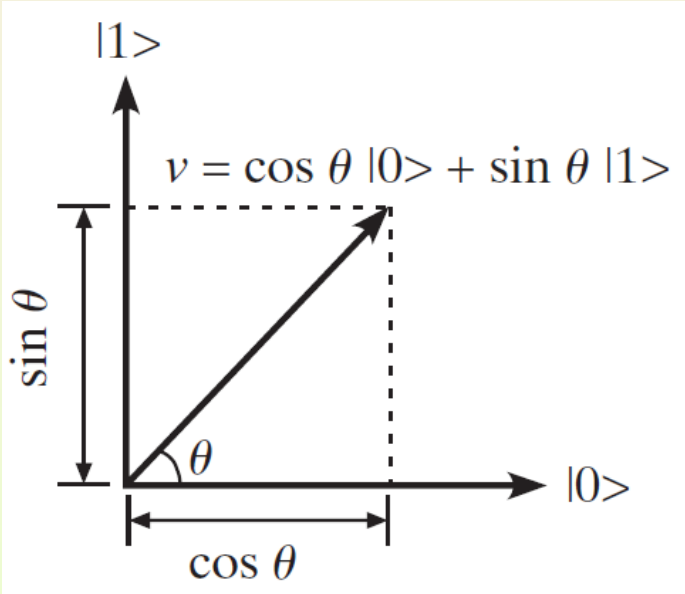
- It is often helpful to think of quantum states geometrically as vectors.
- Consider the case of a single qubit system (with real amplitudes).
- The two basic states can be visualized as two orthogonal unit vectors $|0\rangle$ and $|1\rangle$ in $\mathbb{R}^2$, e.g.,

$$|0\rangle = (1, 0) \quad \text{and} \quad |1\rangle = (0, 1).$$

- We denoted the state of the system by

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

- It can be interpreted as a vector that is the sum of α_0 times the first vector and α_1 times the second.



The Geometry of Quantum States (Cont'd)

- Now α_0, α_1 are real numbers satisfying

$$\alpha_0^2 + \alpha_1^2 = 1.$$

- So there is a unique angle $\theta \in [0, 2\pi)$, such that

$$\alpha_0 = \cos\theta \quad \text{and} \quad \alpha_1 = \sin\theta.$$

- Thus we can think of the system state as

$$\cos\theta|0\rangle + \sin\theta|1\rangle.$$

- That is, it is a unit vector that makes an angle θ with the $|0\rangle$ vector and an angle $\frac{\pi}{2} - \theta$ with the $|1\rangle$ vector.
- When measured, the system's state is revealed to be:
  - $|0\rangle$ with probability $\cos^2\theta$;
  - $|1\rangle$ with probability $\sin^2\theta$.

Example

- The following are two legitimate state vectors for a one-qubit quantum system,

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \quad \text{and} \quad \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

- In both cases, if the qubit is measured, it will contain either 0 or 1 with probability $\frac{1}{2}$.
- Nevertheless, these are considered distinct states.
- We will see that it is possible to differentiate between them using quantum operations.

Additional Notation

- States are always unit vectors.
- So we often drop the normalization factor in the notation.
- E.g., we use

$$|0\rangle - |1\rangle$$

to denote the state $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$.

- We call the state where all coefficients are equal the **uniform state**.

Example: The uniform state for a two-qubit system is

$$|00\rangle + |01\rangle + |10\rangle + |11\rangle.$$

Here we dropped the normalization factor of $\frac{1}{2}$.

Operations on Qubits

- We used the notation $|x\rangle|y\rangle$ for $|xy\rangle$.
- Consider this as an operation.
- As can be checked, it satisfies the distributive law.
- So we can write the uniform state of a two-qubit system as

$$(|0\rangle + |1\rangle)(|0\rangle + |1\rangle).$$

- This expression shows that this state consists of two one-qubit systems in uniform state.
- We manipulate states of a qubit using quantum operations.
- They are functions that maps the current state to the new state.
- We shall only use operations on single qubits.

Operations on Qubits (Cont'd)

- Quantum mechanics allows only unitary operations.
- These are linear operations that preserve the invariant

$$|\alpha_0|^2 + |\alpha_1|^2 = 1.$$

- Unitary operations on a single qubit, with real coefficients, involve one of the following:
  - A reflection of the state vector about a fixed vector in $\mathbb{R}^2$;
  - A rotation of the state vector by some angle $\theta \in [0, 2\pi)$.

The Parity Game

- Two players Alice and Bob are isolated from each other.
- An experiment asks them to participate in the following guessing game.
  1. The experimenter chooses two random bits $x, y \in_R \{0, 1\}$.
  2. He presents x to Alice and y to Bob.
  3. Alice and Bob respond with bits a, b, respectively.
  4. Alice and Bob win if and only if

$$a \oplus b = x \wedge y,$$

where $\oplus$ denotes the XOR operation (addition modulo 2).

Ensuring Non-Communication

- The players' isolation from each other can be ensured using the special theory of relativity.
- The players are separated by a light year, each accompanied by an assistant of the experimenter.
- At a designated time:
  - The assistants toss their independent random coins to create x and y;
  - They present them to Alice and Bob respectively.
  - They receive Alice's and Bob' answers;
  - They transmit everything to the experimenter at a central location.
- Alice and Bob do not have time to exchange information between receiving x, y and before giving their answers.

Maximum Winning Probability

- Alice and Bob can ensure that they win with probability at least $\frac{3}{4}$.
- This can be achieved by, e.g., always sending $a = b = 0$.
- We show that this is the best they can do.
- This seems intuitive, since they are forbidden from coordinating their responses.

Strategies

- A **strategy** for the players is a pair of functions

$$f, g : \{0,1\} \to \{0,1\},$$

  such that the players' answers a, b are computed only as functions of the information they see.
- In this case, we have

$$a = f(x) \quad \text{and} \quad b = g(y).$$

- A **probabilistic strategy** is a distribution on strategies.

Theorem

In the previous scenario, no (deterministic or probabilistic) strategy used by Alice and Bob can cause them to win with probability more than $\frac{3}{4}$.

Proof of the Theorem

- Assume that there is a (possibly probabilistic) strategy that causes them to win with probability more than $\frac{3}{4}$.

  By a standard averaging argument, there is a fixed choice of the players' randomness that succeeds with at least the same probability.

  Hence, we may assume without loss of generality that the players' strategy is deterministic.

  The function $f : \{0,1\} \to \{0,1\}$ that Alice uses can be one of only four possible functions.
  - The constant zero;
  - The constant one;
  - $f(x) = x$;
  - $f(x) = 1 - x$.

  We analyze the case $f(x) = x$ (the other cases are similar).

Proof of the Theorem (Cont'd)

- Alice's response a is merely x.

  So the players win iff

  $$b = (x \wedge y) \oplus x.$$

  On input y, Bob needs to find b that makes them win.
  - Suppose, first, $y = 1$.
    Then $x \wedge y = x$.
    Choosing $b = 0$ ensures Alice and Bob win with probability 1.
  - Suppose, next, $y = 0$.
    Then $(x \wedge y) \oplus x = x$.
    But Bob does not know x.
    So the probability that his output b is equal to x is at most $\frac{1}{2}$.

  Thus, the total probability of a win is at most $\frac{3}{4}$.

The Parity Game with Sharing of Quantum Information

- Suppose, now, Alice and Bob can share a two-qubit system, created in a certain state and split between them before they were taken a light year apart.
- We show that, in that case, they can win the parity game with probability better than $\frac{3}{4}$.
- This can be achieved by using the following strategy.
  1. Before the game begins, Alice and Bob prepare a two-qubit system in the state $|00\rangle + |11\rangle$, which we will call the **EPR** (Einstein, Podosky, Rosen) **state**.
  2. Alice and Bob split the qubits:
     - Alice takes the first qubit;
     - Bob takes the second qubit.

     Quantum mechanics does not require the individual bits of a two-qubit quantum system to be physically close to one another.
     It is important that Alice and Bob have not measured these qubits yet.

The Parity Game with Sharing (cont'd)

3. Alice receives the qubit x from the experimenter.
   - If $x = 1$, then she applies a rotation operation by $\frac{\pi}{8}$ to her qubit. Since the operation involves only her qubit, she can perform it even with no input from Bob.
4. Bob receives the qubit y from the experimenter.
   - If $y = 1$, he applies a rotation operation by $-\frac{\pi}{8}$ to his qubit.
5. Both Alice and Bob measure their respective qubits. They output the values obtained as their answers a and b.

- The order in which Alice and Bob perform their rotations and measurements does not matter.

  It can be shown that all orders yield exactly the same distribution.
- Splitting a two-qubit system and applying unitary transformations to the different parts may sound strange.

  However, this experiment has been performed several times in practice, verifying the prediction in the following theorem.

Increasing the Probability of Winning

Theorem

With the above strategy, Alice and Bob win with probability at least 0.8.

- Recall that Alice and Bob win the game if they output a different answer, when $x = y = 1$, and the same answer, otherwise.
- The intuition behind the proof is that:
  - If it is not the case that $x = y = 1$, the states of the two qubits will be "close" to one another (with the angle between them being at most $\frac{\pi}{8} = 22.5$);
  - If $x = y = 1$, the states will be "far" (having angle $\frac{\pi}{4}$ or 45).
- Specifically we show that, with a Alice's output and by b Bob's:
  1. If $x = y = 0$, then $a = b$ with probability 1;
  2. If $x \neq y$, then $a = b$ with probability $\cos^2 \frac{\pi}{8} \geq 0.85$;
  3. If $x = y = 1$, then $a = b$ with probability $\frac{1}{2}$.

  So the overall acceptance probability is $\geq \frac{1}{4} \cdot 1 + \frac{1}{2} \cdot 0.85 + \frac{1}{4} \cdot \frac{1}{2} = 0.8$.

Analysis of Cases 1 and 2

- Case 1: Both Alice and Bob perform no operation on their qubits.
  So when measured it will be either in the state $|00\rangle$ or $|11\rangle$.
  Both result in Alice and Bob's outputs being equal.
- Case 2: It suffices to consider the case that $x = 0, y = 1$.
  In this case:
  - Alice applies no transformation to her qubit;
  - Bob rotates his qubit in a $-\frac{\pi}{8}$ angle.

  Suppose that:
  - Alice first measures her qubit;
  - Then Bob makes his rotation and measurements.

  Then the following occur:
  - With probability $\frac{1}{2}$, Alice will get the value 0;
  - Bob's qubit will collapse to the state $|0\rangle$;
  - Then, it will be rotated by a $-\frac{\pi}{8}$ angle;
  - Thus, measuring, Bob obtains the value 0 with probability $\cos^2 \frac{\pi}{8}$.

  Similarly, if $x = 1$, then Bob outputs 1 with probability $\cos^2 \frac{\pi}{8}$.

Analysis of Case 3

- Case 3: We use direct computation.

  Suppose both rotations are performed.

  Then the two-qubit system has the state

$$\begin{aligned}
&(\cos\tfrac{\pi}{8}|0\rangle + \sin\tfrac{\pi}{8}|1\rangle)(\cos\tfrac{\pi}{8}|0\rangle - \sin\tfrac{\pi}{8}|1\rangle) \\
&\qquad + (-\sin\tfrac{\pi}{8}|0\rangle + \cos\tfrac{\pi}{8}|1\rangle)(\sin\tfrac{\pi}{8}|0\rangle + \cos\tfrac{\pi}{8}|1\rangle) \\
&= (\cos^2\tfrac{\pi}{8} - \sin^2\tfrac{\pi}{8})|00\rangle - 2\sin\tfrac{\pi}{8}\cos\tfrac{\pi}{8}|01\rangle \\
&\qquad + 2\sin\tfrac{\pi}{8}\cos\tfrac{\pi}{8}|10\rangle + (\cos^2\tfrac{\pi}{8} - \sin^2\tfrac{\pi}{8})|11\rangle.
\end{aligned}$$

  Now we have

$$\cos^2\frac{\pi}{8} - \sin^2\frac{\pi}{8} = \cos\frac{\pi}{4} = \frac{1}{\sqrt{2}} = \sin\frac{\pi}{4} = 2\sin\frac{\pi}{8}\cos\frac{\pi}{8}.$$

  So all coefficients in this state have the same absolute value.

  Hence, when measured, the two-qubit system will yield either one of the four values 00, 01, 10 and 11 with equal probability $\frac{1}{4}$.

Subsection 3

Definition of Quantum Computation and BQP

Some Linear Algebra

- Let $z = a + ib$ be a complex number, where $i = \sqrt{-1}$.
- The **complex conjugate** of z is

$$\overline{z} = a - ib.$$

- Note that

$$z\overline{z} = a^2 + b^2 = |z|^2.$$

- The inner product of two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{C}^m$, is

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{x \in [m]} \boldsymbol{u}_x \overline{\boldsymbol{v}}_x.$$

- The **norm** of a vector $\boldsymbol{u}$ is

$$\|\boldsymbol{u}\|_2 = \sqrt{\langle \boldsymbol{u}, \boldsymbol{u} \rangle} = \sqrt{\sum_{x \in [m]} |\boldsymbol{u}_x|^2}.$$

Some Linear Algebra (Cont'd)

- Two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are **orthogonal** if

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0.$$

- A set $\{\boldsymbol{v}^i\}_{i \in [m]}$ of vectors in $\mathbb{C}^M$ is an **orthonormal basis** of $\mathbb{C}^m$ if, for every $i, j \in [m]$,

$$\langle \boldsymbol{v}^i, \boldsymbol{v}^j \rangle = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

- If A is an $m \times m$ matrix, then A^* denotes the **conjugate transpose** of A:

$$A^*_{x,y} = \overline{A}_{y,x}, \quad \text{for every } x, y \in [m].$$

- An $m \times m$ matrix A is **unitary** if

$$AA^* = I,$$

where I is the $m \times m$ identity matrix.

Real and Unitary Matrices

- If z is a real number, then
$$\overline{z} = z.$$
- If all vectors and matrices involved are real, then:
  - The inner product is equal to the standard inner product of $\mathbb{R}^n$;
  - The conjugate transpose is equal to the standard transpose.
- For real vectors $\boldsymbol{u}, \boldsymbol{v}$,
$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \|\boldsymbol{u}\|_2 \|\boldsymbol{v}\|_2 \cos\theta,$$
where θ is the angle between the $\boldsymbol{u}$ and $\boldsymbol{v}$.

Unitary Matrices

Claim (Unitary Matrices)

For every $m \times m$ complex matrix A, the following conditions are equivalent:

1. A is unitary, i.e., $AA^* = I$;
2. For every vector $\boldsymbol{v} \in \mathbb{C}^m$,

$$\|A\boldsymbol{v}\|_2 = \|\boldsymbol{v}\|_2;$$

3. For every orthonormal basis $\{\boldsymbol{v}^i\}_{i\in[m]}$ of $\mathbb{C}^m$, the set $\{A\boldsymbol{v}^i\}_{i\in[m]}$ is an orthonormal basis of $\mathbb{C}^m$;
4. The columns of A form an orthonormal basis of $\mathbb{C}^m$;
5. The rows of A form an orthonormal basis of $\mathbb{C}^m$.

Quantum Registers and State Vectors

- In a standard digital computer, we implement a bit of memory by a physical object that has two states:
  - The ON or 1 state;
  - The OFF or 0 state.
- By taking m such objects together we have an m-bit register whose state can be described by a string in $\{0,1\}^m$.
- A **quantum register** is composed of m qubits.
- Its state is a superposition of all 2^m basic states, i.e., it is a vector

$$\boldsymbol{v} = \langle \boldsymbol{v}_{0^m}, \boldsymbol{v}_{0^{m-1}1}, \ldots, \boldsymbol{v}_{1^m} \rangle \in \mathbb{C}^{2^m}, \quad \sum_x |\boldsymbol{v}_x|^2 = 1.$$

- According to quantum mechanics, when measuring the register:
  - We obtain the value x with probability $|\boldsymbol{v}_x|^2$;
  - The state of the register collapses to the vector $|x\rangle$, i.e., the coefficients corresponding to the basic states $|y\rangle$, for $y \neq x$ will become 0.

Quantum Operations

Definition (Quantum Operation)

A **quantum operation** for an m-qubit register is a function

$$F : \mathbb{C}^{2^m} \to \mathbb{C}$$

that maps its previous state to the new state and satisfies:

- **Linearity**: F is a linear function, i.e., for every $\boldsymbol{v} \in \mathbb{C}^{2^m}$,

$$F(\boldsymbol{v}) = \sum_{x \in \{0,1\}^m} \boldsymbol{v}_x F(|x\rangle);$$

- **Norm Preservation**: F maps unit vectors to unit vectors, i.e., for every $\boldsymbol{v}$, with $\|\boldsymbol{v}\|_2 = 1$,

$$\|F(\boldsymbol{v})\|_2 = 1.$$

Comments

- Norm preservation is quite natural.

  After all, only unit vectors can describe states.
- Linearity is imposed by the theory of quantum mechanics.
- The two conditions imply that every quantum operation F can be described by a $2^m \times 2^m$ unitary matrix.

Composing Quantum Operations

Lemma (Composition of Quantum Operations)

Let A_1, A_2 be matrices representing quantum operations.
Then their composition, i.e., applying A_1 followed by applying A_2, is also a quantum operation, whose matrix is

$$A_2 A_1.$$

Invertibility and Specification

- Consider a quantum operation.

  Its matrix A is unary.

  Thus, it satisfies $AA^* = I$.

  So every quantum operation has a corresponding "inverse" operation that cancels it.

  That is, quantum computation is "reversible".
- Quantum operations are linear.

  So it suffices to describe their behavior on any linear basis for the space $\mathbb{C}^{2^m}$.

  So we often specify quantum operations by their action on a basis.
- Finally, we note that not every classical operation is unitary.

  So designing quantum operations requires care.

  In particular, one should avoid careless transfer of classical operations.

Flipping Qubits

- Suppose we wish to "flip" the first qubit in an m-qubit register.
- That is, we want to apply the NOT operation on the first qubit.
- This can be done as a quantum operation that operates on basis states by stipulating that, for all $b \in \{0,1\}$, $x \in \{0,1\}^{m-1}$,

$$|b, x\rangle \mapsto |1-b, x\rangle.$$

- The matrix of this operation is a permutation on the standard basis.
- Permutation matrices are always unitary.

Notation

- The preceding example involves an operation on the first qubit.
- So the remaining qubits in x are unaffected and unnecessarily cluttering the notation.
- To unclutter, when we describe operations on only a subset of qubits, we will often drop the unaffected qubits from the notation.
- Accordingly, the NOT operation can be described simply as

$$|0\rangle \mapsto |1\rangle \quad \text{and} \quad |1\rangle \mapsto |0\rangle.$$

Reordering Qubits

- Suppose we wish to exchange the values of two qubits.
- The following operation suffices:

$$|00\rangle \mapsto |00\rangle,\ |01\rangle \mapsto |10\rangle,\ |10\rangle \mapsto |01\rangle,\ |11\rangle \mapsto |11\rangle.$$

- It is unitary, since it is a permutation of basic states.
- This operation is described by the following $2^2 \times 2^2$ matrix, where we index the rows and columns according to lexicographical order $|00\rangle, |01\rangle, |10\rangle, |11\rangle$:

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

- By combining such operations we can arbitrarily reorder the qubits of an m-qubit register.

Copying Qubits

- Suppose we wish to copy the first qubit into the second.
- Proceeding naively, we might try:

$$|00\rangle \mapsto |00\rangle,\ |01\rangle \mapsto |00\rangle,\ |10\rangle \mapsto |11\rangle,\ |11\rangle \mapsto |11\rangle.$$

- However, this is not a reversible operation.
- Hence, it is not unitary.
- In fact, the so-called No Cloning Theorem rules out any quantum operation that copies qubits.

Copying Qubits (Cont'd)

- While designing quantum algorithms it usually suffices to copy a qubit in "write once" fashion.
- This is done by keeping around a supply of fresh qubits in a predetermined state, say $|0\rangle$, and only writing them over once.
- The operation

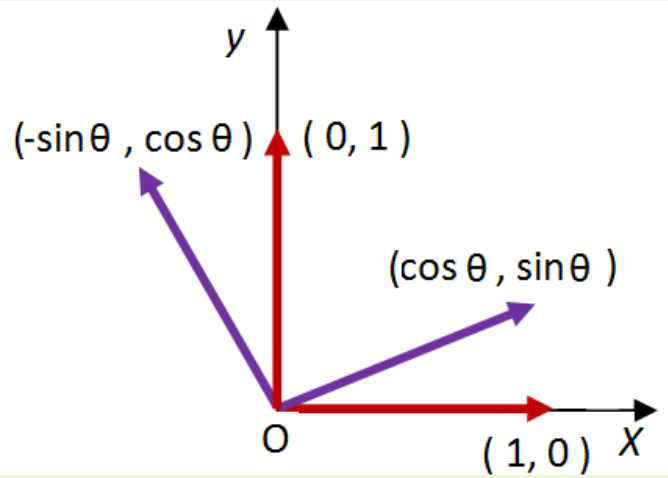
$$|xy\rangle \mapsto |x(x \oplus y)\rangle$$

  provides the effect of copying the first qubit, assuming the algorithm designer takes care to apply it only where the second qubit is a fresh, i.e., unused, qubit in state $|0\rangle$.
- As intended, the operation never receives input $|01\rangle$ or $|11\rangle$.
- It negates the second qubit y if and only if x is in the state $|1\rangle$.
- It is known as the **controlled NOT** (**CNOT** for short).

Rotation on Single Qubit

- Think of the phase of a qubit as a two-dimensional vector.
- We may wish to apply a rotation to this state vector by an angle θ.
- This corresponds to the operation

$$|0\rangle \mapsto \cos\theta|0\rangle + \sin\theta|1\rangle, \quad |1\rangle \mapsto -\sin\theta|0\rangle + \cos\theta|1\rangle.$$



It is described by the matrix

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

which is unitary.

- Note that when $\theta = \pi$, i.e., 180°, this amounts to flipping the sign of the state vector mapping $\boldsymbol{v}$ to $-\boldsymbol{v}$.

AND of Two Bits

- Consider the classical AND operation.
- More concretely, this operation replaces the first qubit of the register by the AND of the first two bits.
- One might think of this as a linear operation

$$|b_1 b_2\rangle \mapsto |b_1 \wedge b_2\rangle |b_2\rangle, \quad b_1, b_2 \in \{0, 1\}.$$

- This is unfortunately not reversible and, hence, not unitary.

Reversible AND of Two Bits

- There is a different way to achieve the effect of an AND operation.
- It uses a "reversible AND", which has an additional scratchpad in the form of a fresh qubit b_3.
- The operation is given, for all $b_1, b_2, b_3 \in \{0, 1\}$, by

$$|b_1\rangle|b_2\rangle|b_3\rangle \mapsto |b_1\rangle|b_2\rangle|b_3 \oplus (b_1 \wedge b_2)\rangle.$$

- This operation is unitary (permutation matrix).
- Thus, it is a valid quantum operation.

Reversible AND of Two Bits (Cont'd)

- The operation "reversible AND"

$$|b_1\rangle|b_2\rangle|b_3\rangle \mapsto |b_1\rangle|b_2\rangle|b_3 \oplus (b_1 \wedge b_2)\rangle,$$

for all $b_1, b_2, b_3 \in \{0,1\}$, is described by the following matrix

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

AND and OR of Two Bits

- The algorithm designer will only apply the "reversible AND" operation when b_3 is a fresh qubit in state $|0\rangle$.
- The "reversible AND" operation is also known in quantum computing as the **Toffoli gate**.
- One can similarly obtain a "reversible OR" quantum operation.
- Together, the reversible OR and AND gates are key in showing that quantum computers can simulate ordinary Turing machines.

The Hadamard Operation

- The **Hadamard gate** is the single qubit operation that (up to normalization) maps

$$|0\rangle \mapsto |0\rangle + |1\rangle, \quad |1\rangle \mapsto |0\rangle - |1\rangle.$$

- More succinctly the state

$$|b\rangle \mapsto |0\rangle + (-1)^b |1\rangle.$$

- The corresponding matrix is

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

The Hadamard Operation

- Suppose we apply a Hadamard gate to every qubit of an m-qubit register.
- Then, for every $x \in \{0,1\}^m$, the state $|x\rangle$ is mapped to

$$\begin{aligned}(|0\rangle + (-1)^{x_1}|1\rangle)(|0\rangle + (-1)^{x_2}|1\rangle)\cdots(|0\rangle + (-1)^{x_m}|1\rangle) \\ = \sum_{y\in\{0,1\}^m} \left(\prod_{i:y_i=1}(-1)^{x_i}\right)|y\rangle \\ = \sum_{y\in\{0,1\}^m} -1^{x\odot y}|y\rangle,\end{aligned}$$

  where $x \odot y$ denotes the dot product modulo 2 of x and y.
- The unitary matrix corresponding to this operation is the $2^m \times 2^m$ matrix whose (x,y)-th entry is

$$\frac{-1^{x\odot y}}{\sqrt{2^n}}.$$

Elementary Quantum Operations or Quantum Gates

Definition (Elementary Quantum Operations or Quantum Gates)

A quantum operation is called **elementary**, or sometimes a **quantum gate**, if it acts on three or less qubits of the register.

- Note that an elementary operation on an m-qubit register can be specified by three indices in $[m]$ and an 8×8 unitary matrix.
- Example: Suppose U is any 8×8 unitary matrix that has to be applied to the qubits numbered 2, 3, 4.

  It can be viewed as an elementary quantum operation $F : \mathbb{C}^{2^m} \to \mathbb{C}^{2^m}$ that, for all $x_1, x_2, \ldots, x_m \in \{0, 1\}$,

$$|x_1 x_2 \ldots x_m\rangle \mapsto |x_1\rangle (U|x_2 x_3 x_4\rangle)|x_5 \ldots x_m\rangle.$$

Quantum computation and BQP

Definition (Quantum Computation and the Class BQP)

Consider functions $f : \{0,1\}^* \to \{0,1\}$ and $T : \mathbb{N} \to \mathbb{N}$.
We say that f is **computable in quantum $T(n)$-time** if there is a polynomial-time classical TM that, on input $(1^n, 1^{T(n)})$, for any $n \in \mathbb{N}$, outputs the descriptions of quantum gates

$$F_1, \ldots, F_T,$$

such that, for every $x \in \{0,1\}^n$, we can compute $f(x)$ by the following process with probability at least $\frac{2}{3}$.

1. Initialize an m qubit quantum register to the state

$$\left|x0^{n-m}\right\rangle,$$

   i.e., x padded with zeroes, where $m \leq T(n)$.

Quantum computation and BQP (Cont'd)

Definition (Quantum Computation and the Class BQP Cont'd)

2. Apply one after the other $T(n)$ elementary quantum operations

$$F_1, \ldots, F_T$$

to the register.

3. Measure the register and let Y denote the obtained value.

That is, if $\boldsymbol{v}$ is the final state of the register, then Y is a random variable that takes the value y with probability $|\boldsymbol{v}_y|^2$, for every $y \in \{0,1\}^m$.

4. Output Y_1.

A Boolean function $f : \{0,1\}^* \to \{0,1\}$ is in BQP if there is some polynomial $p : \mathbb{N} \to \mathbb{N}$, such that f is computable in quantum $p(n)$-time.

Remarks on Quantum Computation

1. The definition of quantum computation generalizes to functions with more than one output bit.
2. Elementary operations are represented by 8×8 matrices of complex numbers, which a classical TM cannot write per se.

   However, it suffices for the TM to write the most significant $O(\log T(n))$ bits of the complex number.
3. The set of elementary operations or gates (which is an infinite set) can be reduced to two universal operations.

Remarks on Quantum Computation

4. The definition of quantum computation disallows several features allowed by quantum mechanics. such as mixed states that involve both quantum superposition and probability and measurement in bases different than the standard basis.

   However, none of these features adds to the computing power.

   Another disallowed feature is performing partial measurements of some of the qubits in the course of the computation.

   However, those can always be eliminated without much loss of efficiency.

Quantum versus Probabilistic Computation

- The fact that the states of registers are described by 2^m-dimensional vectors and operations are described by $2^m \times 2^m$ matrices does not give exponential speedup.
- Even ordinary probabilistic computation can be similarly described.
  - The state of a classical m-bit register can be thought of as a 2^m-dimensional vector whose x-th coordinate denotes the probability that the register contains the string x;
  - Probabilistic operations can be thought of as linear stochastic maps from $\mathbb{R}^{2^m}$ to $\mathbb{R}^{2^m}$.
- The added power of quantum computing seems to derive from the following facts.
  - We allow vectors to have negative coefficients;
  - The norm that is preserved at each step is the Euclidean, i.e., ℓ_2, norm rather than the sum, i.e., ℓ_1, norm.
- Note also that classical computation, whether deterministic or probabilistic, is a subcase of quantum computation.

Quantum Circuits

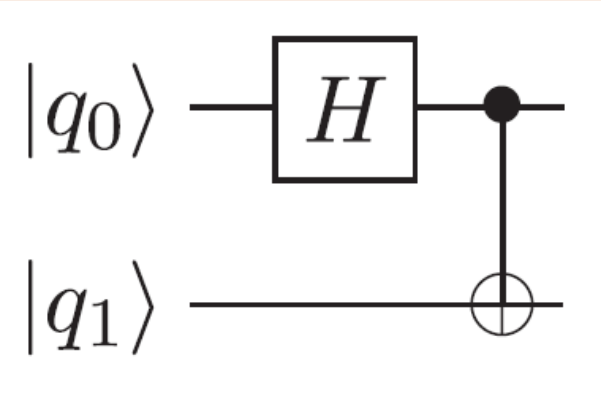
- The definition of quantum computation parallels the definition of classical straight-line programs.
- These constitute an equivalent model to Boolean circuits.
- Similarly, one can define quantum computation and BQP also in terms of quantum circuits.
- These are directed acyclic graphs with:
  - Sources (vertices with in-degree zero) denoting the inputs;
  - Sinks (vertices with out-degree zero) denoting the outputs;
  - Internal nodes denoting the gates.

Quantum Circuits

- The gates are labeled, instead of by the operations AND,OR and NOT, by 2×2, 4×4 or 8×8 unitary matrices.
- Since copying is not allowed, the out-degree of gates and even inputs cannot be arbitrarily large.
- More precisely:
  - The out-degree of each input vertex is one;
  - The in-degree and out-degree of each gate are equal (and at most 3).
- We also allow special "workspace", or "scratchpad", inputs that are initialized to the state $|0\rangle$.

Quantum Circuits: An Illustration

- Quamtum circuits are often described using diagrams.



- The diagram above depicts a quantum circuit.
- On receiving input $|q_0\rangle|q_1\rangle$, it operates as follows.
  - It first applies the Hadamard operation on $|q_0\rangle$;
  - It then applies the mapping $|q_0 q_1\rangle \mapsto |q_0(q_0 \oplus q_1)\rangle$.

Classical as a Subcase of Quantum Computation

- We saw quantum implementations of the classical NOT and AND.
- We can efficiently quantum simulate any classical computation.

Lemma (Boolean Circuits as a Subcase of Quantum Circuits)

Let $f : \{0,1\}^n \to \{0,1\}^m$ be computable by a Boolean circuit of size S. Then there is a sequence of $2S + m + n$ quantum operations computing the mapping

$$|x\rangle \left|0^{2m+S}\right\rangle \mapsto |x\rangle |f(x)\rangle \left|0^{S+m}\right\rangle.$$

- We replace each Boolean gate (AND, OR, NOT) by its quantum analog.

Classical as a Subcase of Quantum Computation (Cont'd)

- The resulting computation maps

$$|x\rangle\left|0^{2m}\right\rangle\left|0^{S}\right\rangle \mapsto |x\rangle|f(x)0^{m}\rangle|z\rangle,$$

where:

- z is the string of values taken by the internal wires in the Boolean circuit (these correspond to scratchpad memory used by the quantum operations at the gates);
- The string 0^m consists of qubits unused so far.

Classical as a Subcase of Quantum Computation (Cont'd)

- Now copy $f(x)$ onto the string 0^m using m operations of the form

$$|bc\rangle \mapsto |b(b \oplus c)\rangle.$$

We have created

$$|x\rangle|f(x)f(x)\rangle|z\rangle.$$

Run the operations corresponding to the Boolean operations in reverse (applying the inverse of each operation).

This erases the original copy of $f(x)$ as well as $|z\rangle$.

It leaves behind clean bits in state $|0\rangle$, together with one copy of $f(x)$.

Simulation of Probabilistic by Quantum Computation

- A classical Turing machine computation running in $T(n)$ steps has an equivalent Boolean circuit of size $O(T(n) \log T(n))$.
- It follows that $P \subseteq BQP$.
- Using the Hadamard operation that maps $|0\rangle$ to $|0\rangle + |1\rangle$, we can get a qubit that, when measured, gives:
  - $|0\rangle$ with probability $\frac{1}{2}$;
  - $|1\rangle$ with probability $\frac{1}{2}$.
- That is, this qubit simulates a coin toss.
- As a consequence, we obtain

Corollary

$BPP \subseteq BQP$.

Universal Operations

- Allowing every three-qubit quantum operation as "elementary" seems problematic since this set is infinite.
- Classical Boolean circuits only need the gates AND, OR and NOT.
- Fortunately, a similar result holds for quantum computation.

Theorem (Universal Basis for Quantum Operations)

For every $D \geq 3$ and $\epsilon > 0$, there is $\ell \geq 100(D \log \frac{1}{\epsilon})^3$, such that the following is true:

- Every $D \times D$ unitary matrix U can be approximated as a product of unitary matrices $U_1, \ldots, U_\ell$, in the sense that its (i,j)-th entry for each $i, j \leq D$, satisfies

$$|U_{i,j} - (U_\ell \cdots U_1)_{i,j}| < \epsilon;$$

Universal Operations (Cont'd)

Theorem (Universal Basis for Quantum Operations Cont'd)

- Each U_r corresponds to applying one of the following on at most three qubits:
  - The Hadamard gate
$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix};$$
  - The Toffoli gate
$$|abc\rangle \mapsto |ab(c \oplus (a \wedge b))\rangle;$$
  - The phase shift gate
$$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

Universality and Time

- It can be shown that such an ϵ-approximation for, say, $\epsilon < \frac{1}{10T}$ suffices for simulating any T-time quantum computation.
- Hence, we can replace any computation using T arbitrary elementary matrices by a computation using only one of the above three gates.
- Other universal gates are also known.
- In particular, for the purpose of quantum computation, the Hadamard and Toffoli gates alone suffice (complex numbers are not necessary for quantum computation).
- One corollary is that three-qubit gates can be used to simulate k-qubit gates for every constant $k > 3$ (but at a cost exponential in k, since the representation is by $2^k \times 2^k$ matrices).
- This means that when designing quantum algorithms, we can consider every k-qubit gate as elementary, as long as k is smaller than some absolute constant.

Universality and Time

- When designing quantum algorithms, we can consider every k-qubit gate as elementary as long as k is smaller than some absolute constant.
- We can use this fact to obtain a quantum analog of the "`if` cond `then`" construct of classical programming languages.

  Given a T step quantum circuit for an n-qubit quantum operation U, then we can compute the quantum operation Controlled-U in $O(T)$ steps, where Controlled-U maps:

$$|x_1 \dots x_n x_{n+1}\rangle \mapsto \begin{cases} |U(x_1 \dots x_n)x_{n+1}\rangle, & \text{if } x_{n+1} = 1; \\ |x_1 \dots x_n x_{n+1}\rangle, & \text{otherwise.} \end{cases}$$

- The reason is:
  - We can transform every elementary operation F in the computation of U to the analogous "Controlled-F" operation.
  - Since the "Controlled-F" operation depends on at most four qubits, it too can be considered elementary.

Subsection 4

Grover's Search Algorithm

Satisfiability Revisited

- Recall the NP-complete problem $\textsc{Sat}$.
- Given an n-variable Boolean formula φ, we ask whether there exists an assignment $a \in \{0,1\}^n$, such that

$$\varphi(a) = 1.$$

- Using "classical" deterministic or probabilistic TM's, we do not know how to solve this problem better than in $\text{poly}(n)2^n$-time.

Grover's Search Algorithm

- Grover's algorithm solves $\textsc{Sat}$ in $\text{poly}(n)2^{n/2}$-time on a quantum computer.
- This is a significant improvement over the classical case, even if way short of showing NP $\subseteq$ BQP.
- Grover's algorithm solves an even more general problem, namely, satisfiability of a circuit with n inputs.

Theorem (Grover's Algorithm)

There is a quantum algorithm that, given as input a polynomial-time computable function

$$f : \{0,1\}^n \to \{0,1\},$$

represented as a circuit computing f, finds in $\text{poly}(n)2^{n/2}$ time a string a, such that $f(a) = 1$, if such a string exists.

The Setting for Grover's Algorithm

- Grover's algorithm is best described geometrically.
- We assume that the function f has a single satisfying assignment a.
- Consider an n-qubit register.
- Let $\boldsymbol{u}$ denote the uniform state vector of this register, i.e.,

$$\boldsymbol{u} = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

The Setting for Grover's Algorithm (Cont'd)

- The angle between $\boldsymbol{u}$ and the basis state $|a\rangle$ is equal to the inverse cosine of their inner product

$$\langle \boldsymbol{u}, |a\rangle \rangle = \frac{1}{2^{n/2}}.$$

- Since this is a positive number, this angle is smaller than $\frac{\pi}{2}$ (90), and hence we denote it by

$$\frac{\pi}{2} - \theta, \quad \sin\theta = \frac{1}{2^{n/2}}.$$

- Using the inequality $\theta \geq \sin\theta$, for $\theta > 0$, we get

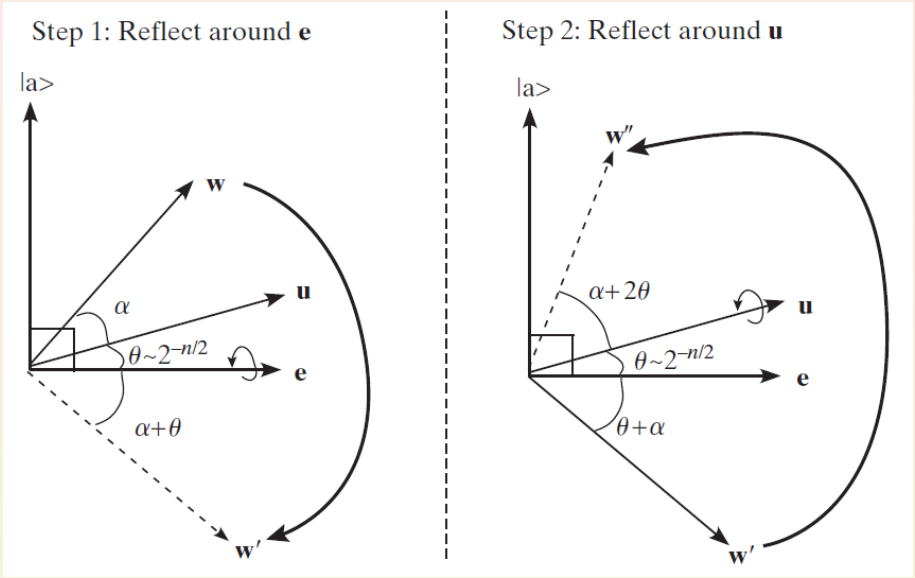
$$\theta \geq 2^{-n/2}.$$

General Description of Grover's Algorithm

- The algorithm starts with the state $\boldsymbol{u}$.
- At each step it gets nearer the state $|a\rangle$.
- If its current state makes an angle $\frac{\pi}{2} - \alpha$ with $|a\rangle$, then at the end of the step it makes an angle $\frac{\pi}{2} - \alpha - 2\theta$.
- Thus, in $\mathrm{O}\left(\frac{1}{\theta}\right) = \mathrm{O}\left(2^{n/2}\right)$ steps, it will get to a state $\boldsymbol{v}$ whose inner product with $|a\rangle$ is larger than, say, $\frac{1}{2}$.
- This implies that a measurement of the register will yield a with probability at least $\frac{1}{4}$.

Idea Behind Grover's Algorithm

- The main idea is that to rotate a vector $\boldsymbol{w}$ toward the unknown vector $|a\rangle$ by an angle of θ, it suffices to take two reflections.



  - First around the vector

$$\boldsymbol{e} = \sum_{x \neq a} |x\rangle,$$

    that is the vector orthogonal to $|a\rangle$ on the plane spanned by $\boldsymbol{u}$ and $|a\rangle$;
  - Second, around the vector $\boldsymbol{u}$.

Setting Up the Reflections

- To complete the algorithm's description, we need to show how we can perform the reflections around the vectors $\boldsymbol{u}$ and $\boldsymbol{e}$.
- We must show how we can transform in polynomial time a state $\boldsymbol{w}$ of the register into the state that is $\boldsymbol{w}$'s reflection around $\boldsymbol{u}$ (respectively, $\boldsymbol{e}$).
- Instead of working with an n-qubit register, we work with an m-qubit register for m that is polynomial in n.
- The extra qubits will only serve as "scratch workspace".
- They will always contain zero except during intermediate computations, and, hence, can be safely ignored.

Reflecting around e

- To reflect a vector w around a vector v:
  - We express w as
$$\alpha v + v^{\perp},$$
where $v^{\perp}$ is orthogonal to v;
  - We output
$$\alpha v - v^{\perp}.$$
- Thus, the reflection of w around e is equal to
$$\sum_{x \neq a} w_x |x\rangle - w_a |a\rangle.$$

Reflecting around e (Cont'd)

- To perform this transformation:
  1. Since f is computable in polynomial time, we can compute the transformation $|x\sigma\rangle \mapsto |x(\sigma \oplus f(x))\rangle$ in polynomial time.
     This transformation maps $|x0\rangle$ to $|x0\rangle$, for $x \neq a$, and $|a0\rangle$ to $|a1\rangle$.
  2. Then, we apply the elementary transformation (known as a Z gate) $|0\rangle \mapsto |0\rangle$, $|1\rangle \mapsto -|1\rangle$ on the qubit σ.
     This maps $|x0\rangle$ to $|x0\rangle$, for $x \neq a$, and maps $|a1\rangle$ to $-|a1\rangle$.
  3. Then, we apply the transformation $|x\sigma\rangle \mapsto x(\sigma \oplus f(x))$ again.
     This maps $|x0\rangle$ to $|x0\rangle$, for $x \neq a$, and maps $|a1\rangle$ to $|a0\rangle$.

  The final result is that the vector $|x0\rangle$ is mapped to itself, for $x \neq a$, but $|a0\rangle$ is mapped to $-|a0\rangle$.

  Ignoring the last qubit, this is exactly a reflection around $|a\rangle$.

Reflecting around $\boldsymbol{u}$

- To reflect around $\boldsymbol{u}$:
  - First, apply the Hadamard operation to each qubit, mapping $\boldsymbol{u}$ to $|0\rangle$.
  - Then, reflect around $|0\rangle$.

    This can be done in the same way as reflecting around $|a\rangle$.
    We use the function $g : \{0,1\}^n \to \{0,1\}$ that outputs 1 iff its input is all zeroes instead of f.
  - Then, apply the Hadamard operation again, mapping $|0\rangle$ back to $\boldsymbol{u}$.
- Together the two operations allow us to take a vector in the plane spanned by $|a\rangle$ and $\boldsymbol{u}$ and rotate it 2θ radians closer to $|a\rangle$.

  Thus, if we start with the vector $\boldsymbol{u}$, we will only need to repeat them $\mathrm{O}\left(\frac{1}{\theta}\right) = \mathrm{O}\left(2^{n/2}\right)$ times to obtain a vector that, when measured, yields $|a\rangle$ with constant probability.

Subsection 5

Simon's Algorithm

Simon's Problem

- Assume given a polynomial-size classical circuit for a function

$$f : \{0,1\}^n \to \{0,1\}^n,$$

such that there exists $a \in \{0,1\}^n$, satisfying, for all $x, y \in \{0,1\}^n$,

$$f(x) = f(y) \quad \text{iff} \quad x = y \oplus a.$$

- Find this string a, called the "**period**" of f.

Theorem (Simon's Algorithm)

There is a polynomial-time quantum algorithm for Simon's problem.

Questions

(1) Why is Simon's Problem interesting?
- A generalization of Simon's problem turns out to be crucial in the quantum polynomial-time algorithm for the integer factorization.

(2) Why do we believe it is hard to solve for classical computers?
- We do not know for certain that this problem does not have a classical polynomial-time algorithm.
- If P = NP, then there obviously exists such an algorithm.
- We next give a rational for believing this not to be the case.

Simon's Problem May Be Hard for Classical Computers

- Why do we believe it is hard to solve for classical computers?
- Suppose that we are given access to a black box or an oracle that, on input $x \in \{0,1\}^n$, returns the value $f(x)$.
- Would we be able to learn a by making at most a subexponential number of queries to the black box?
- Assume that:
  - a is chosen at random from $\{0,1\}^n$,
  - f is chosen at random subject to the condition that $f(x) = f(y)$ iff $x = y \oplus a$,
- Then, it can be seen that no classical algorithm can successfully recover a with reasonable probability using significantly less than $2^{n/2}$ queries to the black box.
  - Suppose an algorithm uses fewer queries.
  - It is very likely to never get the same answer to two distinct queries.
  - In that case it gets no information about the value of a.

Proof of Simon's Algorithm

- Simon's algorithm uses a register of $2n + m$ qubits, where m is the number of workspace bits needed to compute f.
- The last m qubits of the register will be always set to all zeroes except in intermediate steps of f's computation.
- So, in the description, we ignore those qubits.
- We use n Hadamard operations to uniformize the first n qubits.
- We apply $|xz\rangle \mapsto |x(z \oplus f(x))\rangle$ to the register.
- This results in the state

$$\sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle = \sum_{x \in \{0,1\}^n} (|x\rangle + |x \oplus a\rangle)|f(x)\rangle.$$

- We measure the second n bits.
- This collapses the state to

$$|xf(x)\rangle + |(x \oplus a)f(x)\rangle,$$

for some string x (uniform in $\{0,1\}^n$).

Proof of Simon's Algorithm (Cont'd)

- Suppose we measure the first n bits.
- Then we will get:
  - With probability $\frac{1}{2}$ the value x;
  - With probability $\frac{1}{2}$ the value $x \oplus a$.
- a can be deduced from these two values combined, but not from each one of them on its own.
- Perform n Hadamard operations on the first n bits.
- This maps x to

$$\sum_y ((-1)^{x \odot y} + (-1)^{(x \oplus a) \odot y})|y\rangle = \sum_y ((-1)^{x \odot y} + (-1)^{x \odot y}(-1)^{a \odot y})|y\rangle.$$

- For every $y \in \{0,1\}^m$, the y-th coefficient is nonzero if and only if $a \odot y = 0$.
- If measured, we get a uniform $y \in \{0,1\}^n$, with $a \odot y = 0$.

Proof of Simon's Algorithm (Cont'd)

- Repeating the entire process k times, we get k uniform strings $y_1, \ldots, y_k$, satisfying

$$y_i \odot a = 0, \quad i = 1, \ldots, k.$$

- In other words, k linear equations over GF(2) on the variables $a_1, \ldots, a_n$.
- If, say, $k \geq 2n$, then with high probability there will be $n - 1$ linearly independent equations among these.
- Hence, we will be able to retrieve a from these equations using Gaussian elimination.

Subsection 6

Shor's Algorithm: Integer Factorization

Shor's Algorithm: Factoring in BQP

- The integer factorization problem is to find, given an integer N, the set of all prime factors of N, i.e., prime numbers that divide N.
- By a polynomial-time algorithm for this problem, we mean an algorithm that runs in time polynomial in the description of N, i.e., poly($\log N$)-time.
- No classical polynomial time algorithm is known.
- The best classical algorithm takes roughly $2^{(\log N)^{1/3}}$ steps to factor N.
- In 1994 Shor devised a famous algorithm in quantum computing.
- It provides the strongest evidence that BQP may contain problems outside of BPP.

Theorem (Shor's Algorithm: Factoring in BQP)

There is a quantum algorithm that, given a number N, runs in time poly($\log N$) and outputs the prime factorization of N.

The Ideas in the Algorithm

- The algorithm uses the following observations.
- First, N has at most $\log N$ factors.

  So it suffices to find a single factor of N in poly($\log N$) time.

  One can then repeat the algorithm with N divided by that factor, and, thus, find all factors.
- It is a well-known fact that in order to find a single factor, it suffices to be able to find the order of a random number $A \pmod N$.

  This is the smallest r, such that $A^r \equiv 1 \pmod N$.

  The idea is that with good probability:
  - The order r of A will be even;
  - $A^{r/2} - 1$ will have a nontrivial common factor with N, which we can find using a GCD computation.

Using The Ideas in the Algorithm

- The mapping $A \mapsto A^x \pmod N$ is computable in poly$(\log N)$ time even on classical TMs using fast exponentiation.
- Using these, we devise a polylog(N)-time quantum algorithm that transforms an all zeros state into the uniform superposition of all $|x\rangle$, $x \leq N$, satisfying

$$A^x \equiv y_0 \pmod N,$$

for some randomly chosen $y_0 \leq N - 1$.

- These form an arithmetic progression

$$x_0 + ri, \quad i = 1, 2, \ldots,$$

where:

  - $A^{x_0} \equiv y_0 \pmod N$;
  - r is the order of A.

The Plan for Building Shor's Algorithm

- Like in Simon's Problem:
  - We have created a quantum state involving a strong periodicity, an arithmetic progression.
  - We are interested in determining its period.
- A classical tool for detecting periods is the Fourier Transform.

Quantum Fourier Transform

- The Quantum Fourier Transform (QFT) allows us to detect periods in a quantum state.
- It takes a register from some arbitrary state $f \in \mathbb{C}^M$ into a state whose vector is the Fourier transform $\hat{f}$ of f.
- The QFT takes only $O\left(\log^2 M\right)$ elementary steps and is thus very efficient.
- We cannot say that this algorithm "computes" the Fourier transform, since the transform is stored in the amplitudes of the state.
- The only way to get information from a state is by measuring it.
- This yields a single basis state with probability that is related to its amplitude.

The Fourier transform over $\mathbb{Z}_M$

Definition (Fourier transform over $\mathbb{Z}_M$)

For every vector $f \in \mathbb{C}^M$, the **Fourier transform** of f is the vector $\hat{f}$ where the x-th coordinate of $\hat{f}$ is

$$\hat{f}(x) = \frac{1}{\sqrt{M}} \sum_{y \in \mathbb{Z}_M} f(y)\omega^{xy}, \quad \omega = e^{2\pi i/M}.$$

- The Fourier transform is a representation of f in the **Fourier basis** $\{\chi_x\}_{x \in \mathbb{Z}_M}$, where χ_x is the vector/function whose y-th coordinate is

$$\chi_x(y) = \frac{1}{\sqrt{M}}\omega^{xy}.$$

Orthonormality of the Basis

- The inner product of any two vectors χ_x, χ_z in this basis is equal to

$$\langle \chi_x, \chi_z \rangle = \frac{1}{M} \sum_{y \in \mathbb{Z}_M} \omega^{xy} \overline{\omega^{zy}} = \frac{1}{M} \sum_{y \in \mathbb{Z}_M} \omega^{(x-z)y}.$$

  - If $x = z$, then $\omega^{(x-z)} = 1$.
    Hence, this sum is equal to 1.
  - If $x \neq z$, then this sum is equal to

$$\frac{1}{M} \frac{1 - \omega^{(x-y)M}}{1 - \omega^{x-y}} = \frac{1}{M} \frac{1 - 1}{1 - \omega^{x-y}} = 0.$$

- Thus, this is an orthonormal basis.
- So the Fourier transform map $f \mapsto \hat{f}$ is a unitary operation.

The Fourier Basis

- Identify vectors in $\mathbb{C}_M$ with functions mapping $\mathbb{Z}_M$ to $\mathbb{C}$.
- It can be seen that every function χ in the Fourier basis is a **homomorphism** from $\mathbb{Z}_M$ to $\mathbb{C}$, in the sense that

$$\chi(y+z) = \chi(y)\chi(z), \text{ for all } y, z \in \mathbb{Z}_M.$$

- Also, every function χ is **periodic**, in the sense that there exists $r \in \mathbb{Z}_M$, such that

$$\chi(y+r) = \chi(y), \text{ for every } y \in \mathbb{Z}_M.$$

Periodicity and Fourier Transform

- If $\chi(y) = \omega^{xy}$, then we can take r to be $\frac{\ell}{x}$, where ℓ is the least common multiple of x and M.
- Suppose a function $f : \mathbb{Z}_M \to \mathbb{C}$ is itself periodic (or roughly periodic).
- Suppose we represent f in the Fourier basis.
- Then, intuitively, the coefficients of basis vectors with periods agreeing with the period of f should be large.
- So we might be able to discover f's period from this representation.
- This is key in Shor's algorithm.

Introducing the Fast Fourier Transform

- Denote by FT_M the operation that maps every vector $f \in \mathbb{C}^M$ to its Fourier transform $\hat{f}$,

$$\begin{array}{rccc} \mathrm{FT}_M : & \mathbb{C}^M & \to & \mathbb{C}^M; \\ & f & \mapsto & \hat{f}. \end{array}$$

- The operation FT_M is represented by an $M \times M$ matrix whose (x, y)-th entry is ω^{xy}.
- The trivial algorithm to compute it takes M^2 operations.
- The famous **fast Fourier transform** (**FFT**) algorithm computes the Fourier transform in $\mathrm{O}(M \log M)$ operations.

Fast Fourier Transform

- We sketch the idea behind the FFT algorithm.

$$\begin{aligned}\hat{f}(x) &= \frac{1}{\sqrt{M}}\sum_{y\in\mathbb{Z}_M} f(y)\omega^{xy} \\ &= \frac{1}{\sqrt{M}}\sum_{y\in\mathbb{Z}_M, y \text{ even}} f(y)\omega^{-2x(y/2)} \\ &\qquad + \omega^x \frac{1}{\sqrt{M}}\sum_{y\in\mathbb{Z}_M, y \text{ odd}} f(y)\omega^{2x(y-1)/2}.\end{aligned}$$

- ω^2 is an $\frac{M}{2}$-th root of unity.
- Moreover,

$$\omega^{M/2} = -1.$$

- For an M-dimensional vector $\boldsymbol{v}$, we denote by:
  - $\boldsymbol{v}_{\text{even}}$ ($\boldsymbol{v}_{\text{odd}}$) the $\frac{M}{2}$-dimensional vector obtained by restricting $\boldsymbol{v}$ to the coordinates whose indices have least significant bit equal to 0 (1);
  - $\boldsymbol{v}_{\text{low}}$ ($\boldsymbol{v}_{\text{high}}$) the restriction of v to coordinates with most significant bit 0 (1).

Fast Fourier Transform (cont'd)

- We have

$$\hat{f}(x) = \frac{1}{\sqrt{M}} \sum_{y \in \mathbb{Z}_M, y \text{ even}} f(y)\omega^{-2x(y/2)} + \omega^x \frac{1}{\sqrt{M}} \sum_{y \in \mathbb{Z}_M, y \text{ odd}} f(y)\omega^{2x(y-1)/2}.$$

- Let W be the $\frac{M}{2} \times \frac{M}{2}$ diagonal matrix with diagonal entries $\omega^0, \ldots, \omega^{M/2-1}$.
- From the remarks above, we get:

$$\begin{aligned} \mathrm{FT}_M(f)_{\text{low}} &= \mathrm{FT}_{M/2}(f_{\text{even}}) + W\mathrm{FT}_{M/2}(f_{\text{odd}}); \\ \mathrm{FT}_M(f)_{\text{high}} &= \mathrm{FT}_{M/2}(f_{\text{even}}) - W\mathrm{FT}_{M/2}(f_{\text{odd}}). \end{aligned}$$

- So we may replace a size M problem with two size $\frac{M}{2}$ subproblems.
- Thus, we obtain a recursive time bound of the form

$$T(M) = 2T\left(\frac{M}{2}\right) + \mathrm{O}(M).$$

- It shows that $T(M) = \mathrm{O}(M \log M)$.

Quantum Fourier Transform Over $\mathbb{Z}_M$

Lemma (Quantum Fourier Transform)

For every m and $M = 2^m$, there is a quantum algorithm that uses $O\left(m^2\right)$ elementary quantum operations and transforms a quantum register in state

$$f = \sum_{x \in \mathbb{Z}_m} f(x)|x\rangle$$

into the state

$$\hat{f} = \sum_{x \in \mathbb{Z}_M} \hat{f}(x)|x\rangle,$$

where

$$\hat{f}(x) = \frac{1}{\sqrt{M}} \sum_{y \in \mathbb{Z}_m} \omega^{xy} f(y).$$

Quantum Fourier Transform Over $\mathbb{Z}_M$ (Cont'd)

- The crux of the algorithm consists of the recursive equations which allow the problem of size M, to be split into two identical subproblems of size $\frac{M}{2}$ involving computation of $\mathrm{FT}_{\frac{M}{2}}$.
- The transformation W on $m-1$ qubits can be defined by

$$|x\rangle \mapsto \omega^x = \omega^{\sum_{i=0}^{m-2} 2^i x_i},$$

  where x_i is the i-th qubit of x.
- It is the result of applying, for every $i \in \{0, \ldots, m-2\}$, the following elementary operation on the i-th qubit:

$$|0\rangle \mapsto |0\rangle \quad \text{and} \quad |1\rangle \mapsto \omega^{2^i} |1\rangle.$$

- The final state is equal to $\hat{f}$.

Summary of Quantum Fourier Transform FT_M

- Initial State: $f = \sum_{x \in \mathbb{Z}_m} f(x)|x\rangle$
- Final State: $\hat{f} = \sum_{x \in \mathbb{Z}_M} \hat{f}(x)|x\rangle$

Operation	**State**
	$f = \sum_{x \in \mathbb{Z}_M} f(x)\|x\rangle$
Run $\mathrm{FT}_{\frac{M}{2}}$ on $m-1$ most significant qubits.	$(\mathrm{FT}_{\frac{M}{2}} f_{\text{even}})\|0\rangle + (W\mathrm{FT}_{\frac{M}{2}} f_{\text{odd}})\|1\rangle$
If LSB is 1, then compute W on $m-1$ most significant qubits.	$(\mathrm{FT}_{\frac{M}{2}} f_{\text{even}})\|0\rangle + (W\mathrm{FT}_{\frac{M}{2}} f_{\text{odd}})\|1\rangle$
Apply Hadamard gate H to least significant qubit.	$(\mathrm{FT}_{\frac{M}{2}} f_{\text{even}})(\|0\rangle + \|1\rangle) + (W\mathrm{FT}_{\frac{M}{2}} f_{\text{odd}})(\|0\rangle - \|1\rangle)$ $(\mathrm{FT}_{\frac{M}{2}} f_{\text{even}} + W\mathrm{FT}_{\frac{M}{2}} f_{\text{odd}})\|0\rangle +$ $(\mathrm{FT}_{\frac{M}{2}} f_{\text{even}} - W\mathrm{FT}_{\frac{M}{2}} f_{\text{odd}})\|1\rangle$
Move LSB to the most significant position.	$\|0\rangle(\mathrm{FT}_{\frac{M}{2}} f_{\text{even}} + W\mathrm{FT}_{\frac{M}{2}} f_{\text{odd}}) +$ $\|1\rangle(\mathrm{FT}_{\frac{M}{2}} f_{\text{even}} - W\mathrm{FT}_{\frac{M}{2}} f_{\text{odd}})$

Shor's Order-Finding Algorithm

- The central step in Shor's algorithm is a quantum polynomial time algorithm to find the order of an integer A modulo an integer N.

Lemma

There is a polynomial-time quantum algorithm that on input A, N (represented in binary), finds the smallest r, such that

$$A^r \equiv 1 \pmod{N}.$$

- Let $m = \lceil 5 \log N \rceil$ and let $M = 2^m$.

  Our register will consist of $m + \text{polylog}(N)$ qubits.

Shor's Order-Finding Algorithm (Cont'd)

- The function $x \mapsto A^x \pmod N$ can be computed in polylog(N) time.

  So, we will assume that we can compute the map

$$|x\rangle|y\rangle \mapsto |x\rangle\left|y \oplus \llcorner A^x \pmod N \lrcorner\right\rangle,$$

  where $\llcorner X \lrcorner$ denotes the representation of the number $X \in \{0, \ldots, N-1\}$ as a binary string of length $\log N$.

  The order-finding algorithm uses continued fractions.

  They allow approximating an arbitrary real number α with a rational number $\frac{p}{q}$, where there is a prescribed upper bound on q.

  It will suffice to output r with probability at least $\Omega\left(\frac{1}{\log N}\right)$.

An Unrealistic Special Case

- We perform, first, an analysis for $M = rc$, $c \in \mathbb{Z}$.

  This case is unrealistic (M being a power of 2), but illustrates why Fourier transforms are useful for detecting periods.

  Claim: In this case the value measured, x, will be equal to ac for a random $a \in \{0, \ldots, r-1\}$.

  We show that proving this claim suffices

  More specifically, we show that the claim implies that

$$\frac{x}{M} = \frac{a}{r},$$

  where a is random integer less than r.

An Unrealistic Special Case (Cont'd)

- By the Prime Number Theorem, there at least $\Omega\left(\frac{r}{\log r}\right)$ many primes in $[r-1]$.

  Moreover, r has at most $\log r$ prime factors.

  So all but $\log r$ of these $\Omega\left(\frac{r}{\log r}\right)$ primes are co-prime to r.

  Hence, for every r, at least $\Omega\left(\frac{r}{\log r}\right)$ of the numbers in $[r-1]$ are coprime to r.

  Thus, when the algorithm computes a rational approximation for $\frac{x}{M}$, the denominator it will find will indeed be r.

  Next, we prove the Claim.

An Unrealistic Special Case: Proof of the Claim

- Claim: In the case $M = rc$, $c \in \mathbb{Z}$, the value x measured will be equal to ac, for a random $a \in \{0, \ldots, r-1\}$.

  To prove the claim, we compute for every $x \in \mathbb{Z}_M$, the absolute value of $|x\rangle$'s coefficient before the measurement.

  Up to some normalization factor this is

$$\left|\sum_{\ell=0}^{c-1} \omega^{(x_0+\ell r)x}\right| = |\omega^{x_0 c' c}| \left|\sum_{\ell=0}^{c-1} \omega^{r\ell x}\right| = 1 \cdot \left|\sum_{\ell=0}^{c-1} \omega^{r\ell x}\right|.$$

  - If c does not divide x, then ω^r is a c-th root of unity.
    So $\sum_{\ell=0}^{c-1} \omega^{r\ell x} = 0$.
    Hence,, such an x would be measured with zero probability.
  - If $x = cj$, then $\omega^{r\ell x} = \omega^{rcj\ell} = \omega^{Mj\ell} = 1$.
    Hence, the amplitudes of all such x's are equal, for all $j \in \{0, 2, \ldots, r-1\}$.

The General Case

- Suppose r does not necessarily divide M.

  The measured value x may not satisfy $M \mid xr$.

  We show that with $\Omega\left(\frac{1}{\log r}\right)$ probability:

  (1) xr will be "almost divisible" by M, in the sense that $0 \leq xr \pmod{M} < \frac{r}{10}$;
  (2) $\left\lfloor \frac{xr}{M} \right\rfloor$ is coprime to r.

  Condition (1) implies that $|xr - cM| < \frac{r}{10}$, for $c = \left\lfloor \frac{xr}{M} \right\rfloor$.

  Dividing by rM,

$$\left| \frac{x}{M} - \frac{c}{r} \right| < \frac{1}{10M}.$$

The General Case (Cont'd)

- So $\frac{c}{r}$ is a rational number, such that:
  - It has denominator at most N;
  - Approximates $\frac{x}{M}$ to within

$$\frac{1}{10M} < \frac{1}{4N^4}.$$

It can be shown that such an approximation is unique.

Hence, the algorithm will come up with $\frac{c}{r}$ and output the denominator r.

Thus, we are left to prove that:

- There are $\Omega\left(\frac{r}{\log r}\right)$ values of x that satisfy the above two conditions.
- Each is measured with probability $\Omega\left(\left(\frac{1}{\sqrt{r}}\right)^2\right) = \Omega\left(\frac{1}{r}\right)$.

Number of Values of x

Lemma

There exist $\Omega\left(\frac{r}{\log r}\right)$ values $x \in \mathbb{Z}_M$, such that:

1. $0 < xr \pmod M < \frac{r}{10}$.
2. $\left\lfloor \frac{xr}{M} \right\rfloor$ and r are coprime.

- We prove the lemma for the case that r is coprime to M.

  In this case, the map $x \mapsto rx \pmod M$ is a permutation of $\mathbb{Z}_M^*$.

  There are at least $\Omega(\frac{r}{\log r})$ numbers in $[1 \ldots \frac{r}{10}]$ coprime to r (primes in this range that are not one of r's at most $\log r$ prime factors).

Number of Values of x (Cont'd)

- Hence, there are $\Omega\left(\frac{r}{\log r}\right)$ numbers x, such that

$$rx \pmod{M} = xr - \left\lfloor \frac{xr}{M} \right\rfloor M$$

is in $[1 \ldots \frac{r}{10}]$ and coprime to r.

Suppose $\lfloor \frac{rx}{M} \rfloor$ has a nontrivial shared factor with r.

Then this factor would be shared with $rx \pmod{M}$ as well.

So $\lfloor \frac{rx}{M} \rfloor$ can not have a nontrivial shared factor with r.

Probability of Measurement

Lemma

If x satisfies $0 < xr \pmod M < \frac{r}{10}$ then, before the measurement in the final step of the order-finding algorithm, the coefficient of $|x\rangle$ is $\geq \Omega(\frac{1}{\sqrt{r}})$.

- Let x be such that $0 < xr \pmod M < \frac{r}{10}$.

  The absolute value of $|x\rangle$'s coefficient in the state before the measurement is

$$\frac{1}{\sqrt{K}\sqrt{M}} \left| \sum_{\ell=0}^{K-1} \omega^{\ell r x} \right|, \quad K = \left\lfloor \frac{M - x_0 - 1}{r} \right\rfloor.$$

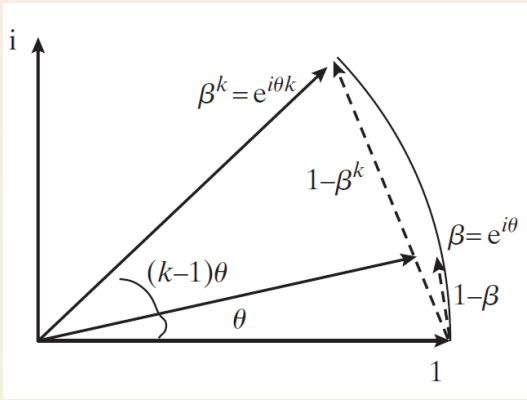
  Since $x_0 < N \ll M$,

$$\frac{M}{2r} < K < \frac{M}{r}.$$

Probability of Measurement (Cont'd)

- Set $\beta = \omega^{rx}$ (since $M \nmid rx$, $\beta \neq 1$).
  Use the formula for the sum of a geometric series, this is

$$\geq \frac{\sqrt{r}}{2M} \left| \frac{1-\beta^{\lceil M/r \rceil}}{1-\beta} \right| = \frac{\sqrt{r}}{2M} \frac{\sin\left(\theta \lceil M/r \rceil/2\right)}{\sin\left(\theta/2\right)},$$



  where $\theta = \frac{rx \pmod M}{M}$ is the angle such that $\beta = e^{i\theta}$.
  Under our assumptions $\lceil \frac{M}{r} \rceil \theta < \frac{1}{10}$.
  Now use $\sin\alpha \sim \alpha$, for small angles α.
  We conclude that the coefficient of x is

$$\geq \frac{\sqrt{r}}{4M} \left\lceil \frac{M}{r} \right\rceil \geq \frac{1}{8\sqrt{r}}.$$

Reducing Factoring to Order Finding: Lemma 1

- The reduction of the factoring problem to the order-finding problem is classical and follows from the following two lemmas.

Lemma

Let N be a non-prime that is not a prime power.
Then, with probability at least $\frac{1}{4}$, a random X in the set

$$\mathbb{Z}_N^* = \{X \in [N-1] : \gcd(X, N) = 1\}$$

satisfies the following:

- It has an even order r;
- $X^{r/2} \neq -1 \pmod{N}$.

Reducing Factoring to Order Finding: Lemma 2

Lemma

Let N and Y be such that

$$Y^2 \equiv 1 \pmod{N} \quad \text{but} \quad Y \pmod{N} \notin \{+1, -1\}.$$

Then $\gcd(Y-1, N) \neq 1, N$.

- Let N be a composite that is not a prime power.

  Let A be random in $[N-1]$.

  By the lemmas, with good probability, one of the following yields a nontrivial factor F of N:
  - $\gcd(A, N)$;
  - $\gcd(A^{r/2} - 1, N)$.

  We can then use recursion to find the prime factors of F and $\frac{N}{F}$.

  Thus, we obtain a polylog(N) time factorization algorithm.

The Second Lemma

Lemma

Let N and Y be such that

$$Y^2 \equiv 1 \pmod{N} \quad \text{but} \quad Y \pmod{N} \notin \{+1, -1\}.$$

Then $\gcd(Y-1, N) \notin 1, N$.

- Under our assumptions:
  - N divides $Y^2 - 1 = (Y-1)(Y+1)$;
  - N does not divide either $Y-1$ or $Y+1$.

This means that $\gcd(Y-1, N) > 1$.

Suppose, to the contrary that $Y-1$ and N were coprime.

Then, since N divides $(Y-1)(Y+1)$, it would divide $Y+1$.

Since $Y-1 < N$, obviously $\gcd(Y-1, N) < N$.

The First Lemma

Lemma

Let N be a non-prime that is not a prime power.
Then, with probability at least $\frac{1}{4}$, a random X in the set

$$\mathbb{Z}_N^* = \{X \in [N-1] : \gcd(X, N) = 1\}$$

has an even order r and satisfies $X^{r/2} \neq -1 \pmod{N}$.

- We prove the lemma for the case $N = PQ$, for primes P, Q.

  The proof can be generalized for every N.

  By the Chinese Remainder Theorem, every $X \in \mathbb{Z}_N^*$ is isomorphic to the pair $\langle X \pmod{P}, X \pmod{Q} \rangle$.

  Thus, choosing random $X \in \mathbb{Z}_N^*$ is equivalent to:
  - Choosing two random Y, Z in $\mathbb{Z}_P^*$ and $\mathbb{Z}_Q^*$, respectively;
  - Setting X to be the unique number corresponding to the pair $\langle Y, Z \rangle$.

The First Lemma: A Reduction

- Now for every k, $X^k \pmod N$ is isomorphic to $\langle Y^k \pmod P, Z^k \pmod Q \rangle$.

  So the order of X is the least common multiple of the orders of Y and Z modulo P and Q, respectively.

  We complete the proof by showing that:

  - With probability at least $\frac{1}{2}$, the order of Y is even.
    That is, a number of the form $2^k c$ for $k \geq 1$ and c odd.
  - With probability at least $\frac{1}{2}$, the order of Z has the form $2^\ell d$, for d odd and $\ell \neq k$.

  These imply that the order of X is $r = 2^{\max\{k,\ell\}} \text{lcm}(c, d)$.

  So $X^{r/2}$ will be equal to 1 in at least one coordinate.

  Since $-1 \pmod N$ is isomorphic to the tuple $\langle -1, -1 \rangle$, this means that $X^{r/2} \not\equiv -1 \pmod P$.

The Statement in the Reduction

- We first show that Y has even order with probability at least $\frac{1}{2}$.

  The set of numbers in $\mathbb{Z}_P^*$ with odd order is a subgroup of $\mathbb{Z}_P^*$.

  Suppose Y, Y' have odd orders r, r', respectively.

  Then $(YY')^{rr'} \equiv 1 \pmod{P}$.

  So the order of YY' divides the odd number rr'.

  Yet -1 has even order.

  Thus, this is a proper subgroup of $\mathbb{Z}_P^*$.

  So it takes at most $\frac{1}{2}$ of $\mathbb{Z}_P^*$.

  There is a number ℓ_0, such that with probability exactly $\frac{1}{2}$, the order of a random $Z \in \mathbb{Z}_Q^*$ is a number of the form $2^\ell c$, for $\ell \leq \ell_0$.

  This implies that for every fixed k, the probability that the order has the form $2^k d$ is at most $\frac{1}{2}$.

The Statement in the Reduction (Cont'd)

- For every ℓ, define G_ℓ to be the subset of $\mathbb{Z}_Q^*$ whose order modulo Q is of the form $2^j c$, where $j \leq \ell$ and c is odd.

  Then, for every ℓ, G_ℓ is a subgroup of $G_{\ell+1}$.

  But modulo a prime P, the mapping $x \mapsto x^2 \pmod P$ is:
  - Two-to-one;
  - Maps $G_{\ell+1}$ into G_ℓ, $|G_\ell| \geq \frac{|G_{\ell+1}|}{2}$.

  It follows that, if we take ℓ_0 to be the largest, such that G_{ℓ_0} is a proper subgroup of $\mathbb{Z}_P^*$, then

$$|G_{\ell_0}| = \frac{|\mathbb{Z}_P^*|}{2}.$$

Rational Approximation of Real Numbers

- Suppose we are provided a real number in the form of a program that can compute its first t bits in poly(t) time.
- We want an approximation of the form $\frac{a}{b}$, with a prescribed upper bound on b.
- Continued fractions is a tool in number theory that is useful for this.
- A **continued fraction** is a number of the form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}},$$

where:

- a_0 is a non-negative integer;
- $a_1, a_2, \ldots$ are positive integers.

Rational Approximation of Real Numbers (Cont'd)

- Given a real number $\alpha > 0$, we can find its representation as an infinite fraction as follows:
  - Split α into the integer part $\lfloor \alpha \rfloor$ and fractional part $\alpha - \lfloor \alpha \rfloor$;
  - Find recursively the representation R of $\frac{1}{\alpha - \lfloor \alpha \rfloor}$;
  - Write
$$\alpha = \lfloor \alpha \rfloor + \frac{1}{R}.$$
- Suppose we continue this process for n steps.
- We get a rational number, denoted by
$$[a_0, a_1, \ldots, a_n].$$
- This number can be represented as $\frac{p_n}{q_n}$, with p_n, q_n coprime.

Rational Approximation of Real Numbers (Cont'd)

- We approximate α by $[a_0, a_1, \ldots, a_n] = \frac{p_n}{q_n}$, with p_n, q_n coprime.
- Using induction on n, we may show:
  - $p_0 = a_0, q_0 = 1$ and, for every $n > 1$,

$$\begin{aligned} p_n &= a_n p_{n-1} + p_{n-2}, \\ q_n &= a_n q_{n-1} + q_{n-2}. \end{aligned}$$

  - $\frac{p_n}{q_n} - \frac{p_{n-1}}{q_{n-1}} = \frac{(-1)^{n-1}}{q_n q_{n-1}}$.
- Furthermore, it is known that, for all n,

$$\left| \frac{p_n}{q_n} - \alpha \right| < \frac{1}{q_n q_{n+1}}.$$

- This implies that $\frac{p_n}{q_n}$ is the closest rational number to α with denominator at most q_n.

Rational Approximation of Real Numbers (Cont'd)

- Suppose α is extremely close to a rational number.
- Say, for some coprime a, b,

$$\left|\alpha - \frac{a}{b}\right| < \frac{1}{4b^4}.$$

- We show we can find a, b by iterating the continued fraction algorithm for polylog(b) steps.
- Let q_n be the first denominator such that $q_{n+1} \geq b$.
  - Suppose, first, $q_{n+1} > 2b^2$.
    Then

$$\left|\frac{p_n}{q_n} - \alpha\right| \leq \frac{1}{q_n q_{n+1}} < \frac{1}{2b^2 b} \leq \frac{1}{2b^2}.$$

    But there is at most one rational number of denominator $\leq b$ that is so close to α. Hence, $\frac{p_n}{q_n} = \frac{a}{b}$.

Rational Approximation of Real Numbers (Cont'd)

- Suppose, next, $q_{n+1} \leq 2b^2$.
  But $\frac{p_{n+1}}{q_{n+1}}$ is closer to α than $\frac{a}{b}$.
  So we have

$$\left|\frac{p_{n+1}}{q_{n+1}} - \alpha\right| \leq \left|\alpha - \frac{a}{b}\right| < \frac{1}{4b^4}.$$

  This means that $\frac{p_{n+1}}{q_{n+1}} = \frac{a}{b}$.

- We can verify that $q_n \geq 2^{n/2}$.
- This implies p_n and q_n can be computed in polylog(q_n) time.

Subsection 7

BQP and Classical Complexity Classes

BQP and PSPACE

- Quantum computers are at least not infinitely more powerful than classical algorithms.

Theorem

BQP $\subseteq$ PSPACE.

- We only provide a rough sketch of the reasoning.

  Consider a T-step quantum computation on an m qubit register.

  We need to devise a procedure $\textsc{Coeff}$ that, for every $i \in [T]$ and $x \in \{0,1\}^m$, outputs the x-th coefficient (up to some accuracy) of the register's state after the i-th step.

  The operation F_i of the i-th step reads and modifies at most 3 qubits.

  So we can compute $\textsc{Coeff}$ on inputs x, i using at most eight recursive calls to $\textsc{Coeff}$ on inputs $x', i-1$ for the at most eight strings x' that agree with x on these three qubits.

Analysis of the Recursion

- Note that we can reuse the space used by the recursive operations.

  Let $S(i)$ denote the space needed to compute $\textsc{Coeff}(x, i)$.

  Suppose ℓ is the number of bits used to store each coefficient.

  Then

  $$S(i) \leq S(i-1) + O(\ell).$$

  Suppose, e.g., that we want to compute the probability that, if measured after the final step, the first register qubit is equal to 1.

  We then compute the sum of $\textsc{Coeff}(x, T)$, for every $x \in \{0,1\}^n$.

  Again, by reusing the space of each computation this can be done using polynomial space.

BQP and BPP

- The main reason to believe that BQP $\neq$ BPP is the polynomial-time quantum algorithm for integer factorization.

  No similar algorithm is believed to exist for probabilistic computation.
- This is not as strong as the evidence for, say NP $\not\subseteq$ BPP.

  NP contains thousands of well-studied problems that have resisted efficient algorithms.
- Still, the factorization problem is one of the oldest and most well studied computational problems.

  The fact that we still know no efficient algorithm for it makes the conjecture that none exists appealing.
- Moreover, unlike other famous problems for which we eventually found efficient algorithms (e.g., linear programming and primality testing), we do not even have a heuristic algorithm that is conjectured to work (even without proof) or experimentally works on, say, numbers that are product of two random large primes.

BQP and NP

- It seems that quantum computers only offer a quadratic speedup (using Grover's search) on NP-complete problems.
- There are also oracle results showing that NP problems require exponential time on quantum computers.
- So most researchers believe that NP $\not\subset$ BQP.
- On the other hand, there is a problem in BQP (the Recursive Fourier Sampling or RFS problem) that is not known to be in the polynomial hierarchy, let alone in NP.
- Thus, it seems that BQP and NP may be incomparable classes.

Quantum Analogs of NP and AM

- The class NP was defined using the notion of a certificate that is checked by a deterministic polynomial time (classical) TM.
- Quantum computation includes probabilistic classical computation as a subcase.
- To relate the two classes, we look at a model in which the certificate is verified by a polynomial-time randomized algorithm, i.e., AM.
- Thus, the quantum analog of NP is denoted by QAM.
- One can define quantum interactive proofs, which generalize the definition of AM[k], which turn out to be surprisingly powerful.
  - Three-round quantum interactive proofs suffice to capture PSPACE.
  - If the same were true of classical interactive proofs, then PH would collapse.

Quantum Cook-Levin Theorem

- A "Quantum Cook-Levin Theorem", proven by Kitaev, shows that a quantum analog of 3SAT, called Q5SAT, is complete for QMA.
- Q5SAT:

  Given: m elementary quantum operations $H_1, H_2, \ldots, H_m$ on an n-bit quantum register.
  Each operation acts upon only 5 bits of the register (so is represented by a $2^5 \times 2^5$ matrix, which implicitly defines a $2^n \times 2^n$ matrix).
  Let H be the $2^n \times 2^n$ matrix $\sum_j H_j$.
  Promise: Suppose

$$0 \leq a \leq b \leq 1 \quad \text{and} \quad b - a \geq \frac{1}{n^c}, \text{ for } c \text{ a constant.}$$

  Then one of the following holds:

  - All eigenvalues of H are $\geq b$;
  - There is an eigenvalue of H that is $\leq a$.

  To determine: Which of the two cases in the promise holds.