Chapter 5

Applications

This is the final chapter discussing some known applications of our theory. We shall first examine the mathematical problem. Then we shall detail how the problem can solved in a quantum matter. Obviously, there are many more unsolved problems, including the conversion of conventional algorithms on a classical computer to quantum computation. I believe that this is a area full of treasures to be discovered or rediscovered.

- RSA encryption
- Period finding and Shor factorization
- Quantum Fourier transform.
- Quantum algorithm for solving algebraic equations
- Quantum algorithm for solving differential equations

5.1 RSA Encryption

- Simon's algorithm amounts to finding the unknown period a of a function on n-bit integers that is periodic under bitwise modulo-2 addition.
- A more difficult problem is to find the period r of a function f on the "true" integers that is periodic under ordinary addition, i.e., $f(x) = f(y)$ if and only if $x = y(\text{mod})r$.
- Any computer that can efficiently find periods would be an enormous threat to the security of both military and commercial communications.
- The RSA cryptosystem is a public key protocol widely used in industry and government to encrypt sensitive information.
  - ◇ The security of RSA rests on the assumption that it is difficult for computers to factor large numbers.
  - ◇ There is no known (classical) computer algorithm for finding the factors of an n-bit number in time that is polynomial in n.

Fermat Little Theorem

- Theorem: Let p be a prime number and a be any positive integer which is not a multiple of p. Then

$$a^{p-1} = 1 \ (\mathsf{mod}\ p). \tag{5.1}$$

  - ◇ Claim that $m^p - m = 0$ ($\mathsf{mod}\ p$) for any integer m.
    - ▷ True if $m = 1$.
    - ▷ Suppose that the claim is true for $m = k$. Observe that

$$(k+1)^p-(k+1) = \sum_{j=0}^{p} \binom{p}{j} k^{p-j}-(k+1) = k^p-k \ (\mathsf{mod}\ p)$$

      - ✓ The mathematical induction kicks in.
    - ▷ Take $m = a$. Then $a^p - a = a(a^{p-1} - 1) = 0$ ($\mathsf{mod}\ p$).
    - ▷ Since a is not divisible by p, $a^{p-1} - 1 = 0$ ($\mathsf{mod}\ p$).

- Let p and q be two distinct prime number and a be any positive integer not divisible by either p or q.
  - ◇ No power of a is divisible by either p or q.
  - ◇ By (5.1),

$$\begin{aligned}(a^{q-1})^{p-1} &= 1 \ (\mathsf{mod}\ p),\\ (a^{p-1})^{q-1} &= 1 \ (\mathsf{mod}\ q).\end{aligned}$$

  - ◇ The number $a^{(p-1)(q-1)} - 1$ is divisible by p, q, and pq.

$$a^{(p-1)(q-1)} = 1 \ (\mathsf{mod}\ pq). \tag{5.2}$$

    - ▷ This is the basis of the RSA encription.

Group $\mathbb{Z}/n\mathbb{Z}$

- Given a positive integer n, the set

$$\mathbb{Z}/n\mathbb{Z} := \{a | 1 \leq a < n, \gcd(a, n) = 1\} \tag{5.3}$$

  is called the multiplicative group of integers modulo n.

  - ◇ If $\gcd(a, n) = 1$ and $\gcd(b, n) = 1$, then $\gcd(ab, n) = 1$. So $\mathbb{Z}/n\mathbb{Z}$ is closed under multiplication.
  - ◇ If $\gcd(a, n) = 1$, then by the Bézout lemma there are integers x and y satisfying $ax + ny = 1$. Thus $ax = 1 \pmod{n}$. (See also the Euclidean long division.)

- Take $n = (p - 1)(q - 1)$ and $c \in \mathbb{Z}/n\mathbb{Z}$.

  - ◇ Let d be the multiplicative inverse of c.
  - ◇ Therefore, for some integer s,

$$cd = 1 + s(p - 1)(q - 1).$$

  - ◇ From (5.2), it holds that

$$a^{1+s(p-1)(q-1)} = a \pmod{pq}.$$

  - ◇ Take advantage of this simple relationship

$$b := a^c \pmod{pq} \Rightarrow b^d = a \pmod{pq}. \tag{5.4}$$

Communication between Alice and Bob

- Suppose Alice (A) wants to send an encoded message so that Bob (B) alone can read it, but Eve (E) always wants to eavesdrop.
- Bob:
  - ◇ Choose two large, say 200-digit, prime numbers p and q, and a number c which is coprime to $n = (p-1)(q-1)$.
  - ◇ Send Alice the product $N = pq$ and c. These are the *public keys*.
    - ▷ Even if Eve knows about N, it is difficult to figure out p and q.
  - ◇ Compute the inverse $d = c^{-1} \pmod{n}$, but keep it strictly for himself for use in decoding.
- Alice:
  - ◇ Encode a message (or a segment of a long message) by representing it as a number $a < N$.
  - ◇ Use the public keys, calculate $b = a^c$ and send it to Bob.
- Bob:
  - ◇ Upon receiving b, use d to calculate $a = b^d \pmod{pq}$.
- Eve:
  - ◇ Try hard to factorize $N = pq$.
  - ◇ Find the "period".

Decoding by Eve

- Assume that Eve has intercepted the encoded message b.
  - ◇ Since p and q are large prime numbers, with good chance that b is coprime to p and q, i.e., $b \in \mathbb{Z}/(pq)\mathbb{Z}$.
  - ◇ The group $\mathbb{Z}/(pq)\mathbb{Z}$ has exactly $pq - 1 - (p-1) - (q-1)$ elements. (Why?)
- Because $b = a^c$ and $a = b^d$, the cyclic subgroups generated by a and b, respectively, are identical.
  - ◇ If r is the number of elements in the cyclic subgroup, then $r|(p-1)(q-1)$.
    - ▷ $b^r = 1$.
    - ▷ By choice, $c \nmid (p-1)(q-1)$.
    - ▷ $\gcd(r, c) = 1 \Rightarrow c \in \mathbb{Z}/r\mathbb{Z}$.
  - ◇ Over the group $\mathbb{Z}/r\mathbb{Z}$, c has an inverse $\widetilde{d} = c^{-1} \pmod{r}$.

$$c\widetilde{d} = 1 \pmod{r}.$$

- If Eve can somehow find the value r, then

$$b^{\widetilde{d}} = (a^c)^{\widetilde{d}} = a^{1+kr} = a(a^r)^k = a \pmod{pq}.$$

- Given N and the intercepted b, define

$$f(x) := b^x \pmod{N}. \tag{5.5}$$

  Find r such that $f(x+r) = f(x)$.
  - ◇ Is this a special case of the Simon problem?

5.2 Quantum Fourier Transform

- The discrete Fourier transform (DFT) is the equivalent of the continuous Fourier transform for signals known only at finitely many instants separated by sample times.
- The quantum Fourier transform (QFT) assumes a similar form but has an entirely different meaning.
- The fast Fourier transform (FFT) exploits the structure of DFT and is critical in modern applications.
- The QFT is fast in its parallelism.

Quick Recap of Classical DFT

- Given samples $\{f_0, f_1, \ldots, f_{N-1}\}$ of a signal at fixed intervals in the time domain, the DFT in the frequency domain is defined by

$$\mathfrak{F}_j := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-\imath \frac{2\pi}{N} jk} f_k, \quad j = 0, 1, \ldots, N-1. \tag{5.6}$$

  - ◇ The relationship (5.6) can be expressed as

$$\mathfrak{F} = W\mathbf{f}. \tag{5.7}$$

    - ▷ The coefficient matrix W is unitary. (Prove this.)
  - ◇ Exploiting the cyclic nature in W leads to the fast Fourier transform (FFT) which is one of the most important algorithms with many applications.
    - ▷ At the cost of $O(N \log_2 N)$ computations.
    - ▷ Usually prefer that $N = 2^n$. Therefore, the overhead of FFT is $O(n2^n)$.
      - ✓ The significance of QFT is that the overhead is $O(n^2)$, i.e., exponentially faster than fast.
- It is easy to check the inverse relationship (IDFT):

$$f_j := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\imath \frac{2\pi}{N} jk} F_k, \quad j = 0, 1, \ldots, N-1. \tag{5.8}$$

- Some scholars/books weight the coefficient matrices differently for the convenience of trigonometric interpolation.

n-qubit QFT

- Let $N := 2^n$. The n-qubit *quantum Fourier transform (QFT)* is a unitary transformation $\mathbf{U}_{FT}$ defined by

$$\mathbf{U}_{QFT} |k\rangle_n := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-\imath \frac{2\pi}{N} jk} |j\rangle_n . \tag{5.9}$$

- Suppose that $g : \mathbb{Z}_N \to \mathbb{C}$. Consider its vector representation

$$\mathbf{g} = \sum_{k=0}^{N-1} g(k) |k\rangle_n .$$

Then

$$\begin{aligned} \mathbf{U}_{QFT}(\mathbf{g}) &= \sum_{k=0}^{N-1} g(k) \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-\imath \frac{2\pi}{N} jk} |j\rangle_n \\ &= \sum_{j=0}^{N-1} (\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} g(k) e^{-\imath \frac{2\pi}{N} jk}) |j\rangle_n \\ &= \sum_{j=0}^{N-1} \mathfrak{G}(j) |j\rangle_n . \end{aligned}$$

◇ The Fourier coefficients of g are defined as

$$\mathfrak{G}(j) := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} g(k) e^{-\imath \frac{2\pi}{N} jk}. \tag{5.10}$$

▷ In complete sync with (5.6).

Confusing Notation

- Conventions for the sign of the phase factor exponent vary. Some literature prefers to define the n-qubit QFT via

$$\mathbf{U}_{FT}^{\dagger} \left|j\right\rangle_n := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\imath \frac{2\pi}{N} jk} \left|k\right\rangle_n . \tag{5.11}$$

  - ◇ This is indeed a resemblance to the IDFT in (5.8).

- Though look alike, do not confused $\mathbf{U}_{FT}^{\dagger}$ with (3.12) where

$$\mathbf{H}^{\otimes n} \left|j\right\rangle_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\imath \pi j \cdot k} \left|k\right\rangle_n .$$

  - ◇ The product jk in (5.9) is the ordinary multiplication, but the product $j \cdot k$ is the bitwise inner product modulo 2.
  - ◇ $e^{\imath \pi j \cdot k} = (-1)^{j \cdot k}$ are ± 1, but $e^{\imath \frac{2\pi}{N} jk}$ has many phases.

- An important question is to show that an n-qubit QFT can be implemented out of 1-qubit and 2-qubit unitary gates and that the number of gates grows only quadratically in n.
  - ◇ This realization, at least in theory, will become clear at the end of of the next chapter on phase estimation.

Using QFT for Period Finding

- Consider as a demo the case $n = 3$ where we are looking for the period of $f : \{0,1\}^3 \to \{0,1\}^3$. (Simon Problem)
- Start with the initial state

$$|\Psi_0\rangle := \mathbf{H}^{\otimes 3} |0\rangle_3 = \frac{1}{\sqrt{8}} \sum_{j=0}^{7} |j\rangle_3 .$$

- Apply $\mathbf{U}_f$ to obtain

$$|\Psi_1\rangle := \mathbf{U}_f(\mathbf{H}^{\otimes 3} |0\rangle_3 |0\rangle_3) = \frac{1}{\sqrt{8}} \sum_{j=0}^{7} |j\rangle_3 |f(j)\rangle_3 .$$

  ◇ All output registers are equally weighted.

- Apply the QFT to obtain

$$\begin{aligned} |\Psi_2\rangle &:= \mathbf{U}_{QFT}(|\Psi_1\rangle) = \frac{1}{8} \sum_{j=0}^{7} \sum_{k=0}^{7} e^{-\imath \frac{2\pi}{8} jk} |k\rangle_3 |f(j)\rangle_3 \\ &= \frac{1}{8} \sum_{k=0}^{7} |k\rangle_3 (\sum_{j=0}^{7} e^{-\imath \frac{2\pi}{8} jk} |f(j)\rangle_3). \end{aligned}$$

- Suppose that the period is $p = 2$. Define the abbreviation

$$\begin{cases} a := f(0) = f(2) = f(4) = f(6), \\ b := f(1) = f(3) = f(5) = f(7). \end{cases}$$

  ◇ Check the second summation $s_k := \sum_{j=0}^{7} e^{-\imath \frac{2\pi}{8} jk} |f(j)\rangle_3$.

- Define $\omega_k := e^{-\imath \frac{2\pi}{8} k}$. Then

$$s_k = |a\rangle_3 (\omega_k^0 + \omega_k^2 + \omega_k^4 + \omega_k^6) + |b\rangle_3 (\omega_k^1 + \omega_k^3 + \omega_k^5 + \omega_k^7).$$

  - ◇ $s_0 = 4\,|a\rangle_3 + 4\,|b\rangle_3$.
  - ◇ $s_4 = 4\,|a\rangle_3 - 4\,|b\rangle_3$.
  - ◇ All other $s_k = 0$.

- In short, after the quantum Fourier transform, we see a redistribution of the weights, i.e.,

$$|\Psi_2\rangle = \frac{1}{2}(|0\rangle_3\,|a\rangle_3 + |0\rangle_3\,|b\rangle_3 + |4\rangle_3\,|a\rangle_3 - |4\rangle_3\,|b\rangle_3).$$

  If we take the measurement, input registers $|0\rangle$ and $|4\rangle$ will show up with equal probability which is a consequence of $p = 2$.

- The cancelation observed here is more extensively exploited in Shor's factorization algorithm.

General Algorithm for $N = 2^n$

- Prepare the uniform superposition of all basic $|x\rangle$.

$$|\Psi_0\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle \,.$$

- Apply the oracle $\mathbf{U}_f$ to obtain

$$|\Psi_1\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle \, |f(x)\rangle \,.$$

- Measure an output register and take record of the input register of the collapsed state.
- Apply QFT to the input register.
- Repeat enough times to create equations for estimating the period p.

Analysis I

- Denote the image set of f by $\mathcal{O}$.
  - ◇ $\mathcal{O}$ contains exactly p elements.
  - ◇ For each $c \in \mathcal{O}$, define the indicator function $f_c : \mathbb{Z}_N \to \{0, 1\}$ by

$$f_c(x) = \begin{cases} 1 & \text{if } f(x) = c, \\ 0 & \text{otherwise.} \end{cases}$$

- Rewrite $|\Psi_1\rangle$ as

$$|\Psi_1\rangle = \sum_{c \in \mathcal{O}} \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} (f_c(x) |x\rangle) |f(x)\rangle .$$

  - ◇ The probability of observing c is $\frac{1}{p}$ for every $c \in \mathcal{O}$. (Why?)
  - ◇ At the observation c, the system collapse to

$$\begin{aligned} |\Phi\rangle &:= \sqrt{\frac{p}{N}} \sum_{x \in \mathbb{Z}_N} (f_c(x) |x\rangle) |c\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} (\sqrt{p} f_c(x) |x\rangle) |c\rangle . \end{aligned}$$

- Which output register c is to be measured?

Periodic Spike Function

- Given $g : \mathbb{Z}_N \to \mathbb{C}$, suppose $\widetilde{g}(k) = g(k+T)$. Then

$$\begin{aligned}
\mathbf{U}_{QFT}(\widetilde{\mathbf{g}}) &= \sum_{k=0}^{N-1} \widetilde{g}(k) \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-\imath \frac{2\pi}{N} jk} |j\rangle_n \\
&= \sum_{j=0}^{N-1} \Big(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} g(k+T) e^{-\imath \frac{2\pi}{N} jk} \Big) |j\rangle \\
&= \sum_{j=0}^{N-1} \Big(\frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} g(y) e^{-\imath \frac{2\pi}{N} j(y-T)} \Big) |j\rangle \\
&= \sum_{j=0}^{N-1} e^{\imath \frac{2\pi}{N} jT} \mathfrak{G}(j) |j\rangle .
\end{aligned}$$

  - ◇ The Fourier coefficients of g and $\widetilde{g}$ have the same magnitude.

- Suffice to concentrate on one output register c.

- Without loss of generality, we take $c = f(0)$.

  - ◇ $f_c(x) = 1$ only at the set $\mathcal{H} = \{0, p, 2p, 3p, \ldots\}$.
  - ◇ $\mathcal{H}$ has exactly $\frac{N}{p}$ elements.

Analysis II

- Define $\mathcal{D} := \{0, \frac{N}{p}, \frac{2N}{p}, \ldots\} \subset \mathbb{Z}_N$.
  - ◇ $\mathcal{D}$ has exactly p elements. (Why?)
- Apply the QFT to the input register of $|\Phi\rangle$.
  - ◇ The input register at observation c is

$$\mathbf{g} := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} \sqrt{p} f_c(x) \, |x\rangle \, .$$

  - ◇ Look for

$$\mathbf{U}_{QFT}(\mathbf{g}) = \sum_{j=0}^{N-1} \mathfrak{G}(j) \, |j\rangle \, .$$

  - ◇ By (5.10), the Fourier coefficients are given by

$$\mathfrak{G}(j) := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \sqrt{p} f_c(k) e^{-\imath \frac{2\pi}{N} jk}.$$

- Two cases:
  - ◇ If $j \in \mathcal{D}$,

$$\mathfrak{G}(j) = \frac{1}{\sqrt{N}} \sum_{k \in \mathcal{H}} \frac{1}{\sqrt{N}} \sqrt{p} e^{-\imath \frac{2\pi}{N} jk} = \frac{\sqrt{p}}{N} \frac{N}{p} = \frac{1}{\sqrt{p}}.$$

  - ◇ If $j \notin \mathcal{D}$,

$$\mathfrak{G}(j) = 0.$$

    - ▷ Note that $\sum_{j \in \mathcal{D}} |\mathfrak{G}(j)|^2 = 1$.
    - ▷ No more probability for $j \notin \mathcal{D}$.
- So what to do with these results?

Analysis III

- We measure each $j \in \mathcal{D}$ with probability $\frac{1}{p}$ and others with probability 0.
- $j \in \mathcal{D}$ if and only if $jp = 0 \ (\mathsf{mod}\ N)$.
  - ◇ That is, we sample a uniformly random j, which are multiples of the ratio $\frac{N}{p}$, such that $jp = 0 \ (\mathsf{mod}\ N)$.
- This is similar to what has happened in Simon algorithm!
  - ◇ In the Simon algorithm, we sampled a uniformly random such that $z_i \perp p$ under the dot product of two n-qubit modulo 2.
  - ◇ Here, we consider multiplication modulo N.
- How to find the ratio $m = \frac{N}{p}$?
  - ◇ The algorithm samples a random integer multiple of m.
  - ◇ Suppose we have two random samples, which we can write am and bm.
  - ◇ Note that $\gcd(am; bm) = \gcd(a; b)m$, if $\gcd(a; b) = 1$, then m is found.
- If a and b are uniformly and independently sampled integers from $\mathbb{Z}_N$, what is the probability that $\gcd(a; b) = 1$?

5.3 Shor Factorization

- Shors factorization algorithm is used to factor numbers into their components (which can potentially be prime).
- It is one of the most significant examples in which a quantum computer demonstrates enormous power surpassing its classical counterpart.
  - ◇ The algorithm does the factorization in roughly $O(n^3)$ quantum operations.
  - ◇ In contrast, the best known classical algorithms are exponential.
- Since the basis of most modern cryptography system is relying on the impossibility of exponential cost of factorization, being able to factor in polynomial time on a quantum computer has attracted significant interest.

Basic Idea

- Given positive integers x and N, $x < N$, x is coprime to N, the order of $x \pmod N$ is the least positive integer r such that $x^r \pmod N = 1$.
  - ◇ Since $x \pmod N$ can only take a finite number of different values, x must have a finite order, denoted by $ord_N(x)$.
  - ◇ The order $ord_N(x)$ divides the order of $\mathbb{Z}/N\mathbb{Z}$.
- A basic idea of Shor's algorithm:
  - ◇ Suppose p and q are prime number and $N = pq$.
  - ◇ Choose some random x which is co-prime of N
  - ◇ Use quantum parallelism to compute x^r for all r simultaneously. (This is only an easy brute force way. It could be done in a better way.)
  - ◇ Interfere all of the x^r's to obtain knowledge about its period.
  - ◇ Use this period to find the factor p or q of N.

Order Finding ⇒ Factorization

- A quantum computation is needed to find the order r, which will be discussed later.
- If the order r of x is odd, choose another random x.
- Suppose an even order r is found. Write

$$(x^{r/2} - 1)(x^{r/2} + 1) = x^r - 1 = 0 \ (\mathsf{mod}\ N).$$

  - ◇ If $x^{r/2} + 1 = 0 \ (\mathsf{mod}\ N)$, then $\gcd(x^{r/2} - 1, N) = 1$; choose a different x.
  - ◇ If $x^{r/2} + 1 \neq 0 \ (\mathsf{mod}\ N)$, then

$$d := \gcd(x^{r/2} - 1, N)$$

    must be either p or q.

    - ▷ $x^{r/2} - 1$ cannot be a multiple of N; otherwise, $x^{r/2} = 1 \ (\mathsf{mod}\ N)$, contradicting $ord_N(x) = r$.
    - ▷ $x^{r/2} - 1$ contains either p or q.

- No need to assume $N = pq$.
  - ◇ The number d gives us a nontrivial factor of N.
  - ◇ Factorize d and $\frac{N}{d}$ recursively and obtain all prime factors.
  - ◇ We can efficiently test primality. (How?)
- Suppose N has at least two distinct prime factors. If $x \in \mathbb{Z}/N\mathbb{Z}$ is selected randomly, then the probability that $ord_N(x)$ is even and is at least $\frac{1}{2}$.

Order Finding

- Choose n large enough such that $N^2 < Q := 2^n < 2N^2$. Define $f : \mathbb{Z}_Q \to \mathbb{Z}_N$ via
$$f(z) := x^z \pmod{N}.$$
  - ◇ Finding $ord_N(x)$ is almost equivalent to finding the period of f.
$$f(z+r) = f(z).$$
  - ◇ However, r does not necessarily divide Q. This is the main difference between order-finding and period finding.
- For a measurement of the output register c,
  - ◇ It appears only D times where D is either $\lfloor \frac{Q}{r} \rfloor$ or $\lceil \frac{Q}{r} \rceil$.
  - ◇ The system collapse to
$$|\Phi\rangle \ := \ \frac{1}{\sqrt{D}} \sum_{z \in \mathbb{Z}_Q} \left(f_c(z) \, |z\rangle \right) |c\rangle \, .$$
  - ◇ Replace the set $\mathcal{D}$ used in Analysis II by $\mathcal{D} := \{0, D, 2D, \ldots\}$.
  - ◇ Apply the QFT to $|\Phi\rangle$ to obtain
$$(\mathbf{U}_{QFT} \otimes I)(|\Phi\rangle) = \sum_{j=0}^{Q-1} \mathfrak{G}(j) \, |j\rangle \, |c\rangle \, .$$
    with the Fourier coefficients
$$\mathfrak{G}(j) \ := \ \frac{1}{\sqrt{QD}} \sum_{k=0}^{Q-1} f_c(k) e^{-\imath \frac{2\pi}{Q} jk} = \frac{1}{\sqrt{QD}} \sum_{t=0}^{D-1} e^{-\imath \frac{2\pi}{Q} jtr}$$
    - ▷ This is not as simple as that for the period-finding.

Continued Fraction

(This is to be completed later.)

5.4 Solving Linear Algebra Problems

- Eigenvalue Problem
- Quantum Phase Estimation
- System of Linear Equations

Eigenvalue Problem

- Given a unitary operator $\mathbf{U}$ that operates on m qubits with an eigenvector $|\Psi\rangle$ such that

$$\mathbf{U}\,|\Psi\rangle = e^{\imath 2\pi\omega}\,|\Psi\rangle\,,$$

estimate ω.

  ◇ With high probability;

  ◇ Within additive error ϵ;

  ◇ Using $O(\log \frac{1}{\epsilon})$ qubits;

  ◇ Using $O(\frac{1}{\epsilon})$ controlled-$\mathbf{U}$ operations.

- The following process $\mathbf{U}_{phase}$ that does the transformation

$$|0\rangle^n\,|\Psi\rangle \rightarrow |\widetilde{\omega}\rangle\,|\Psi\rangle$$

is called a quantum phase algorithm.

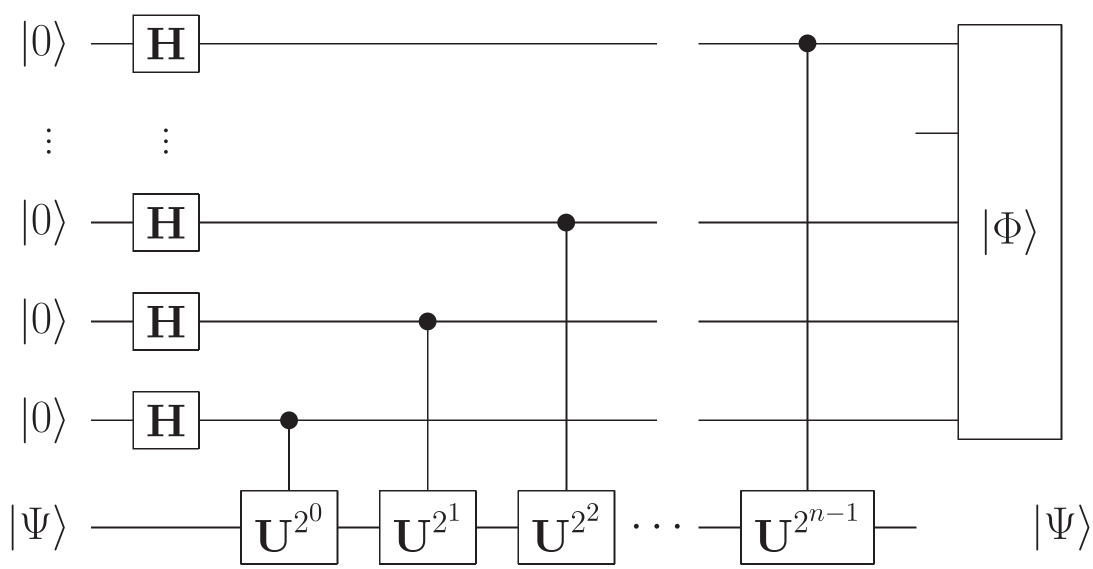
  ◇ $\widetilde{\omega}$ is an estimate of ω with known error.

Setup

- Create superposition:

$$|\Phi\rangle = (\mathbf{H}^{\otimes n} \otimes I)(|0\rangle^{\otimes n} |\Psi\rangle) = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n} |\Psi\rangle .$$

- Apply controlled-$\mathbf{U}^{2^j}$ gates, $j = 0, 1, \ldots, n-1$, sequentially according to the circuit:



◇ First, apply the controlled-$\mathbf{U}^{2^0}$:

$$|\Psi\rangle \overset{c\mathbf{U}^{2^0}}{\Rightarrow} \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes(n-1)} \otimes (|0\rangle|\Psi\rangle + |1\rangle \mathbf{U}^{2^0}|\Psi\rangle$$
$$= (\frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes(n-1)} \otimes (|0\rangle + e^{\imath 2\pi 2^0 \omega}|1\rangle)) \otimes |\Psi\rangle\,.$$

◇ Followed by the controlled-$\mathbf{U}^{2^1}$:

$$\overset{c\mathbf{U}^{2^1}}{\Rightarrow} \frac{1}{\sqrt{2^n}}(|0\rangle+|1\rangle)^{\otimes(n-3)} \otimes (|0\rangle + e^{\imath 2\pi 2^1 \omega}|1\rangle) \otimes (|0\rangle + e^{\imath 2\pi 2^0 \omega}|1\rangle) \otimes |\Psi\rangle\,.$$

◇ At the end, receive the identity for the input register:

$$\bigotimes_{j=1}^{n} \frac{1}{\sqrt{2}}(|0\rangle + e^{\imath 2\pi\omega 2^{n-j}}|1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\imath 2\pi\omega k}|k\rangle_n\,. \quad (5.12)$$

▷ Prove the identity!

▷ Same problem, but without the knowledge of $|\Psi\rangle$.

▷ Estimate ω.

Quantum Phase Estimation

- Problem: Given a state

$$|\Phi\rangle := \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\imath 2\pi\omega k} |k\rangle_n \,, \tag{5.13}$$

obtain a good estimate of the phase parameter ω.

- Some preliminary facts:
  - ◇ Because $0 < \omega < 1$, can express

$$\omega = (.d_1 d_2 d_3 \ldots)_2 = \frac{d_1}{2} + \frac{d_2}{2^2} + \ldots$$

  - ◇ Then

$$\begin{aligned} e^{\imath 2\pi(2^k \omega)} &= e^{\imath 2\pi(d_1 d_2 d_3 \ldots d_k . d_{k+1} d_{k+2} \ldots)_2} \\ &= e^{\imath 2\pi(0.d_{k+1} d_{k+2} \ldots)_2} \end{aligned}$$

  - ◇ Therefore, can rewrite $|\Phi\rangle$ as

$$|\Phi\rangle = \bigotimes_{j=1}^{n} \frac{1}{\sqrt{2}}(|0\rangle + e^{\imath 2\pi(.d_{n-j+1} d_{n-j+2} \ldots)_2} |1\rangle).$$

An Example

- Consider the case $n = 2$ and $\omega = (.d_1 d_2)_2$.
- By (5.12),

$$|\Phi\rangle = (\frac{|0\rangle + e^{\imath 2\pi (.d_2)_2} |1\rangle}{\sqrt{2}}) \otimes (\frac{|0\rangle + e^{\imath 2\pi (.d_1 d_2)_2} |1\rangle}{\sqrt{2}}).$$

- Applying $\mathbf{H}$ to the first qubit, we see that

$$\mathbf{H}(\frac{|0\rangle + e^{\imath 2\pi (.d_2)_2} |1\rangle}{\sqrt{2}}) = \frac{|0\rangle + |1\rangle}{2} + \frac{|0\rangle - |1\rangle}{2}(-1)^{d_2} = |d_2\rangle .$$

- To determine d_1,
  - ◇ If $d_2 = 0$, obtain d_1 by applying $\mathbf{H}$ to the second qubit.
  - ◇ If $d_2 = 1$,
    - ▷ Define the 1-qubit phase rotation operator $\mathcal{R}_2$

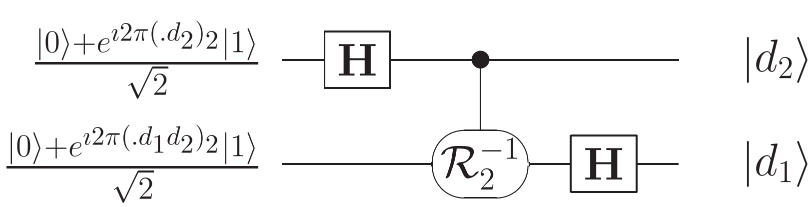
$$\mathcal{R}_2 := \begin{bmatrix} 1 & 0 \\ 0 & e^{\imath \frac{2\pi}{2^2}} \end{bmatrix}. \quad (5.14)$$

.

    - ▷ Observe

$$\mathcal{R}_2^{-1}(\frac{|0\rangle + e^{\imath 2\pi (.d_1 1)_2} |1\rangle}{\sqrt{2}}) = \frac{|0\rangle + e^{\imath 2\pi (.d_1)_2} |1\rangle}{\sqrt{2}}.$$

  - ◇ This is a controlled-$\mathcal{R}_2^{-1}$ gate.
- The phase can be recovered exactly from the circuit:



Generalization

- Consider the case $n = 3$ and $\omega = (.d_1 d_2 d_3)_2$.
- By (5.12),

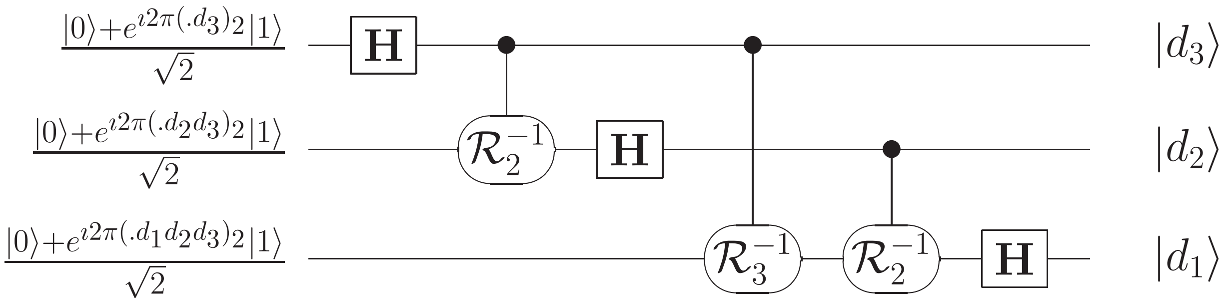
$$|\Psi\rangle = (\frac{|0\rangle + e^{\imath 2\pi(.d_3)_2}|1\rangle}{\sqrt{2}}) \otimes (\frac{|0\rangle + e^{\imath 2\pi(.d_2 d_3)_2}|1\rangle}{\sqrt{2}}) \otimes (\frac{|0\rangle + e^{\imath 2\pi(.d_1 d_2 d_3)_2}|1\rangle}{\sqrt{2}}).$$

- Define the phase rotation operator $\mathcal{R}_k$ by

$$\mathcal{R}_k := \begin{bmatrix} 1 & 0 \\ 0 & e^{\imath \frac{2\pi}{2^k}} \end{bmatrix}. \tag{5.15}$$

.

- Build the circuit as



- Note that the above operations does the inverse of QFT.

$$\frac{1}{\sqrt{2^3}} \sum_{k=0}^{2^3-1} e^{\imath 2\pi(.d_1 d_2 d_3)_2 k} |k\rangle_n \to |d_1 d_2 d_3\rangle .$$

Inverse QFT

- Recall the linear relationships $\mathbf{F} = W\mathbf{f}$ in the DFT and $\mathbf{f} = W^*\mathbf{F}$ in the IDFT.
  - ◇ Is there a similar inverse relationship to the QFT (5.9)?
- In the above construction, ω is of the form $\frac{x}{2^n}$ where x is an n-qubit, i.e.,

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\imath \frac{2\pi}{2^n} xk} |k\rangle_n \to |x\rangle . \tag{5.16}$$

  - ◇ By applying the circuit in reverse order whereas every gate is replaced by its inverse, we have a circuit for the QFT.
  - ◇ Show that the transformation (5.16) can be expressed as the map

$$\mathbf{U}_{QFT} |j\rangle_n := \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{-\imath \frac{2\pi}{2^n} jk} |k\rangle_n . \tag{5.17}$$

- What if ω is not of the form $\frac{x}{2^n}$, say, what if ω is irrational?

Quantum Phase Algorithm

- Applying the QFT to the right side of (5.12) yields

$$\begin{aligned} \mathbf{U}_{QTF}\left|\Phi\right\rangle &= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{\imath 2\pi\omega k}\mathbf{U}_{QFT}\left|k\right\rangle_n \\ &= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{\imath 2\pi\omega k}\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1} e^{-\imath\frac{2\pi}{2^n}jk}\left|j\right\rangle_n \\ &= \frac{1}{2^n}\sum_{j=0}^{2^n-1}\Big(\sum_{k=0}^{2^n-1} e^{\imath 2\pi\frac{k}{2^n}(2^n\omega-j)}\Big)\left|j\right\rangle_n \end{aligned}$$

- Approximate the value of $\omega \in [0,1]$ as follows:
  - ◇ Round $2^n\omega$ to the nearest integer, say,

$$2^n\omega = a + 2^n\delta,$$

    with $0 \le \delta \le \frac{1}{2^{n+1}}$.
  - ◇ The final state appears in the form

$$\frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{-\imath 2\pi\frac{k}{2^n}(x-a)}e^{\imath 2\pi\delta k}\left|x\right\rangle\otimes\left|\Psi\right\rangle.$$

  - ◇ Make a measurement. The probability of yielding $x = a$ is given by

$$Pr(x=a) = \frac{1}{2^{2n}}\left|\sum_{k=0}^{2^n-1} e^{\imath 2\pi\delta k}\right|^2 = \begin{cases} 1, & \text{if } \delta = 0, \\ \frac{1}{2^{2n}}\left|\frac{1-e^{\imath 2\pi 2^n\delta}}{1-e^{\imath 2\pi\delta}}\right|^2, & \text{if } \delta \neq 0. \end{cases}$$

    - ▷ Can prove that for $\delta \neq 0$, $Pr(a) \ge \frac{4}{\pi^2} \approx 0.40$.

System of Linear Equations

- Solving linear systems is fundamental in virtually all areas of science and engineering. A quantum algorithm for linear systems of equations has the potential for widespread applicability.
- Thus far, the quantum algorithm can only work for Hermitian matrices, preferably large and sparse.
- Can provide only a scalar measurement on the solution vector. Not the values of the solution vector itself.
  - ◇ What to expect from a quantum machine when trying to determine the intersection of two lines? Can quantum computing determine the geometry?

Quantum Linear System Problem

- Given an $N \times N$ Hermitian matrix A, a unit vector $\mathbf{b}$, and an operator M, find the value

$$\langle \mathbf{x} | M | \mathbf{x} \rangle$$

where $\mathbf{x}$ is a solution to the system

$$A |\mathbf{x}\rangle = |\mathbf{b}\rangle .$$

- It has been declared that the quantum algorithm has an exponential speedup, $O(\log_2 N)$, as opposed to $O(N^3)$ of the classical Gaussian elimination method. However, we must clarify some assumptions:
  - ◇ To merely read in the entries of the vector $|\mathbf{b}\rangle$ will cost $O(N)$. So $|\mathbf{b}\rangle$ is assumed to be already prepared by some other means.
  - ◇ As a state, $|\mathbf{b}\rangle$ must be normalized.
  - ◇ If we have a means to measure one component of $|\mathbf{x}\rangle$, the system collapses. If we want to measure all components, we must repeat the algorithm N times. This is exponentially more than $O(\log_2 N)$.
  - ◇ The algorithm is suitable only for the class of quantum linear system problems (QLSP).

Basic Steps

- If A is Hermitian, then $U := e^{\imath At}$ is unitary.
  - ◇ The eigenvalues λ_j of A must be real.
  - ◇ The eigenvalues of A are precisely the phases of U.
  - ◇ A and U have the same set of eigenvectors $|\Psi_j\rangle$.
- Apply the phase estimation gate $\mathbf{U}_{phase}$ to Ψ_j, we have an estimate $\widetilde{\lambda}_j$ of λ_j.

$$|0\rangle^n |\Psi_j\rangle \rightarrow \left|\widetilde{\lambda}_j\right\rangle |\Psi_j\rangle$$

  without knowing $|\Psi_j\rangle$.
- Expand $|\mathbf{b}\rangle$ in terms of the basis of eigenvectors,

$$|\mathbf{b}\rangle = \sum_{j=1}^{N} \beta_j |\Psi_j\rangle .$$

  - ◇ Via $\mathbf{U}_{phase}$, we obtain the transformation

$$|\mathbf{b}\rangle \rightarrow \sum_{j=1}^{N} \beta_j \left|\widetilde{\lambda}_j\right\rangle |\Psi_j\rangle .$$

- Add a third register (ancilla) to make a controlled rotation $e^{\imath\theta\mathbf{Y}}$ conditioned upon $\left|\widetilde{\lambda}_j\right\rangle$ to obtain

$$\left|\widetilde{\lambda}_j\right\rangle |\Psi_j\rangle \rightarrow \left|\widetilde{\lambda}_j\right\rangle |\Psi_j\rangle \left(\sqrt{1-\left(\frac{\gamma_j}{\widetilde{\lambda}_j}\right)^2}\,|0\rangle + \frac{\gamma_j}{\widetilde{\lambda}_j}|1\rangle\right).$$

  - ◇ The control is per j and, hence, this is parallelism.
  - ◇ γ_j is for normalization.

- Together, the state is now

$$\sum_{j=1}^{N} \beta_j \left|\widetilde{\lambda}_j\right\rangle |\Psi_j\rangle \, (\sqrt{1-(\frac{\gamma_j}{\widetilde{\lambda}_j})^2}\,|0\rangle + \frac{\gamma_j}{\widetilde{\lambda}_j}\,|1\rangle).$$

- Undo (reverse) the phase estimation to obtain

$$|0\rangle^{\otimes n} \otimes \sum_{j=1}^{N} \beta_j \, |\Psi_j\rangle \, (\sqrt{1-(\frac{\gamma_j}{\widetilde{\lambda}_j})^2}\,|0\rangle + \frac{\gamma_j}{\widetilde{\lambda}_j}\,|1\rangle).$$

- Employ an amplifying operator to enlarge the magnitude of $|1\rangle$. The second term is proportional to

$$A^{-1}\,|\mathbf{b}\rangle = \sum_{j=1}^{N} \beta_j \frac{1}{\lambda_j}\,|\Psi_j\rangle \, .$$

  ◇ The whole process is nothing but an analogue of the linear algebra.

    ▷ If $A = Q\Lambda Q^*$, then

$$A^{-1} = Q\Lambda^{-1}Q^*.$$

    ▷ Therefore,

$$\mathbf{x} = Q\Lambda^{-1} \underbrace{Q^*\mathbf{b}}_{\beta_1,\ldots,\beta_n} \, .$$

- Note that only a quantum description of the solution vector is output from HHL.

  ◇ For applications that need a full classical description of $\mathbf{x}$, this may not be satisfactory.

5.5 Differential Equations

- Classical ODE Methods
- Quantum Formulations

Classical ODE Methods

- Differential equations are ubiquitous in science and engineering. Solving differential equations with high precision is of paramount importance.
- Both the theory and techniques of numerical ODE methods have reached the state-of-the-art in digital computation.
  - ◇ Can be programmed to automatically adapt optimal step sizes and orders along the integration.
  - ◇ Efficient for fast integration and effective for meeting error tolerance.
- Two main basic notions:
  - ◇ Nonlinear one-step methods.
    - ▷ Self-sufficient.
    - ▷ Can be used for help generate starting values.
    - ▷ More expensive.
  - ◇ Linear multi-step methods.
    - ▷ Require starting values.
    - ▷ Much cheaper with higher order precision.

Euler Method

- The simplest numerical method,
  - ◇ If $y(x)$ is a solution, then

$$y(x_n + h) = y(x_n) + hy^{'}(x_n) + O(h^2)$$

    is the Taylor series expansion of $y(x_n + h)$ near x_n.
  - ◇ Suppose the *accepted* solution at x_n is given $y(x_n) \approx y_n$, then the truncated Taylor series suggests a scheme:

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{5.18}$$

    should be a reasonable approximation.
- Questions always asked in numerical ODE:
  - ◇ What is the magnitude of the *global error*

$$e_n := y_n - y(x_n)$$

    at the n-th step?
  - ◇ (Stability) How does the error propagate?
  - ◇ (Precision) How does the step size h affect the accuracy?
  - ◇ How to control the step size and error growth to get the best possible accuracy?

An Improved Idea

- Consider a 2-stage scheme:
  - ◇ Take one half-step Euler shooting:
  $$y_{n+\frac{1}{2}} := y_n + \frac{h}{2} f(x_n, y_n).$$
  - ◇ Use the midpoint, instead of the endpoint, to estimate the slop:
  $$f_{n+\frac{1}{2}} := f\left(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}\right).$$
  - ◇ Take one full Euler shooting:
  $$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}, \mathbf{y_n} + \frac{\mathbf{h}}{\mathbf{2}}\mathbf{f}(\mathbf{x_n}, \mathbf{y_n})\right).$$
- Note that there are two function evaluations involved.
  - ◇ This is a smart way to implement the Taylor's series expansion.
  - ◇ Can match up the terms up to $O(h^3)$.
  - ◇ The method has one-order better precision than the Euler method.

Runge-Kutta Methods

- A general R-stage Runge-Kutta method is defined by the one-step method:

$$y_{n+1} = y_n + h\phi(x_n, y_n, h) \tag{5.19}$$

where

$$\phi(x_n, y_n, h) = \sum_{r=1}^{R} c_r k_r \tag{5.20}$$

$$\sum_{r=1}^{R} c_r = 1$$

$$k_r = f(x_n + a_r h, y_n + h\sum_{s=1}^{R} b_{rs} k_s) \tag{5.21}$$

$$\sum_{s=1}^{R} b_{rs} = a_r.$$

- It is convenient to characterize the scheme in the *Butcher array*:

$$\begin{array}{c|cccc} a_1 & b_{11} & b_{12} & \dots & b_{1R} \\ a_2 & b_{21} & b_{22} & \dots & b_{2R} \\ \vdots & \vdots & & & \vdots \\ a_R & b_{R1} & b_{R2} & \dots & b_{RR} \\ \hline & c_1 & c_2 & \dots & c_R \end{array}$$

- Lots of research has been done in the past few decades to find out the *best* combinations of these coefficients.

Some Popular RK Schemes

- Two fourth-order 4-stage explicit Runge-Kutta methods

$$\begin{array}{c|cccc} 0 & 0 & & & \\ 1/2 & 1/2 & 0 & & \\ 1/2 & 0 & 1/2 & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array}$$

$$\begin{array}{c|cccc} 0 & 0 & & & \\ 1/3 & 1/3 & 0 & & \\ 2/3 & -1/3 & 1 & 0 & \\ 1 & 1 & -1 & 1 & 0 \\ \hline & 1/8 & 3/8 & 3/8 & 1/8 \end{array}$$

- The unique 2-stage implicit Runge-Kutta method of order 4:

$$\begin{array}{c|cc} 1/2+\sqrt{3}/6 & 1/4 & 1/4+\sqrt{3}/6 \\ 1/2-\sqrt{3}/6 & 1/4-\sqrt{3}/6 & 1/4 \\ \hline & 1/2 & 1/2 \end{array}$$

- A 3-stage semi-explicit Runge-kutta method or order 4:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ 1 & 0 & 1 & 0 \\ \hline & 1/6 & 4/6 & 1/6 \end{array}$$

Linear Multi-step Methods

- A *linear* $(p+1)$-step method of step size h is a numerical scheme of the form

$$y_{n+1} = \sum_{i=0}^{p} a_i y_{n-i} + h \sum_{i=-1}^{p} b_i f_{n-i} \tag{5.22}$$

  where $x_k = x_0 + kh$, $f_{n-i} := f(x_{n-i}, y_{n-i})$, and $a_p^2 + b_p^2 \neq 0$.

  ◇ Note that the information used involves past values the approximate solution y_i and its first order derivative f_i.

  ◇ If $b_{-1} = 0$, then the method is said to be explicit; otherwise, it is implicit.

    ▷ In order to obtain y_{n+1} from an implicit method, usually it is necessary to solve a nonlinear equation.

    ▷ Implicit methods are more expensive.

    ▷ Implicit methods has better stability.

- The Adams family:

$$y_{n+1} = y_n + \sum_{i=-1}^{p} \beta_{pi} f_{n-i} \tag{5.23}$$

  ◇ Obtained from the Fundamental Theorem of Calculus

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(x, y(x))dx.$$

  ◇ The unknown $f(x, y(x))$ is approximated by polynomial interpolation.

Some Popular Multi-step Schemes

- Some Adams-Bashforth methods:

	0	1	2	3	4
β_{0i}	1				
$2\beta_{1i}$	3	-1			
$12\beta_{2i}$	23	-16	5		
$24\beta_{3i}$	55	-59	37	-9	
$720\beta_{4i}$	1901	-2774	2616	-1274	251

- Some Adams-Moulton methods:

	-1	0	1	2	3
β_{0i}	1				
$2\beta_{1i}$	1	1			
$12\beta_{2i}$	5	8	-1		
$24\beta_{3i}$	9	19	-5	1	
$720\beta_{4i}$	251	646	-264	106	-19

Quantum Formulation

- A quantum algorithm for general nonlinear differential equations might be too ambitious.
  - ◇ The complexity of the simulation scaled exponentially in the number of time-steps.
  - ◇ The quantum nonlinear Schrödinger equation is nonlinear in the field operators, but it is still linear in the quantum state.
  - ◇ Most operators act linearly on quantum superpositions.
- A more natural application for quantum computers is linear differential equations:

$$\frac{d\mathbf{y}}{dt} = A(t)\mathbf{y} + \mathbf{b}(t); \quad \mathbf{y}(t_0) = \mathbf{y}_0 \in \mathbb{C}^N. \tag{5.24}$$

- Do we really have to solve an ODE by quantum algorithms?
  - ◇ The complexity of solving the differential equation in the classical algorithm must be at least linear in N.
  - ◇ One goal of the quantum algorithm is to solve the differential equation in time $O(\text{poly}(\log N))$. (Is this really important?)
  - ◇ Another goal is to provide efficient scaling in the evolution in time T.
  - ◇ Can the numerical analysis community accept quantum algorithms? (In terms of precision and stability)

Feynman Clock

- If a quantum system moves stepwise forward and then backward in time in equal increments, it would necessarily return to its original state.
- Traditional algorithms utilize parallelization in space.
  - ◇ A supercomputer comprising many processors spatially distributes and advances the problem in single temporal increments.
- On a quantum machine, think about the possibility of setting up a calculation that is parallel in time.
  - ◇ Different points in time has to be *simultaneously* calculated on many processors.

Key Idea

- Assume $t_j = t_0 + jh$, $\mathbf{y}_k \approx \mathbf{y}(t_j)$, and a total of N_t steps are taken over $[t_0, T]$ (so $N_t = \frac{T-t_0}{N_t}$).
- Wish to obtain the final state in the form
$$|\psi\rangle = \sum_{j=0}^{N_t} |t_j\rangle \, |\mathbf{y}_j\rangle \, . \tag{5.25}$$
  - ◇ Temporarily ignore the normalization.
  - ◇ By measuring the time register, we get the approximation $\mathbf{y}_j$.
  - ◇ The probability of obtaining the final time is small $\frac{1}{N_t+1}$. (How to boost the probability?)
- Try the Euler's method:
$$\mathbf{y}_{n+1} = \mathbf{y}_n + h(A(t_n)\mathbf{y}_n + \mathbf{b}_n).$$
- Code the solution at $\mathbf{y}_2$ via
$$\begin{bmatrix} I & 0 & 0 & 0 & 0 \\ -(I + A(t_0)h) & I & 0 & 0 & 0 \\ 0 & -(I + A(t_1)h) & I & 0 & 0 \\ 0 & 0 & -I & I & 0 \\ 0 & 0 & 0 & -I & I \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \mathbf{y}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{b}(\mathbf{t_0})h \\ \mathbf{b}(\mathbf{t_1})h \\ 0 \\ 0 \end{bmatrix} .$$
  - ◇ Note that $\mathbf{y}_3 = \mathbf{y}_4 = \mathbf{y}_2$ artificially. Therefore, the probability of getting $\mathbf{y}_2$ is boosted from $\frac{1}{3}$ to $\frac{3}{5}$.
  - ◇ This linear system is to be solved via the HHL algorithm.

Criticisms

- Is this a good idea? (All because of the uncertainty in quantum computing.)
- To achieve high precision, N_t has to be extremely large.
- To know the intermediate solutions, lots of measurements need be made.
- How to deal with stability? The error will grow.