

NUMERICAL ANALYSIS ON QUANTUM COMPUTERS

MARKUS FAUSTMANN, MICHAEL FEISCHL, SEBASTIAN HIRNSCHALL

CONTENTS

1. Basics	2
1.1. Quantum Mechanics	2
1.2. Dirac Notation (Bra-Ket-Notation)	2
1.3. Digression: Tensor product of Hilbert spaces	3
2. Fundamental operations on qubits-gates	6
2.1. Single qubit gates	6
2.2. Two qubit gates	7
2.3. Three qubit gates	7
2.4. Quantum information	8
3. Quantum Algorithms	10
3.1. Deutsch-Josza-Algorithm	10
3.2. Simon's algorithm	16
3.3. The quantum Fourier Transform	17
3.4. Application Phase Estimation	19
3.5. Shor's integer factorization	21
3.6. Shor's period finding algorithm	24
3.7. Grover's algorithm	25
3.8. Numerical Quadrature	30
4. Linear systems of equations on a quantum computer	32
4.1. Quantum version of linear systems of equations	33
4.2. HHL-algorithm (Harrow-Hassidim-Lloyd)	34
4.3. Hamiltonian simulation	34
4.4. Graph coloring method	37
4.5. Modification for 3-diagonal matrices	41
4.6. Loading the right hand side	41
4.7. HHL revisited	43
4.8. Error and complexity analysis	45
4.9. Improvements on the HHL-algorithm	48
5. Solving Linear Differential Equations	54
5.1. Multi-step methods	55
References	58

1. BASICS

1.1. Quantum Mechanics. Our goal is eliminating physics in quantum computing by means of continuous quantum-models, formalized by evolution of $U(t)$ of state $\psi(t)$.

Remark 1. Consider a continuous operator U , i.e. $U(\varepsilon) = \text{id} - i\varepsilon H$ with H the Hamiltonian and $\varepsilon > 0$, small. Furthermore U is invertible with $U(t)^T U(t) = \text{id}$. Thus

$$\begin{aligned} U(\varepsilon)\psi(t) &= \psi(t + \varepsilon) = \psi(t) - i\varepsilon H\psi(t) \\ \iff \frac{\psi(t + \varepsilon) - \psi(t)}{\varepsilon} &= -iH\psi(t). \end{aligned}$$

For $\lim_{\varepsilon \rightarrow 0}$, we obtain the Schrödinger equation

$$\psi'(t) = -iH\psi(t),$$

with solution

$$\psi(p + t) = \underbrace{e^{-iHt}}_{U(t) \dots \text{unitary operator}} \psi(t).$$

1.2. Dirac Notation (Bra-Ket-Notation). Let V be a complex vector space with $v \in V$. Using the bra-ket-notation, we will write $|v\rangle := v$, called *ket-vector*. If $N = \dim V < \infty$, we may choose an orthonormal basis $\{|0\rangle, |1\rangle, \dots, |N-1\rangle\}$ of V .

Definition 2. A quantum state is a superposition of basis states

$$|\Phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \dots + \alpha_{N-1}|N-1\rangle$$

where $\alpha_i \in \mathbb{C}$ is the amplitude of $|i\rangle$ in $|\Phi\rangle$. We can therefore interpret

$$|\Phi\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{pmatrix}$$

as vector in \mathbb{C}^N . The bra-vector $\langle\Phi|$ is the conjugate transpose of $|\Phi\rangle$, i.e.

$$\langle\Phi| := |\Phi\rangle^H = (\bar{\alpha}_0, \bar{\alpha}_1, \dots, \bar{\alpha}_{N-1}).$$

The inner product of $|\Phi\rangle$ and $|\Psi\rangle$ is given by

$$\langle\Phi, \Psi\rangle =: \langle\Phi|\Psi\rangle = \langle\Phi| \cdot |\Psi\rangle$$

Throughout the rest of this section, we will consider V to be a Hilbert space.

(QM1): A state of the (isolated) quantum system is described by a unit vector ($\|\cdot\|_H = 1$) in a complex valued Hilbert space H .

Example 3. For $N = 2$ we have $|\Phi\rangle = a|0\rangle + b|1\rangle$ with $|a|^2 + |b|^2 = 1$.

Definition 4. A quantum system with 2-dimensional state space (ONB $|0\rangle, |1\rangle$) is called *qubit*.

(QM2): Let S_1, S_2 be systems with state spaces V, W . The composition of S_1 and S_2 is described by the tensor product $V \otimes W$.

1.3. **Digression: Tensor product of Hilbert spaces.** Let V, W be Hilbert spaces, our goal is to formally define the tensor product $V \otimes W$. Consider the following steps:

- define the *free vector space* $\mathcal{F}(V, W)$ as the set of all finite linear combinations of elements of $V \times W$, i.e.,

$$\mathcal{F}(V, W) = \left\{ \sum_{j=1}^n \alpha_j (v_j, w_j) : v_j \in V, w_j \in W, \alpha_j \in \mathbb{C} \right\}.$$

$\mathcal{F}(V, W)$ is closed under addition and scalar multiplication and is therefore a vector space. Note: $(v_1, w) + (v_2, w)$ is formally not the same as $(v_1 + v_2, w)$.

- Broadcast the vector space structure of V, W to $\mathcal{F}(V, W)$. The goal is to end up with

$$\begin{aligned} (v_1 + v_2) \otimes w &= v_1 \otimes w + v_2 \otimes w \\ v \otimes (w_1 + w_2) &= v \otimes w_1 + v \otimes w_2 \\ \alpha(v \otimes w) &= (\alpha v) \otimes w + v \otimes (\alpha w). \end{aligned}$$

Furthermore, define the subspace

$$U(V, W) = \text{span} \left\{ \sum_{j,k=1}^n \alpha_j \beta_k (v_j, w_j) - \left(\sum_{j=1}^n \alpha_j v_j, \sum_{k=1}^n \beta_k w_k \right) \right\}.$$

By this definition, $U(V, W) \subseteq \mathcal{F}(V, W)$, $U(V, W) \neq \emptyset$, $U(V, W)$ is a vector space, and an equivalence relation \sim is given by

$$(v_1, w_1) \sim (v_2, w_2) \iff (v_1, w_1) - (v_2, w_2) \in U(V, W).$$

- Let $\mathcal{F}(V, W)/\sim$ be the set of equivalence classes of $\mathcal{F}(V, W)$ under \sim . As $v \otimes w := (v, w) \in \mathcal{F}(V, W)/\sim$, $\mathcal{F}(V, W)/\sim$ has a vector space structure.
- Topology: Let $\langle \cdot, \cdot \rangle_V, \langle \cdot, \cdot \rangle_W$ be the inner products on V, W . Define

$$\langle v_1 \otimes w_1, v_2 \otimes w_2 \rangle := \langle v_1, v_2 \rangle_V \langle w_1, w_2 \rangle_W.$$

$\langle \cdot, \cdot \rangle$ is a sesquilinear form since

$$b : (V \times W) \times (V \times W) \rightarrow \mathbb{C} : (v_1, w_1), (v_2, w_2) \mapsto \langle v_1, v_2 \rangle_V \langle w_1, w_2 \rangle_W$$

is multi linear for Hilbert spaces V and W . Likewise,

$$\otimes \times \otimes : (V \times W) \times (V \times W) \rightarrow (V \otimes W) \times (V \otimes W)$$

is bilinear by construction and

$$b = \langle \cdot, \cdot \rangle \circ (\otimes \times \otimes).$$

- Definiteness: We have to show that $\langle \psi, \psi \rangle > 0$ for $\psi \neq 0$. To that end, let $\psi = \sum_i \lambda_i v_i \otimes w_i \in \mathcal{F}/\sim$ and choose the ONB $\{\phi_i\}$ for $\text{span}\{v_i\}$ and $\{\psi_i\}$ for $\text{span}\{w_i\}$. Then as per the definition of \sim , we have

$$\begin{aligned} \psi &= \sum_{j,k} \alpha_{jk} \phi_j \otimes \psi_k \\ \alpha_{jk} &= \sum_i \lambda_i \langle \phi_j, v_i \rangle_V \langle \psi_k, w_i \rangle_W, \end{aligned}$$

which implies

$$\langle \psi, \psi \rangle = \sum_{j,k} |\alpha_{jk}|^2 > 0.$$

Suppose $\langle \psi, \psi \rangle = 0$, then $\alpha_{jk} = 0$ for all j, k and thus $\psi = 0$.

- Completion of \mathcal{F}/\sim with norm $\|\psi\|_2 = \langle \psi, \psi \rangle$ will lead to $V \otimes W$.

Lemma 5. *Let $\{\phi_i\}$ be an ONB of V and $\{\psi_i\}$ be an ONB of W . Then $\{\phi_i \otimes \psi_j\}$ is an ONB of $V \otimes W$.*

Proof. The orthonormality of $\{\phi_i \otimes \psi_j\}$ follows from the orthonormality of $\{\phi_i\}$ and $\{\psi_j\}$ on V and W respectively and the definition of $\langle \cdot, \cdot \rangle$. Since ϕ_i, ψ_j are ONBs, $\text{span}\{\phi_i \otimes \psi_j\}$ is dense in \mathcal{F}/\sim and hence also in $V \otimes W$. \square

Example 6.

- $\mathbb{R}^2 \otimes \mathbb{R}^2$ is 4-dimensional with

$$\mathbb{R}^2 \otimes \mathbb{R}^2 = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

and is called a 2-qubit state.

- $\mathbb{R}^3 \otimes \mathbb{R}^2$ is 6-dimensional.

In general, Lemma 5 implies that $\dim(V \otimes W) = \dim V \cdot \dim W$.

Example 7. Let H be a Hilbert space then

$$H \otimes \mathbb{C}^n = H^n = H \times \dots \times H$$

Remark 8. The construction of the tensor product is unique up to unitary transformations. Let $\bar{\otimes}$ be a sesquilinear form compatible with $\langle \cdot, \cdot \rangle_V, \langle \cdot, \cdot \rangle_W$. Then there exists a unitary operator $U : V \otimes W \rightarrow V \bar{\otimes} W$.

n -qubit systems have 2^n -dimensional state spaces. We will write

$$|b_1 b_2 \dots b_n\rangle := |b_1\rangle \otimes |b_2\rangle \otimes \dots \otimes |b_n\rangle$$

and relabel the basis states as $|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$.

Registers of n -qubits: superposition

$$|\Phi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

with $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$.

Definition 9. A state $|\psi\rangle$ is called product state if it can be written as

$$|\psi\rangle = |v_1\rangle \otimes \dots \otimes |v_n\rangle, \quad v_i \in H,$$

otherwise it is called entangled.

Example 10. The state

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

is entangled. To see this, consider

$$\begin{aligned} \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle &= (a|0\rangle + b|1\rangle) - (c|0\rangle + d|1\rangle) \\ &= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle, \end{aligned}$$

which implies

$$\implies ac = bd = 0 \quad \wedge \quad ad = bc = 0 \implies a \vee d = 0$$

which is a contradiction. \nexists

Example 11. The state $1/\sqrt{2}|00\rangle + 1/\sqrt{2}|10\rangle = (1/\sqrt{2}|0\rangle + 1/\sqrt{2}|1\rangle) \otimes |0\rangle$ is a product state.

(QM3): The evolution of a quantum system is described only by unitary operations acting on the state space.

Conservation of length: Let $|\Phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$, $|\alpha_0|^2 + |\alpha_1|^2 = 1$, be a qubit with

$$U(\Phi) = |\psi\rangle = \beta_0|0\rangle + \beta_1|1\rangle.$$

As U is unitary, we have $|\beta_0|^2 + |\beta_1|^2 = 1$ for $|\psi\rangle$ to be a qubit.

(QM4): Measurement outcome of quantum states are probability distributions.

Remark 12. Born's rule: The probability of measuring $|j\rangle$ is $|\alpha_j|^2$. Measuring a quantum state is invasive, i.e., the state $|\Phi\rangle$ collapses to the measured state $|j\rangle$.

1.3.1. *Projective measurement:* Let V be a state space with $\dim V = n < \infty$ and P_i , $i = 1, \dots, m$ the orthogonal projections onto $V_i \subseteq V$ with

$$\begin{aligned} \sum_{j=1}^m P_j &= \text{id} \\ P_i P_j &= 0 \quad \text{for } i \neq j \\ P_i P_i &= P_i. \end{aligned}$$

There are m possible outcomes, hence

$$|\Phi\rangle = \sum_{j=1}^m |\Phi_j\rangle$$

with $|\Phi\rangle \in V$ and $|\Phi_j\rangle = P_j|\Phi\rangle \in V_j$. The probability of measuring j is

$$\| |\Phi_j\rangle \|^2 = \langle \Phi | P_j P_j | \Phi \rangle = \langle \Phi | P_j | \Phi \rangle$$

and the state collapses to $|\Phi_j\rangle / \| |\Phi_j\rangle \|$.

Remark 13.

- A-priori it is not possible to choose which projections P_j will be used, only their probabilities.
- If $|\Phi\rangle \in V_j$, measurement will result in j with $P = 1$.

Example 14. Let $V = \text{span}\{|0\rangle, \dots, |N-1\rangle\}$ with $V_j = |j\rangle$. We can use Born's rule as $P_j|\Phi\rangle = \alpha_j|j\rangle$ is likely with $\|P_j|\Phi\rangle\|^2 = \|\alpha_j|j\rangle\|^2 = |\alpha_j|^2$. We define P_j as $P_j := |j\rangle\langle j|$, with $\text{rank}(P_j) = 1$.

This is called complete measurement.

Example 15. We will now consider a measurement that for $|j\rangle$ only discriminates between $j < N/2$ and $j \geq N/2$. Let

$$\begin{aligned} P_1 &:= \sum_{j < N/2} |j\rangle\langle j| \\ P_2 &:= \sum_{j \geq N/2} |j\rangle\langle j| \end{aligned}$$

be projections for

$$|\Phi\rangle = \frac{1}{\sqrt{3}}|1\rangle + \sqrt{\frac{2}{3}}|N\rangle.$$

Hence, we measure

- 1 with probability $\|P_1|\Phi\rangle\|^2 = \frac{1}{3}$
- 2 with probability $\|P_2|\Phi\rangle\|^2 = \frac{2}{3}$.

This is called *incomplete measurement*.

2. FUNDAMENTAL OPERATIONS ON QUBITS-GATES

Definition 16. A unitary operator U on a small number of qubits is called *gate*.

A *gate* is similar to e.g., AND , XOR , and OR gates in classical computing.

The programming is done using *quantum circuits*. On classical computers we have Boolean circuits, i.e., finite oriented graphs with AND ,OR , and NOT operations.

$$\xrightarrow{\text{input}} \{\text{AND , OR , NOT } s\} \xrightarrow{\text{output}}$$

Quantum circuits replace AND ,OR , and NOT with quantum gates.

2.1. Single qubit gates.

2.1.1. *Bit flip.* The *bit flip* gate X (*NOT*) swaps $|0\rangle$ and $|1\rangle$ and is defined by

$$X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

2.1.2. *Phase flip.* The *phase flip* gate Z mirrors $|1\rangle$ and is defined by

$$Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Note that this is a special case of the *phase-gate* R_Φ defined as

$$R_\Phi := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\Phi} \end{pmatrix}$$

which rotates $|1\rangle$ by $\Phi_G[-\bar{u}, \bar{u}]$. $R_{\pi/4}$ is called *T-gate*.

2.1.3. *Hadamard gate.* The *Hadamard-gate* $\boxed{\text{H}}$ is the most important single qubit operator. It is defined as

$$\boxed{\text{H}} := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Since

$$\boxed{\text{H}}|0\rangle = \boxed{\text{H}} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \simeq \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle,$$

we see that the probability for $|0\rangle$ and $|1\rangle$ is equal.

Furthermore, we have

$$\boxed{\text{H}} = \boxed{\text{H}}^T = \boxed{\text{H}}^{-1} \implies \boxed{\text{H}} \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \right) = |0\rangle.$$

2.2. Two qubit gates.

2.2.1. *XOR* . Given $a, b \in \{0, 1\}$, we want to implement the operation $(a, b) \mapsto a \text{ XOR } b$ defined by the following truth table:

a \ b	0	1
0	0	1
1	1	0

However, as $(a, b) \mapsto a \text{ XOR } b$ is not unitary, we define $(a, b) \mapsto (a, a \text{ XOR } b)$ as

(a, b)	$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
$(a, a \text{ XOR } b)$	$ 00\rangle$	$ 01\rangle$	$ 11\rangle$	$ 10\rangle$

which is a permutation and thus unitary. This operation is called *Controlled Not* (CNOT) and can be represented by the circuit diagram shown in Figure 1 below.

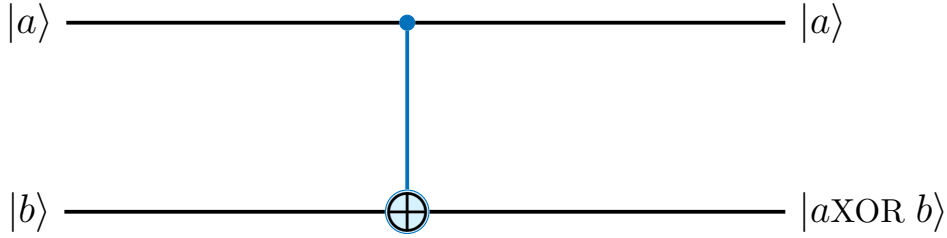


FIGURE 1. Controlled Not (*CNOT*) circuit diagram

If we identify $|00\rangle, \dots, |11\rangle$ with e_1, \dots, e_4 we may represent CNOT by the unitary matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (1)$$

2.3. Three qubit gates.

2.3.1. *AND* . The AND operation, $(a, b) \mapsto a \text{ AND } b$ with $a, b \in \{0, 1\}$, is defined by the following truth table:

a \ b	0	1
0	0	0
1	0	1

Note that $(a, b) \mapsto (a, a \text{ AND } b)$ is not bijective.

Requiring an extra qubit $|c\rangle$, we may define a permutation $|a, b, c\rangle \mapsto |a, b, c \text{ XOR } (a \text{ AND } b)\rangle$ as

$ abc\rangle$	$ 000\rangle$	$ 001\rangle$	$ 010\rangle$	$ 011\rangle$	$ 100\rangle$	$ 101\rangle$	$ 110\rangle$	$ 111\rangle$
	$ 000\rangle$	$ 001\rangle$	$ 010\rangle$	$ 011\rangle$	$ 100\rangle$	$ 101\rangle$	$ 111\rangle$	$ 110\rangle$

We see that for inputs $|c\rangle = 0$, the output $c' = |a, b, a \text{ AND } b\rangle$. The extra qubit $|c\rangle$ is called *ancilla*-qubit (ancilla = Diener). Figure 2 shows the circuit diagram for this operation.

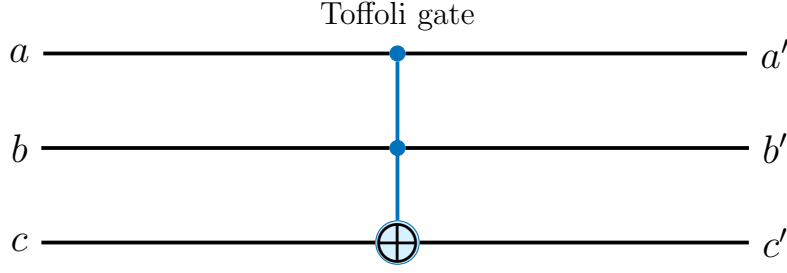


FIGURE 2. Toffoli gate circuit diagram

Remark 17. Let U be a unitary n -qubit operation and let $\text{id} \in \mathbb{R}^{2^n} \times \mathbb{R}^{2^n}$, then the operation

$$\begin{pmatrix} \text{id} & 0 \\ 0 & U \end{pmatrix} \in \mathbb{C}^{2^{n+1}} \times \mathbb{C}^{2^{n+1}}$$

is called *controlled- U* .

Remark 18. The Toffoli gate can be used to implement *NOT*. Consider

$$\text{Toffoli}(|11a\rangle) = |111a\rangle$$

and fix the first 2 qubits since

$$A \text{ OR } B = \text{NOT}(\text{NOT } A \text{ AND } \text{NOT } B).$$

Thus, any classical circuit can be implemented by a circuit of Toffoli gates.

2.4. Quantum information. So far, we know that:

- 2 states can only be distinguished with probability 1 if they are in orthogonal subspaces.
- There are 3 possible manipulations of $|\psi\rangle$:
 - **(Ancilla)** combining a known state $|A\rangle$ with $|\psi\rangle$ we obtain $|\psi\rangle|A\rangle$, enlarging the dimension of the state space,
 - **(Unitary operation)** $U|\psi\rangle$ with U unitary,
 - **(Measurement)** a state $|\psi\rangle$ collapses to $|j\rangle$.

Can we also copy or save information? The answer is no, as the *no-cloning theorem* states that states cannot be copied.

2.4.1. Setting. Let H be a state space for 2 quantum systems

$$\begin{array}{ll} A & \text{(contains } |\psi\rangle, \text{ the state to be copied)} \\ B & \text{(contains } |0\rangle, \text{ the state to be copied to)} \end{array}$$

and one quantum system

$$C \quad (\text{"copier", state } |M_0\rangle).$$

2.4.2. Goal. The goal is to find an operation

$$|\psi\rangle_A |0\rangle_B |M_0\rangle_M \rightarrow |\psi\rangle_A |\psi\rangle_B |M_\psi\rangle_M$$

that works for all states in A .

Theorem 19. No-cloning theorem

Suppose $S \subseteq H$ such that S contains at least a few different, non-orthogonal states. Then

$$\nexists \text{ unitary } U \text{ on } S, \text{ that can copy all states.}$$

Proof. Let $|\xi\rangle, |\eta\rangle \in S$ be non-orthogonal. Furthermore,

$$\begin{aligned} U|\xi\rangle_A|0\rangle_B|M_0\rangle_M &= |\xi\rangle_A|\xi\rangle_B|M_\xi\rangle_M \\ U|\eta\rangle_A|0\rangle_B|M_0\rangle_M &= |\eta\rangle_A|\eta\rangle_B|M_\eta\rangle_M \end{aligned}$$

has to hold. Now, since U is unitary, U preserves the inner product, i.e.,

$$\begin{aligned} \langle\xi|\eta\rangle\langle\xi|\eta\rangle\langle M_s|M_s\rangle &= \langle|\xi\rangle|\xi\rangle|M_\xi\rangle, |\eta\rangle|\eta\rangle|M_\eta\rangle\rangle \\ &= \langle|\xi\rangle|0\rangle|M_0\rangle, |\eta\rangle|0\rangle|M_0\rangle\rangle = \langle\xi|\eta\rangle \underbrace{\langle 0|0\rangle}_{|\cdot|=1} \underbrace{\langle M_0|M_0\rangle}_{|\cdot|=1}. \end{aligned}$$

Hence, we have

$$|\langle\xi|\eta\rangle|^2|\langle M_\xi|M_\eta\rangle| = |\langle\xi|\eta\rangle|$$

and since $|\langle\xi|\eta\rangle| \neq 0$ as $|\xi\rangle, |\eta\rangle$ are non-orthogonal, we get

$$|\langle\xi|\eta\rangle||\langle M_\xi|M_\eta\rangle| = 1.$$

Using Cauchy-Schwarz inequality, we also know that

$$|\langle M_\xi|M_\eta\rangle| \leq \underbrace{|\langle M_\xi|M_\xi\rangle|}_{=1} \underbrace{|\langle M_\eta|M_\eta\rangle|}_{=1} = 1.$$

Combining the last two equations leads to

$$|\langle\xi|\eta\rangle| = 1.$$

However, this is a contradiction as $|\xi\rangle, |\eta\rangle$ are non-orthogonal. \nexists

□

Corollary 20. *No-deleting theorem*

Assume the same setting as in Theorem 19. Then \nexists unitary U such that

$$U : |\psi\rangle_A|\psi\rangle_B|M_0\rangle_M \rightarrow |\psi\rangle_A|0\rangle_B|M_\psi\rangle_M.$$

The solution for information transfer is *entanglement*.

Example 21. Consider the following setting:

- Alice has a qubit $|\alpha\rangle := \alpha_0|0\rangle + \alpha_1|1\rangle$.
- Bob is far away but wants information about Alice's qubit.
- Alice and Bob share an entangled state

$$\begin{array}{c} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \\ \swarrow \quad \downarrow \quad \downarrow \quad \swarrow \\ \text{Alice} \quad \text{Bob} \quad \text{Alice} \quad \text{Bob} \end{array}$$

How can Bob get information about Alice's qubit without physically transferring Alice's qubit?

Teleportation:

- There are 3 qubits involved:
 - 1. qubit: Alice's qubit $|\alpha\rangle$,
 - 2. qubit: Alice's part of the entangled qubit, and
 - 3. qubit: Bob's part of the entangled qubit.

The combined system is given by

$$(\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

- Alice applies CNOT to the first and second qubit, i.e.,

$$\frac{\alpha_0}{\sqrt{2}}|0\rangle(|00\rangle + |11\rangle) + \frac{\alpha_1}{\sqrt{2}}|1\rangle(|01\rangle + |11\rangle).$$

- Alice applies $\boxed{\text{H}}$ to the first qubit to obtain

$$\begin{aligned} & \frac{1}{2}\alpha_0(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \frac{1}{2}\alpha_1(|0\rangle - |1\rangle)(|10\rangle + |01\rangle) \\ &= \frac{1}{2}|00\rangle(\alpha_0|0\rangle + \alpha_1|1\rangle) + \frac{1}{2}|01\rangle(\alpha_0|1\rangle + \alpha_1|0\rangle) \\ &+ \frac{1}{2}|10\rangle(\alpha_0|0\rangle - \alpha_1|1\rangle) + \underbrace{\frac{1}{2}|11\rangle}_{A. \text{ qubits}}(\alpha_0|1\rangle - \alpha_1|0\rangle). \end{aligned}$$

- Alice measures both qubits. The state collapses to one of 4 possible states with probability $P = 1/4$.

measurement	state after measurement
$ 00\rangle$	$ 00\rangle \alpha\rangle$
$ 01\rangle$	$ 01\rangle X \alpha\rangle$ (bitflip)
$ 10\rangle$	$ 10\rangle Z \alpha\rangle$ (phaseflip)
$ 11\rangle$	$ 11\rangle XZ \alpha\rangle$ (bit+phaseflip)

- Alice sends the measurement result to Bob (e.g. 10).
- Bob looks up the corresponding operation in the table above and applies the inverse transformation to his qubit. In this case (10), Bob applies the inverse of Z to his qubit. Thus Bob's qubit is guaranteed to be in the state $|\alpha\rangle$.

Remark 22.

- Physical processes in space between Alice and Bob have no influence.
- Only information can be transported, not matter. No StarTrek-beaming!
- Alice has to measure her qubits. This is invasive and the state collapses (no longer in superposition!).

3. QUANTUM ALGORITHMS

3.1. Deutsch-Josza-Algorithm.

3.1.1. *Problem.* Given $\mathcal{F}(x_1, \dots, x_n) \in \{0, 1\} : x_i \in \{0, 1\}$ determine if

- $\mathcal{F} = 0$,
- $\mathcal{F} = 1$, or
- \mathcal{F} is *balanced*, i.e. exactly half of (x_1, \dots, x_n) lead to $\mathcal{F}(x_1, \dots, x_n) = 1$ and the other half leads to $\mathcal{F}(x_1, \dots, x_n) = 0$.

We will assume that \mathcal{F} satisfies one of the three options.

Although this problem is not useful in practice, it demonstrates that hard problems can be easier on quantum computers.

3.1.2. *Classic Algorithm.* To exclude the *balanced* case, we need to check at least $2^{n-1} + 1$ inputs $(x_1, \dots, x_n) \in \{0, 1\}^n$ which results in problem size of $N = 2^n$ with run time $N/2 + 1$.

3.1.3. *Physical Inspiration.* Consider the setup depicted in Figure 3: A light source (Sun) emits light rays which are directed towards a wall with 2^n holes $(x_1, \dots, x_n) \in \{0, 1\}^n$. The light passes through the holes and is detected by a detector. Assume that the holes are arranged such that the light amplitude at the detector $\simeq (-1)^{\mathcal{F}(y)}$ if the light travels through hole y . The light intensity at the detector is determined by the phase difference of light paths (interference) and is given by

$$\simeq \sum_{y \in \{0,1\}^n} (-1)^{\mathcal{F}(y)} = \begin{cases} 0 & \mathcal{F} \text{ is balanced} \\ \pm 1 & \mathcal{F} \text{ is const.} \end{cases} \quad (2)$$

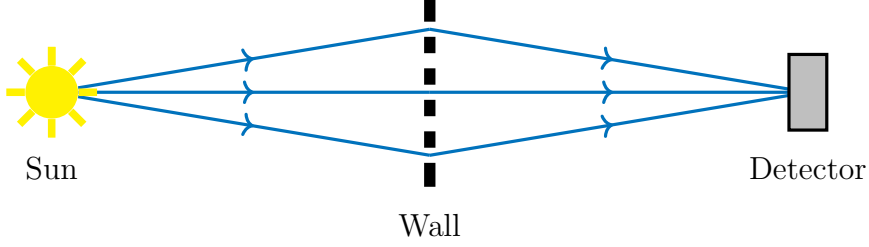


FIGURE 3. Physical inspiration for Deutsch-Josza algorithm

3.1.4. *Quantum Algorithm.* We may identify $(x_1, \dots, x_n) \in \{0, 1\}^n$ with the basis element $|i\rangle$, where $i = x_1 + x_2 2 + x_3 4 + \dots + x_n 2^{n-1}$. In this sense, we define $\mathcal{F}(|i\rangle) := |\mathcal{F}(x_1, \dots, x_n)\rangle$. An essential part of this (and many other algorithms) is the *Query*: Given a 1-qbit state $|b\rangle$ and $\mathcal{F}(x_1, \dots, x_n) \in \{0, 1\}$, define

$$\mathcal{Q}_{\mathcal{F}}(|i\rangle \otimes |b\rangle) = |i\rangle \otimes (|b\rangle \oplus \mathcal{F}(|i\rangle)). \quad (3)$$

Particularly for $|b\rangle = |-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ we have

$$\mathcal{Q}_{\mathcal{F}}(|i-\rangle) = |i\rangle \otimes \frac{1}{\sqrt{2}} [\mathcal{F}(|i\rangle) - |1\rangle \oplus \mathcal{F}(|i\rangle)] \quad (4)$$

$$= \begin{cases} |i\rangle \otimes \frac{1}{\sqrt{2}} [|0\rangle - |1\rangle] & \mathcal{F}(|i\rangle) = |0\rangle \\ |i\rangle \otimes \frac{1}{\sqrt{2}} [|1\rangle - |0\rangle] & \mathcal{F}(|i\rangle) = |1\rangle \end{cases} \quad (5)$$

$$= (-1)^{\mathcal{F}(|i\rangle)} |i-\rangle. \quad (6)$$

Note that we identify $\mathcal{F}(|i\rangle)$ with $\mathcal{F}(x_1, \dots, x_n)$ instead of $|\mathcal{F}(x_1, \dots, x_n)\rangle$, i.e., the last equality only makes sense for basis elements $|i\rangle$. Furthermore, $\mathcal{Q}_{\mathcal{F}}$ is unitary since

$$|b\rangle \mapsto (|b\rangle \oplus \mathcal{F}(|a\rangle))$$

is a permutation for any value of $\mathcal{F}(a)$.

We make extensive use of the Hadamard gate $\boxed{\text{H}}$ given by $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. We already know that

$$\begin{aligned}\boxed{\text{H}}|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ \boxed{\text{H}}|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).\end{aligned}$$

Lemma 23. *There holds*

$$\begin{aligned}\boxed{\text{H}}^{\otimes n}|0^n\rangle &:= \bigotimes_{i=1}^n \boxed{\text{H}}|0\rangle = \bigotimes_{i=1}^n \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \sum_{(x_1, \dots, x_n)} \frac{1}{\sqrt{2^n}} |x_1, \dots, x_n\rangle \\ &= \frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} |i\rangle\end{aligned}$$

as well as for $i = x_1 + 2x_2 + \dots + 2^{n-1}x_n$

$$\begin{aligned}\boxed{\text{H}}^{\otimes n}|i\rangle &:= \bigotimes_{i=1}^n \boxed{\text{H}}|x_i\rangle = \bigotimes_{i=1}^n \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_i}|1\rangle) \\ &= \sum_{(y_1, \dots, y_n) \in \{0,1\}^n} \frac{1}{\sqrt{N}} (-1)^{x \cdot y} |y_1, \dots, y_n\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{2^n-1} (-1)^{i \cdot j} |j\rangle\end{aligned}$$

where $i \cdot j := x \cdot y := \sum_{k=1}^n x_k y_k$.

Algorithm 1 (Deutsch-Josza Quantum Algorithm).

- 1: Initial State $|0^n\rangle := |0\rangle \otimes \dots \otimes |0\rangle$
- 2: Apply $\boxed{\text{H}}$ to each qubit to obtain uniform state

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} |i\rangle$$

- 3: Tensorize result with $|-\rangle$ to obtain

$$\frac{1}{\sqrt{N}} \sum |i-\rangle$$

- 4: Apply query with $\boxed{Q_{\mathcal{F}}}$ to obtain $|b\rangle = |-\rangle$

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} (-1)^{\mathcal{F}(|i\rangle)} |i-\rangle$$

- 5: Ignore last qubit $|-\rangle$ and apply $\boxed{\text{H}}$ to remaining state to obtain Lemma 23

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} \sum_{j=0}^{2^n-1} (-1)^{i \cdot j} (-1)^{\mathcal{F}(|i\rangle)} |j\rangle$$

- 6: The amplitude of the $j = 0 \iff |0^n\rangle$ -state is

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} (-1)^{i \cdot 0} (-1)^{\mathcal{F}(|i\rangle)}$$

7: **if** \mathcal{F} is balanced **then**

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} (-1)^{\mathcal{F}(|i\rangle)} = 0$$

8: **else if** $\mathcal{F} = 1$ **then**

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} (-1)^{\mathcal{F}(|i\rangle)} = -1$$

9: **else if** $\mathcal{F} = 0$ **then**

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} (-1)^{\mathcal{F}(|i\rangle)} = 1$$

10: **end if**

3.1.5. *Complexity.* Algorithm 1 needs $\mathcal{O}(1)$ steps, $\mathcal{O}(\log_2(N))$ quantum gates and $\mathcal{O}(\log(N))$ qubits, constituting an *exponential speedup*.

Example for \mathcal{F} :

$$\mathcal{F}(x_1, x_2, x_3) := x_1 + x_2x_3 \bmod 2$$

is *balanced* since $\mathcal{F}(0, x_2, x_3) = 1 - \mathcal{F}(1, x_2, x_3)$. For the algorithm to work, \mathcal{F} needs to be a quantum circuit, i.e. $\mathcal{F}(x_1, x_2, x_3) = x_1 \text{XOR } (x_2 \text{AND } x_3)$ as shown in Figure 4

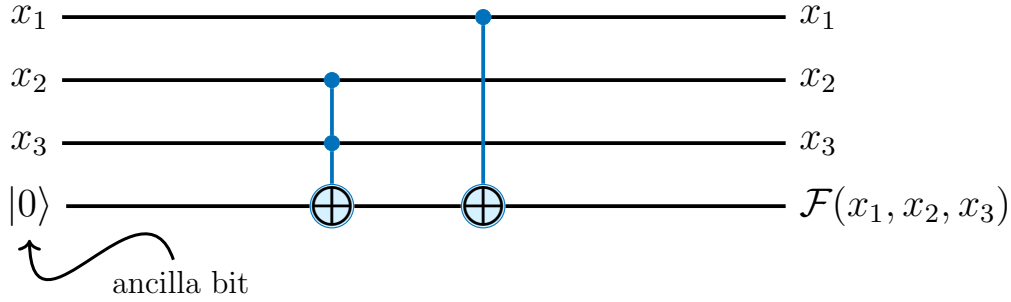


FIGURE 4. Circuit diagram for \mathcal{F}

Next, implement the Query $\boxed{Q_{\mathcal{F}}}(|i-\rangle) = |i\rangle \otimes (|i-\rangle \oplus \mathcal{F}(|i\rangle))$.

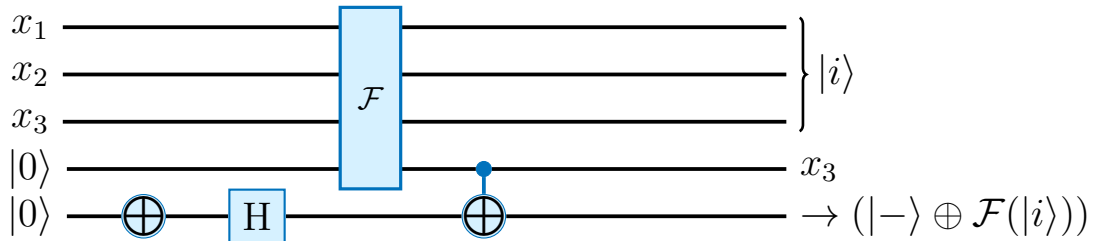


FIGURE 5. $\boxed{Q_{\mathcal{F}}}$ circuit diagram

Apply Lines 2 to 4 of Algorithm 1. The circuit diagram for this operation is depicted in Figure 6.

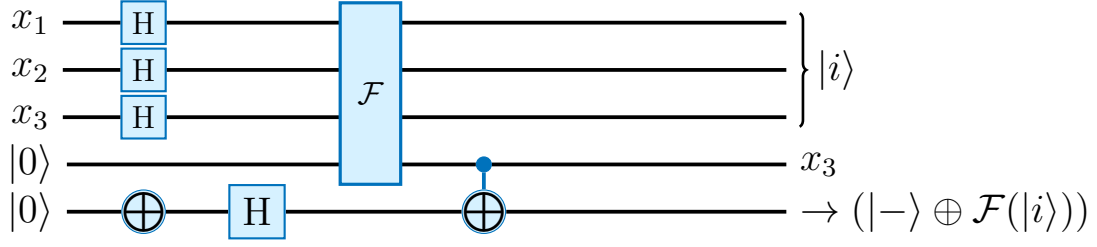


FIGURE 6. Implementation of Lines 2 to 4

Now, as $|y\rangle$ and $|i\rangle$ are entangled, reverse \mathcal{F} such that $|y\rangle = |0\rangle$ (Figure 7).

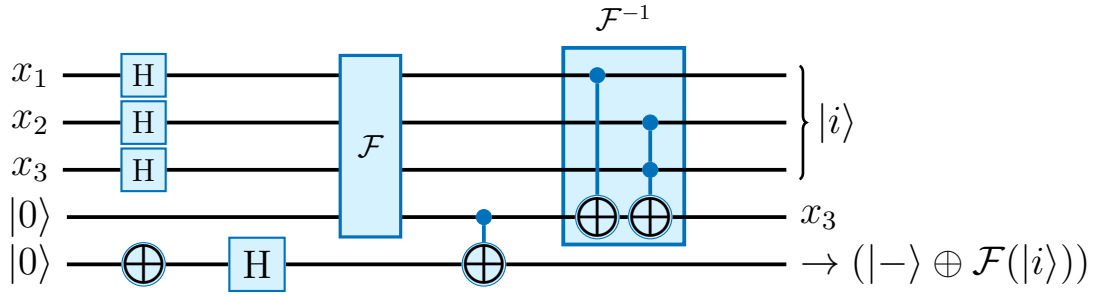


FIGURE 7. Circuit diagram after reversing \mathcal{F}

At this point we have the desired state

$$|i\rangle \otimes (|i\rangle \oplus \mathcal{F}(|i\rangle)) (\otimes |0\rangle).$$

Finally, apply Line 5 of Algorithm 1 and measure the first 3 qubits.

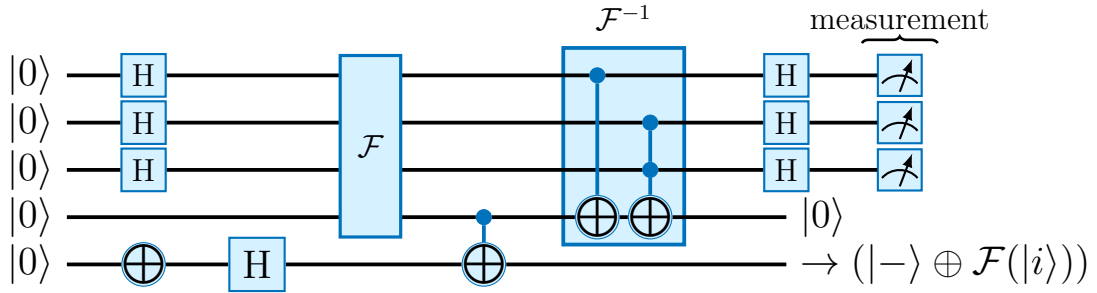
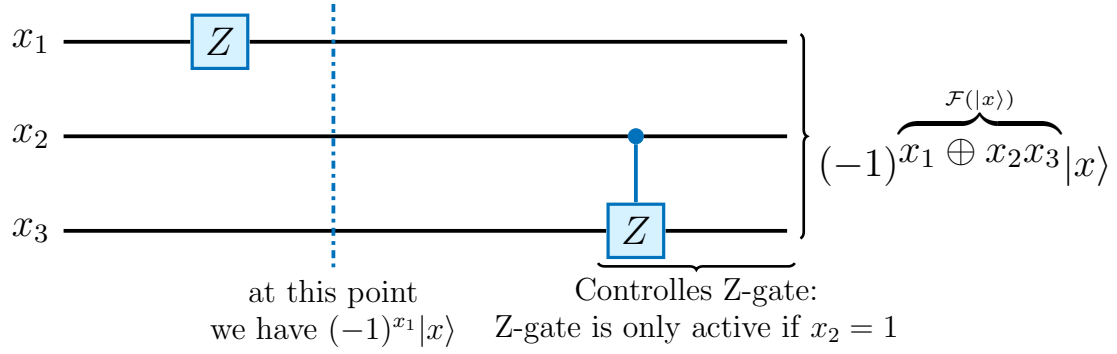


FIGURE 8. Circuit diagram for Algorithm 1

The measurement must contain the projection onto $|000\rangle \otimes \text{span}\{|0\rangle, |1\rangle\}^{\otimes 2}$.

Note that this is not the most efficient implementation of $\boxed{Q_{\mathcal{F}}}$



$$\boxed{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \Rightarrow (-1)^{x_2 x_3} |x\rangle$$

Remark 24. *Deutsch-Josza*

- First algorithm with exponential speed-up over classical computer
- However, consider the following randomized classical algorithm $\mathcal{R}(\mathcal{F})$:
 - 1: Generate two random $x, y \in \{0, 1\}^n$
 - 2: Evaluate $\mathcal{F}(x), \mathcal{F}(y)$
 - 3: Output:

$$\begin{aligned} \mathcal{R}(\mathcal{F}) &= 1 && \text{if } \mathcal{F}(x) = \mathcal{F}(y) = 1 \\ \mathcal{R}(\mathcal{F}) &= 0 && \text{if } \mathcal{F}(x) = \mathcal{F}(y) = 0 \\ \mathcal{R}(\mathcal{F}) &\text{ balanced} && \text{if } \mathcal{F}(x) \neq \mathcal{F}(y). \end{aligned}$$

- The algorithm is correct if \mathcal{F} is constant and answers correctly with prob. $1/2$ if \mathcal{F} is balanced.
- Apply the algorithm k -times with i.i.d. random samples to get $(\mathcal{R}_1(\mathcal{F}), \dots, \mathcal{R}_n(\mathcal{F}))$.
Return

$$\begin{aligned} 1 &&& \text{if } \mathcal{R}_1(\mathcal{F}) = \dots = \mathcal{R}_n(\mathcal{F}) = 1 \\ 0 &&& \text{if } \mathcal{R}_1(\mathcal{F}) = \dots = \mathcal{R}_n(\mathcal{F}) = 0 \\ \text{balanced} &&& \text{otherwise.} \end{aligned}$$

This results in an error probability of 2^{-k} and cost $\mathcal{O}(k)$.

Remark 25. Why do you have to uncompute \mathcal{F} in order to remove entanglement? After Line 4, we have

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} (-1)^{\mathcal{F}(|i\rangle)} |i-\rangle$$

but in the implementation, we actually have

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} (-1)^{\mathcal{F}(|i\rangle)} |i\rangle \otimes |a_i\rangle \otimes |-\rangle$$

for some ancilla qubit $|a_i\rangle$. We may ignore the last qubit since it is not entangled.

Line 5 applies $\boxed{H}^{\otimes n}$ to obtain

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (-1)^{\mathcal{F}(|i\rangle)} (-1)^{i \cdot j} |j\rangle \otimes |a_i\rangle.$$

It follows, that the Amplitude of $|j\rangle = |0\rangle$ is

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{\mathcal{F}(|i\rangle)} |0\rangle \otimes |a_i\rangle$$

which might be non-zero even for balanced \mathcal{F} .

3.2. Simon's algorithm. First quantum algorithm with exponential speed-up over any (randomized) classical algorithm. Given $i, j \in \{0, 1\}^n$, define $i \oplus j := (i_1 \oplus j_1, \dots, i_n \oplus j_n)$.

3.2.1. Simon's problem. Let $\mathcal{F} : \{0, 1\}^n \rightarrow \{0, 1\}^n$ s.t. there exists $s \in \{0, 1\}^n$

$$\mathcal{F}(x) = \mathcal{F}(y) \iff x = y \text{ or } x \oplus s = y \quad \forall x, y \in \{0, 1\}^n$$

3.2.2. Goal. Find s

3.2.3. Quantum Algorithm.

Algorithm 2. Simon's problem quantum algorithm

1: Start with $|0^n\rangle|0^n\rangle$ and apply $\boxed{\text{H}}$ to the first n qubits to obtain

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} |i\rangle|0^n\rangle$$

2: Apply the query $|i\rangle \otimes |b^n\rangle \mapsto |i\rangle \otimes (|b^n\rangle \oplus \mathcal{F}(|i\rangle))$ with $|b^n\rangle = |0^n\rangle$ to obtain

$$\frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} |i\rangle|\mathcal{F}(|i\rangle)\rangle$$

3: Measure the second n qubit in computational basis, i.e. $\text{span}\{|j\rangle, j = 0, \dots, 2^n - 1\} \otimes \text{span}\{|i\rangle, i = 0, \dots, 2^n - 1\}$ to obtain some output

$$|a\rangle \otimes |j\rangle = \frac{\mathcal{P}_j \left(\frac{1}{\sqrt{N}} \sum |i\rangle \mathcal{F}(|i\rangle) \right)}{\|\mathcal{P}_j(\dots)\|}.$$

Note that $|i\rangle \mathcal{F}(|i\rangle) \perp \text{ran } \mathcal{P}_j$ for $\mathcal{F}(|i\rangle) \neq |j\rangle$. By assumption, there exist exactly two inputs $|i_0\rangle$ and $|i_1\rangle = |i_0 \oplus s\rangle$ with $\mathcal{F}(|i_k\rangle) = |j\rangle$, $k = 0, 1$. Hence $|a\rangle = 1/\sqrt{2}(|i_0\rangle + |i_0 \oplus s\rangle)$.

4: Ignore second n qubits and apply $\boxed{\text{H}}$ to the first n to obtain

$$\frac{1}{\sqrt{2N}} \sum_{j=0}^{2^n-1} [(-1)^{i \cdot j} + (-1)^{(i_0 \oplus s) \cdot j}] |j\rangle.$$

5: Measure in computational basis:

$|j\rangle$ has non-zero duplicate if $(i_0 \oplus s) \cdot j = i_0 \cdot j \iff s \cdot j = 0 \pmod{2}$.

Thus, we obtain a random element of the set $\{j | s \cdot j = 0 \pmod{2}\}$.

6: Repeat the procedure to obtain $n-1$ linear independent elements $j^{(1)}, \dots, j^{(n-1)}$ with

$$j^{(i)} \cdot s = 0 \pmod{2}$$

or

$$\begin{pmatrix} j^{(1)} \\ \vdots \\ j^{(n-1)} \end{pmatrix} s = 0 \pmod{2}$$

7: Solve linear system $\pmod{2}$ on classical computer in $\mathcal{O}(n^3)$

Remark 26. Note that $\#\text{span}\{j^{(1)}, \dots, j^{(k)}\} \leq 2^k$. Hence, with probability $\frac{2^n - 2^k}{2^n} = 1 - 2^{k-n} \geq \frac{1}{2}$ for $k \leq n-1$, we find a linear independent vector j_{k+1} .

3.2.4. *Conclusion.* Algorithm 2 requires $\mathcal{O}(n)$ qubits, $\mathcal{O}(n)$ gates, and $\mathcal{O}(n)$ iterations with high probability $+ \mathcal{O}(n^3)$.

3.2.5. *Classical algorithms for Simon's problem.*

Lemma 27. Any (randomized) classical algorithm with less than $\delta 2^{n/2}$ queries to \mathcal{F} fails with probability $\geq \exp(-(5/4)\delta^2)$ (deterministic algorithm fails certainly if less than $2^{n/2}$ queries).

Proof. Every algorithm generates a sequence of queries x_1, \dots, x_n with $\underbrace{\mathcal{F}(x_1)}_{=:y_1}, \dots, \underbrace{\mathcal{F}(x_n)}_{=:y_n}$.

If all y_i are distinct, the algorithm can't distinguish between $s = 0$ and $s \neq 0$.

Assume all y_1, \dots, y_n are distinct. Then the algorithm chooses x_{k+1} based on some probability. Measure μ on $\{0, 1\}^n$ [in the det. case, μ is a delta distribution].

$$\begin{aligned} \sum_{k=1}^c \sum_{s \in \{0,1\}^n} \sum_{i=1}^k \mu(x_i \oplus s) &= \sum_{k=1}^c \sum_{i=1}^k \underbrace{\sum_{s \in \{0,1\}^n} \mu(x_i \oplus s)}_{=1} = \frac{c(c+1)}{2} \\ \implies \exists s \in \{0, 1\}^n : \underbrace{\sum_{k=1}^c \sum_{i=1}^k \mu(x_i \oplus s)}_{p_i \leq 1/2} &\leq \frac{c(c+1)}{2^{n+1}} \leq \frac{1}{2} \quad \text{for } c(c+1) \leq 2^n \end{aligned}$$

Hence, $\mathbb{P}(y_1, \dots, y_{k+1} \text{ distinct}) = \underbrace{\mathbb{P}(y_{k+1} \notin \{y_1, \dots, y_n\} | y_1, \dots, y_n \text{ distinct})}_{\geq 1 - 0_i} \cdot \mathbb{P}(y_1, \dots, y_k \text{ distinct})$.

Iterate the argument to obtain

$$\mathbb{P}(y_1, \dots, y_c \text{ distinct}) \geq \prod_{i=1}^c (1 - p_i).$$

Taking logarithms shows

$$\begin{aligned} \log \left(\prod_{i=1}^c (1 - p_i) \right) &= \sum_{i=1}^c \log(1 - p_i) \stackrel{(p_i \leq \frac{1}{2})}{\geq} -\frac{5}{4} \sum_{i=1}^c p_i \\ &\geq -\frac{5}{4} \frac{c(c+1)}{2^{n+1}} \\ \implies \mathbb{P}(y_1, \dots, y_c \text{ distinct}) &\geq e^{-\frac{5}{4} \frac{c(c+1)}{2^{n+1}}}. \end{aligned}$$

Choosing $c+1 = \delta 2^{n/2}$ satisfies $c(c+1) \leq 2^n$ and $\mathbb{P}(y_1, \dots, y_c \text{ distinct}) \geq \exp(-(5/4)\delta^2)$. \square

Remark 28. For some $1 - p \leq \xi \leq p$,

$$\log(1 - p) = 0 + \frac{1}{1}(-p) - \frac{1}{2\xi^2}(-p)^2 \geq -p - \frac{p^2}{2} = -p \left(1 + \frac{p}{2}\right) \geq -\frac{5}{4}p.$$

3.3. **The quantum Fourier Transform.**

3.3.1. *Discrete FT*. For x_0, \dots, x_{n-1} define

$$\hat{x}_k := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp\left(-\frac{2\pi i}{N} k j\right)$$

with inverse

$$x_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \hat{x}_j \exp\left(\frac{2\pi i}{N} k j\right).$$

In matrix notation we get

$$\begin{aligned} \hat{X} &= F_N X \quad \text{with } F_N \in \mathbb{R}^{N \times N} \\ (F_N)_{kj} &:= \frac{1}{\sqrt{N}} \exp\left(-\frac{2\pi i}{N} k j\right), \end{aligned}$$

whereby F_N is a unitary matrix.

3.3.2. *Fast FT*. Assuming $N = 2^n$, we see that

$$\hat{x}_k = \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{N/2}} \sum_{j=0, j \text{ even}}^{N-1} x_j e^{-\frac{2\pi i}{N/2} k \frac{j}{2}} + e^{-\frac{2\pi i}{N} k} \frac{1}{\sqrt{N/2}} \sum_{j \text{ odd}} e^{-\frac{2\pi i}{N/2} k \frac{j}{2}} \right).$$

This splitting of FT into two FT s of half size leads to $\mathcal{O}(N \log N)$ complexity.

The QFT maps a state $|x\rangle = \sum_{j=0}^{N-1} \hat{x}_j |j\rangle$ to $|y\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$ where x_j is given by the classical FT (this is the convention, everything works with FT).

If $|x\rangle = |j_0\rangle$ then $x_j = \delta_{jj_0}$ with

$$\hat{x}_k = \frac{1}{\sqrt{N}} \exp\left(\frac{2\pi i}{N} k j_0\right)$$

and hence

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(\frac{2\pi i}{N} k j\right) = F_N |j\rangle.$$

To implement F_N efficiently, we first observe that for $k = k_1 2^{n-1} + k_2 2^{n-2} + \dots + k_n$,

$$\frac{k}{N} = \frac{k}{2^n} = \sum_{l=1}^n k_l 2^{-l}.$$

Therefore

$$\begin{aligned} F_N |j\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k \in \{0,1\}^n} \underbrace{\exp\left(2\pi i j \sum_{l=1}^n k_l 2^{-l}\right)}_{\bigotimes_{l=1}^n \exp(-2\pi i j k_l 2^{-l})} |k_1 \dots k_n\rangle \\ &= \bigotimes_{l=1}^n \left(|0\rangle + \exp\left(\frac{2\pi i j}{2^l}\right) |1\rangle \right) \frac{1}{\sqrt{2}}. \end{aligned}$$

Furthermore note that because

$$\exp\left(2\pi i j / 2^l\right) = \exp\left(2\pi i \sum_{m=1}^{n-l} j_m 2^{n-m-l}\right) \exp\left(2\pi i \sum_{m=n-l+1}^n j_m 2^{n-m-l}\right),$$

the first $n - l$ significant bits of j do not matter.
To implement

$$\frac{1}{\sqrt{2}} \left(|0\rangle + \exp \left(2\pi i \sum_{m=n-l+1}^n j_m 2^{n-m-l} \right) |1\rangle \right)$$

we use the $\boxed{R_s}$ gate shown in Figure 9, given by the matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & \exp \left(\frac{2\pi i}{2} \right) \end{pmatrix}$$

(or \boxed{P} on IBM q-composer).

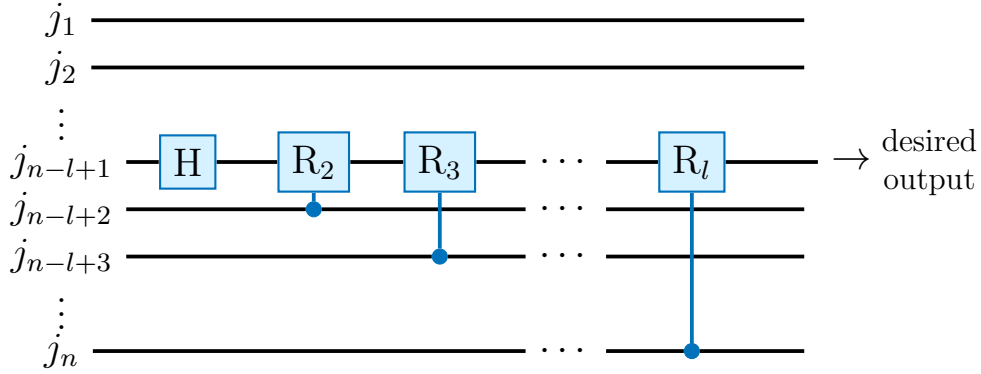


FIGURE 9. $\boxed{R_s}$ circuit diagram

3.3.3. Complexity. The *QFT* needs n qubits and $\mathcal{O}(n)$ gates per qubit, which adds up to $\mathcal{O}(n^2)$ gates. This is an exponential speed-up over *FFT* with $\mathcal{O}(n2^n)$ operations.

Remark 29. *Strictly speaking, QFT does something different than FT. The state*

$$QFT(|x\rangle) = \sum_{j=0}^{N-1} x_j |j\rangle$$

can only be accessed via measurement and hence will collapse to some $|j'\rangle$ with a certain probability. We will see that this is still very useful.

Remark 30. $\boxed{R_s}$ gates don't do very much for large s . One can show that $\mathcal{O}(n \log n)$ gates suffice if one accepts a small error probability.

Remark 31. *Reversing the order of the gates and using the adjoint gates gives an efficient implementation of the inverse QFT F_N^{-1} .*

3.4. Application Phase Estimation. Suppose we have a unitary operator $U : V \rightarrow V$, $\dim V = 2^n$, with eigenvector ψ , i.e.,

$$U\psi = e^{2\pi i \Phi} \psi \quad \text{for some } \Phi \in [0, 1).$$

Assume that $\Phi = \sum_{j=1}^n \Phi_j 2^{-j}$ can be written with n bits.

Since U is an operator on a 2^n dimensional Hilbert space, classical computation of $U\Psi \cdot \Psi$ requires at least $\mathcal{O}(2^n)$ operations.

3.4.1. Quantum Algortihm.

Algorithm 3.

- 1: Start with $|0\rangle|\Psi\rangle$
- 2: Apply $H^{\otimes n}$ to the first n qubits to obtain

$$\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle |\Psi\rangle$$

- 3: Apply $|j\rangle|\Psi\rangle \mapsto |j\rangle U^j |\Psi\rangle$ to obtain

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j \Phi} |j\rangle |\Psi\rangle$$

Note that the first n qubits satisfy

$$\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j \Phi} |j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j N \Phi / N} |j\rangle = F_N(|N\Phi\rangle)$$

This might be a bit confusing since suddenly $\Phi \in \mathbb{R}$ is interpreted as an element of the vector space in which Ψ is contained. However, $N\Phi = \sum_{j=1}^n \Phi_j 2^{n-j}$ is a basis element and hence makes sense.

- 4: Apply QFT to the first n qubits to obtain

$$F_N^{-1} \left(\frac{1}{\sqrt{N}} \sum e^{2\pi i j \Phi} |j\rangle \right) = |N\Phi\rangle$$

- 5: Measure in computational basis to obtain $N\Phi$ and hence Φ .

Remark 32. The circuit diagram implementing Line 4 is shown in Figure 10 below.

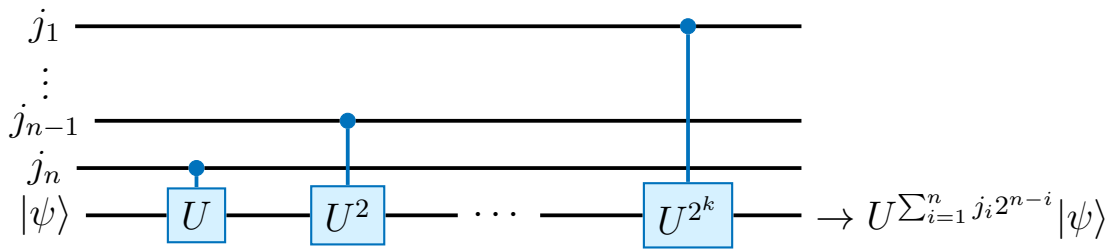


FIGURE 10. Circuit diagram for Line 4

However, we will need to assume that $\boxed{U^{2^k}}$ can be implemented efficiently. This might not be true in general, but is in the applications below.

Remark 33. Note that the input doesn't need to be a single eigenvector. Let $|\Psi\rangle, |\Psi'\rangle$ be two eigenvectors with phase Φ, Φ' respectively. Linearity implies that the input $1/\sqrt{2}(|\Psi\rangle + |\Psi'\rangle)$ will produce the output $1/\sqrt{2}(|N\Phi\rangle + |N\Phi'\rangle)$. A measurement will produce either Φ or Φ' with equal probability.

If Φ requires more than n bits, the final state is a small perturbation δ of $F_N(|N\Phi_n\rangle)$ with

$$|\delta| = \left| \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \underbrace{[e^{2\pi i j \Phi} - e^{2\pi i j \Phi_n}]}_{\mathcal{O}(2^{-n})} |j\rangle \right| / \left| \frac{1}{\sqrt{N}} \sum e^{2\pi i j \Phi} |j\rangle \right| \simeq \mathcal{O}(2^{-n}).$$

3.5. Shor's integer factorization.

3.5.1. *Problem.* Given $n \in \mathbb{N} \setminus \{\text{primes}\}$, find $1 < k < n$ with $n/k \in \mathbb{N}$.

Remark 34.

- $n \in \mathbb{N}$ can be defined using $\log_2(n)$ bits.
Polynomial complexity of algorithms means $\mathcal{O}(\log_2(n)^p)$ operations.
- There exist efficient classical algorithms to check whether n is prime or power of prime (see, e.g. AKS-test or BPSW-test).
- The security of many crypto-systems is based on the fact that integer factorization is hard.
- The best known classical algorithm is the general number field sieve with complexity $\mathcal{O}(\exp((\log_2 n)^{1/3}(\log_2 \log_2 n)^{2/3}))$.
Note that the run time is a conjecture.
- It is not known that classical algorithms can not be faster. Latest paper which (falsely) claimed an $\mathcal{O}(\log n^p)$ -algorithm was by Schnour (Bonn) in 2021.

3.5.2. *Reduction to period finding.* Choose $x \in \{2, \dots, N-1\}$ s.t. $\gcd(x, N) = 1$, then the factorization of $N \in \mathbb{N}$ can be reduced to finding the period of x (Lemma 35). Consider $x \in (\mathbb{Z}/N\mathbb{Z})$ which is a multiplicative group mod n . Thus, by Lemma 35, x has period r with $x^r \bmod N = 1$. Furthermore, r is even, $x^{r/2} \pm 1 \not\equiv 0 \bmod N$ with probability $1/2$ and $(x^{r/2} - 1)(x^{r/2} + 1) \equiv x^r - 1 \equiv 0 \bmod N$. We have thus found $\gcd(x^{r/2} - 1, N)$ and $\gcd(x^{r/2} + 1, N)$ which are non-trivial factors of N .

Lemma 35. Every $x \in (\mathbb{Z}/N\mathbb{Z})$ has a period r which is minimal for $r \in \mathbb{N}$ with $x^r \bmod N = 1$.

Proof. Consider the set $S = \{1, x^1, \dots\} \subset (\mathbb{Z}/N\mathbb{Z})^+$. Since $(\mathbb{Z}/N\mathbb{Z})^+$ is finite, there exists $j \neq k \in \mathbb{N}$ with $x^j = x^k \bmod N$. W.l.o.g. $j > k$.

Then $x^{k-j} \equiv 1 \bmod N$. □

The Euler totient function ϕ is given by $\phi(n) := \#\{1 \leq k \leq n \mid \gcd(k, n) = 1\}$ and

$$\phi(n) = n \prod_{\substack{p|n \\ p \text{ prime}}} \left(1 - \frac{1}{p}\right) \implies \phi(p^\alpha) = p^\alpha \left(1 - \frac{1}{p}\right) = p^{\alpha-1}(p-1).$$

Lemma 36. *Chinese remainder Theorem* Let $m_1, \dots, m_n \in \mathbb{N}$ with $\gcd(m_i, m_j) = 1$ for $i \neq j$. Then the problem: Find x s.t.

$$\begin{aligned} x &= a_1 \bmod m_1 \\ &\vdots \\ x &= a_n \bmod m_n \end{aligned}$$

is solvable and any two solutions are equal mod $M = m_1 m_2 \dots m_n$.

Proof. Define $M_i := M/m_i$, then $\gcd(M_i, m_i) = 1$ and M_i has a multiplicative inverse mod m_i denoted by N_i .

$x = \sum_i a_i M_i N_i$ is a solution since $M_i N_i \equiv 1 \pmod{m_i}$ and $M_i N_i \equiv 0 \pmod{m_j}$ for $i \neq j$.

Two solutions x, x' satisfy $x - x' \equiv 0 \pmod{m_i}$ for all i and hence $M = m_1 m_2 \dots m_n$ divides $x - x'$ since m_i are coprime. \square

Lemma 37. Let $p > 2$ be prime, let 2^d be the maximal power of 2 dividing $\phi(p^\alpha)$, and let x denote a randomly chosen element in $(\mathbb{Z}/p^\alpha \mathbb{Z})^*$. Then

$$\mathbb{P}(2^d \text{ divides order of } x) = \frac{1}{2}.$$

Proof. If $\phi(p^\alpha) = p^{\alpha-1}(p-1)$ is even then $2^d \geq 1$.

Let g denote generator of $(\mathbb{Z}/p^\alpha \mathbb{Z})^*$. Then $x \equiv g^k \pmod{p^\alpha}$ for some $k \in \{1, \dots, \phi(p^\alpha)\}$.

Let r denote order of x . Then we have for

- k odd: $g^{kr} \equiv 1 \pmod{p^\alpha} \implies \phi(p^\alpha) | kr$ since $\phi(p^\alpha)$ is minimal with $g^{\phi(p^\alpha)} = 1$. For Furthermore, since k is odd, $2^d | r$.
- k even: $g^{k\phi(p^\alpha)/2} \equiv 1 \pmod{p^\alpha} \implies r | \phi(p^\alpha)/2$ since r is minimal with $x^r \equiv g^{kr} \equiv 1 \pmod{p^\alpha}$.

Hence $2^d \nmid r$ and

thus, for exactly half of $x \in (\mathbb{Z}/p^\alpha \mathbb{Z})^*$, we have $x = g^k$ with k even/odd. \square

Lemma 38. Let $N = p_1^{\alpha_1} \dots p_m^{\alpha_m}$ be a prime factorization of N , odd. Let x be randomly chosen in $(\mathbb{Z}/N\mathbb{Z})^*$ with order $r \pmod{N}$. Then

$$\mathbb{P}(r \text{ is even and } x^{\frac{r}{2}} \not\equiv -1 \pmod{N}) \geq 1 - 2^{-m+1}.$$

Proof. Following Lemma 36, choosing x randomly is equivalent to choosing x_i randomly in $(\mathbb{Z}/p_j^{\alpha_j} \mathbb{Z})^*$ with $x = x_j \pmod{p_i^{\alpha_i}}$ for $i = j, \dots, m$ since $x \iff (x_1, \dots, x_m)$ is one-to-one.

Let r_j be the order of $x_j \pmod{p_j^{\alpha_j}}$ and let 2^{d_j} be the maximal power of 2 dividing r_j . We will show that

$$r \text{ is odd or } x^{\frac{r}{2}} \equiv -1 \pmod{N} \implies d = d_2 = \dots = d_m.$$

If this is the case, the following argument concludes the proof. Let d'_j be maximal s.t. $2^{d'_j}$ divides $\phi(p_j^{\alpha_j})$. Lemma 37 states that

$$\mathbb{P}(d_j = d'_j) = \frac{1}{2} \implies \mathbb{P}(d_j = k) \leq \frac{1}{2} \quad \forall k \in \mathbb{N}.$$

Furthermore, since d_j are independent, we have

$$\mathbb{P}(d_1 = \dots = d_m) \leq 2^{-m+1}.$$

It remains to show that $d_1 = \dots = d_m$.

Case 1: r is odd and $x^r \equiv 1 \pmod{N}$. Then

$$x^r \equiv 1 \pmod{p_j^{\alpha_j}} \implies r_j | r \implies r_j \text{ is odd} \implies d_j = 0.$$

Case 2: r is even and $x^{r/2} \equiv -1 \pmod{N}$. Then

$$x^{\frac{r}{2}} \equiv -1 \pmod{p_j^{\alpha_j}}.$$

If r_j would divide $r/2$, we would have

$$\left. \begin{array}{l} x^{\frac{r}{2}} \equiv x^{kr_j} \equiv -1 \pmod{p_j^{\alpha_j}} \\ x^{kr_j} \equiv x_j^{kr_j} \pmod{p_j^{\alpha_j}} \\ x^{kr_j} \equiv 1 \pmod{p_j^{\alpha_j}} \end{array} \right\} \implies r_j \nmid \frac{r}{2}$$

but r_j divides r . Hence largest power of 2 dividing r must be equal to d_j . \square

Note that if r is period of $x \bmod N$ and r is even, then $x^{r/2} \not\equiv 1 \bmod N$ with

$$\mathbb{P}(r \text{ even}, x^{r/2} \not\equiv \pm 1 \bmod N) \geq 1 - 2^{-m+1} \geq \frac{1}{2}.$$

If $m \geq 2$, i.e., if N is not a prime or power of prime, define $U|y\rangle := |xy \bmod N\rangle$. Note that $|j\rangle \mapsto |xj \bmod N\rangle$ is bijective since $xj = xj' \bmod N$ and $x(j - j') = 0 \bmod N$ and thus $\gcd(x, N) \neq 1$. Therefore U is a permutation of basis elements and hence unitary.

Lemma 39. *Let r denote the period of $x \in (\mathbb{Z}/N\mathbb{Z})^*$. Then,*

$$|U_s\rangle := \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-2\pi i s k}{r}\right) |x^k \bmod N\rangle$$

are eigenvectors of U with eigenvalues

$$\lambda_s := \exp\left(\frac{2\pi i s}{r}\right), \quad s = 0, \dots, r-1.$$

Proof. Note that $x^r \equiv x^0 \bmod N$ and $\exp(-2\pi i s) \equiv \exp(0)$. Hence

$$\begin{aligned} U|U_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-2\pi i s k}{r}\right) |x^{k+1} \bmod N\rangle \\ &= \lambda_s \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-2\pi i s (k+1)}{r}\right) |x^{k+1} \bmod N\rangle \\ &= \lambda_s |U_s\rangle. \end{aligned}$$

\square

Now, since

$$\begin{aligned} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |U_s\rangle &= \frac{1}{r} \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} \exp\left(\frac{-2\pi i s k}{r}\right) |x^k \bmod N\rangle \\ &= \frac{1}{r} \sum_{k=0}^{r-1} \underbrace{\sum_{s=0}^{r-1} \exp\left(\frac{-2\pi i s k}{r}\right)}_{\begin{cases} 0 & k \neq 0 \\ r & k = 0 \end{cases}} |x^k \bmod N\rangle \\ &= |x^0 \bmod N\rangle = |1\rangle, \end{aligned}$$

quantum phase estimation with input $|1\rangle$ produces the state

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |2^n \frac{s}{r}\rangle.$$

Because $r \leq N$, s/r can be exactly represented with n bits.

To efficiently implement

$$U^{2^k} |y\rangle = |xy^{2^k} \bmod N\rangle$$

we use the fact that

$$y^{2^k} = \left(\cdots \left((y^2)^2 \right) \cdots \right)^2.$$

Hence, $|xy^{2^k} \bmod N\rangle$ requires 1 multiplication and k squares mod N . It can be implemented as in classical circuits with XOR & AND gates.

3.6. Shor's period finding algorithm. Note that this is the only quantum part of Shor's algorithm. Let $N \leq 2^n$.

Algorithm 4 (Shor's period finding algorithm).

1: Prepare $|\Psi\rangle = |1\rangle = |\underbrace{0 \dots 0}_n 1\rangle$ and U as before

2: Use quantum phase estimation with U and $|\Psi\rangle$ to obtain the state

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |2^n \frac{s}{r}\rangle$$

3: Measurement gives a random number s/r from the set

$$\left\{ \frac{1}{r}, \dots, \frac{r-1}{r} \right\} \subseteq \left\{ \frac{i}{2^n} : i = 0, \dots, 2^n - 1 \right\}$$

4: Write $s/r = s_0/r_0$ with $\gcd(s_0, r_0) = 1$. If s and r were coprime already, we have $r = r_0$ and found the period. Otherwise, repeat.

Remark 40. There are $\mathcal{O}\left(\frac{r}{\log \log r}\right)$ Euler totient function numbers $1 \leq s < r$ with $\gcd(s, r) = 1$. Therefore, the success-probability of Line 4 is $\mathcal{O}\left(\frac{1}{\log \log r}\right)$. Furthermore, since

$$\frac{1}{\log \log r} \geq \frac{1}{\log \log N} \geq \frac{1}{\log n},$$

Line 4 requires on average $\log n$ repetitions.

Remark 41. Implicitly, we used QFT to find the frequency of $x^k \bmod N$, i.e., the period x .

3.7. Grover's algorithm.

3.7.1. *Problem.* Given $\mathcal{F}\{0,1\}^n \rightarrow \{0,1\}$, find $x \in \{0,1\}^n$ with $\mathcal{F}(x) = 1$ or determine that $\mathcal{F} = 0$.

Recall the query

$$\boxed{Q_{\mathcal{F}}}(|i-\rangle) = (-1)^{\mathcal{F}(|i\rangle)}|i-\rangle$$

where we identify $x \in \{0,1\}^n$ with $i = 0, \dots, 2^n - 1$ and define the *Grover diffusion operator*

$$U_s := 2 \underbrace{|s\rangle\langle s|}_{\text{projection onto } |s\rangle} - \text{id}$$

where $|s\rangle$ denotes the uniform state

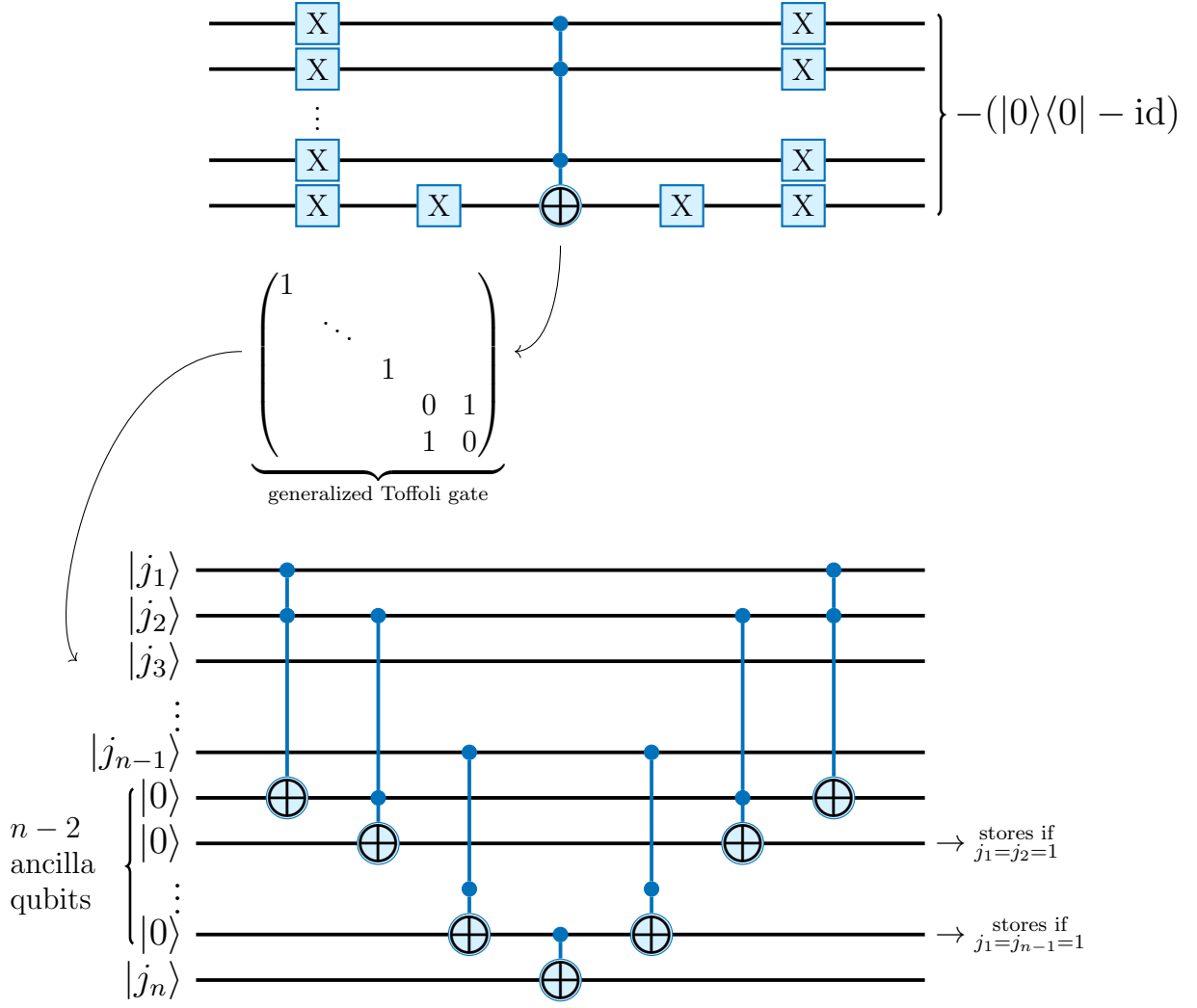
$$|s\rangle := \frac{1}{N} \sum_{j=0}^{N-1} |j\rangle.$$

Note that $|s\rangle = \boxed{\text{H}}^{\otimes n} |0^n\rangle$ and hence

$$U_s = \boxed{\text{H}}^{\otimes n} (2|0^n\rangle\langle 0^n| - \text{id}) \boxed{\text{H}}^{\otimes n},$$

where $(2|0^n\rangle\langle 0^n| - \text{id})$ corresponds to the matrix

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & -1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & -1 \end{pmatrix} \in \mathbb{R}^{2^n \times 2^n}.$$



3.7.2. Algorithm.

Algorithm 5 (Grover's algorithm). 1: Apply $\boxed{\text{H}}^{\otimes n}$ to obtain

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle$$

2: For $k = 1, \dots, r(N)$ do

a: Apply $\boxed{Q_{\mathcal{F}}}$ to correct state $\sum_{j=0}^{N-1} \alpha_j |j\rangle$ to obtain

$$\sum_{j=0}^{N-1} (-1)^{\mathcal{F}(j)} \alpha_j |j\rangle$$

b: Apply U_s to first n qubits

3: Measure in computational basis

Theorem 42. Let

$$r(N) = \frac{\arccos\left(\sqrt{\frac{t}{N}}\right)}{2 \arcsin\left(\sqrt{\frac{t}{N}}\right)},$$

then Grover's algorithm finds state $|x\rangle$ with $\mathcal{F}(x) = 1$ with probability at least $1 - \frac{t}{N}$.

Proof. Define

$$|B\rangle := \frac{1}{\sqrt{N-t}} \sum_{\mathcal{F}(|j\rangle)=0} |j\rangle,$$

where $t = \#\mathcal{F}^{-1}(\{1\})$. Observe that the first n qubits of $|x\rangle \mapsto \boxed{\mathbf{Q}_{\mathcal{F}}}(|x\rangle)$ satisfy

$$|x\rangle \mapsto (2|B\rangle\langle B| - \text{id})|x\rangle = U_B(|x\rangle),$$

which can be checked for basis elements $|x\rangle = |j\rangle$.

Define $|\zeta\rangle := 1/\sqrt{t} \sum_{\mathcal{F}(|j\rangle)=1} |j\rangle$ and note that $|s\rangle \in \text{span}\{|\zeta\rangle, |B\rangle\}$, and $U_s, U_B : \text{span}\{|\zeta\rangle, |B\rangle\}$. Hence, the Grover iteration never leaves the plane $\text{span}\{|\zeta\rangle, |B\rangle\}$ as depicted in Figure 11.

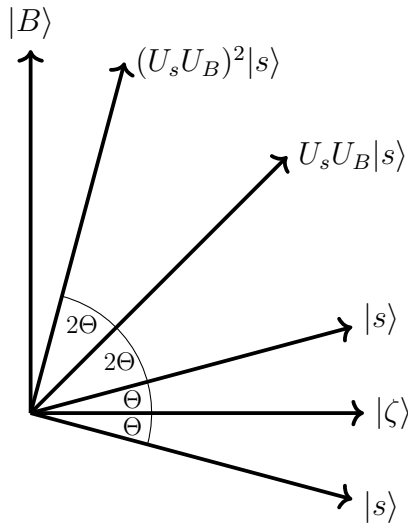


FIGURE 11. Grover iterates in the $|\zeta\rangle, |B\rangle$ -plane

Each application of $U_s U_B$ rotates a state $|x\rangle \in \text{span}\{|\zeta\rangle, |B\rangle\}$ towards $|\zeta\rangle$ by an angle Θ given by

$$\cos \Theta = \frac{\langle s|B\rangle}{\|s\|\|B\|} = \frac{1}{\sqrt{N}} \frac{1}{\sqrt{N-t}} \sum_{\mathcal{F}(|j\rangle)=0} \langle j|j\rangle = \frac{N-t}{\sqrt{N(N-t)}} = \sqrt{1 - \frac{t}{N}}.$$

Furthermore, using the trigonometric identity shows that

$$\sin^2 \theta = 1 - \cos^2 \theta = 1 - 1 + \frac{t}{N} \implies \sin \theta = \sqrt{\frac{t}{N}}.$$

For large N , we have $\sin \theta \approx \sqrt{t/N}$ and $\cos \theta \approx \sqrt{1 - (t/N)}$. Since the angle between $|\zeta\rangle$ and $|s\rangle$ is

$$\cos \alpha = \frac{1}{\sqrt{N}} \frac{1}{\sqrt{t}} \sum_{\mathcal{F}(|j\rangle)=1} \langle j|j\rangle = \sqrt{\frac{t}{N}},$$

$$r(N) = \text{round} \left(\frac{\arccos \left(\sqrt{\frac{t}{N}} \right)}{2 \arcsin \left(\sqrt{\frac{t}{N}} \right)} \right)$$

to obtain a state $|x\rangle$ with $\langle \zeta | x \rangle \geq \cos \theta$ and $|\langle \zeta | x \rangle|^2 \geq \cos^2 \theta = 1 - (t/N)$. \square

Remark 43. If $t = 1$, a classical algorithm requires at least N evaluations of \mathcal{F} . Grover's algorithm only needs $\mathcal{O}(\sqrt{N})$ iterations with $\mathcal{O}(n)$ gates.

3.7.3. Optimality of Grover's algorithm.

Lemma 44. Any quantum algorithm based on the query $\boxed{Q_{\mathcal{F}}}$ requires at least $\mathcal{O}(\delta\sqrt{N})$ applications of $\boxed{Q_{\mathcal{F}}}$ to succeed with probability at least δ^2 .

Proof. Any quantum algorithm starts with some state $|\psi\rangle$ and applies unitary transformations as well as $\boxed{Q_{\mathcal{F}}}$. I.e., we may write the state after k applications of $\boxed{Q_{\mathcal{F}}}$ as

$$|\psi_k^{\mathcal{F}}\rangle = U_k Q_{\mathcal{F}} U_{k-1} Q_{\mathcal{F}} \cdots U_1 Q_{\mathcal{F}} |\psi\rangle,$$

where U_k, \dots, U_1 are unitary and $Q_{\mathcal{F}}$ is the query.

Additionally, we will consider

$$|\psi_k\rangle := U_k U_{k-1} \cdots U_1 |\psi\rangle.$$

Let $\mathcal{F}_j : \{0, 1\}^n \rightarrow \{0, 1\}$ with $\mathcal{F}_j(|j\rangle) = \delta_{ij}$ and define

$$D_k = \sum_{j=0}^{N-1} \|\psi_k^{\mathcal{F}_j} - \psi_k\|^2.$$

Idea: If D_k is small, the evaluation of \mathcal{F} does not make a big difference and it will be hard to find $\mathcal{F}(|i\rangle) = 1$.

Step 1: Show that $D_k \leq 4k^2$ by induction.

$k = 0$: $D_0 = 0$.

$k \mapsto k + 1$:

$$\begin{aligned} D_{k+1} &= \sum_{j=0}^{N-1} \|U_{k+1} Q_{\mathcal{F}_j} \psi_k^{\mathcal{F}_j} - U_{k+1} \psi_k\|^2 \\ &= \sum_{j=0}^{N-1} \|Q_{\mathcal{F}_j} \psi_k^{\mathcal{F}_j} - \psi_k\|^2 \\ &= \sum_{j=0}^{N-1} \|Q_{\mathcal{F}_j} (\psi_k^{\mathcal{F}_j} - \psi_k) + (Q_{\mathcal{F}_j} - \text{id}) \psi_k\|^2 \end{aligned}$$

Note that $Q_{\mathcal{F}_j} = \text{id} - 2|j\rangle\langle j| \implies (Q_{\mathcal{F}_j} - \text{id})\psi_k = -2|j\rangle\langle j|\psi_k$ and thus

$$D_k + 1 \leq \sum_{j=0}^{N-1} \|\psi_k^{\mathcal{F}_j} - \psi_k\|^2 + 4\|\psi_k^{\mathcal{F}_j} - \psi_k\|^2 |\langle j|\psi_k\rangle| + 4|\langle j|\psi_k\rangle|^2.$$

Furthermore, Cauchy-Schwarz shows that

$$\begin{aligned} D_{k+1} &\leq D_k + 4 \left(\sum_{j=0}^{N-1} \|\psi_k^{\mathcal{F}_j} - \psi_k\|^2 \right)^{\frac{1}{2}} \underbrace{\left(\sum_{j=0}^{N-1} |\langle j|\psi_k\rangle|^2 \right)^{\frac{1}{2}}}_{=1} + 4 \underbrace{\langle \psi_k | \psi_k \rangle}_{=1} \\ &\leq D_k + 4\sqrt{D_k} + 4 \end{aligned}$$

and using the induction hypotheses, $D_k + 4\sqrt{D_k} + 4 \leq 4k^2 + 8k + 4 = 4(k+1)^2$.

This concludes the induction and shows that $D_k \leq 4k^2$.

Step 2: Assume that $|\langle j | \psi_k^{\mathcal{F}_j} \rangle| \geq \delta^2 > 0 \quad \forall j = 1, \dots, N-1$, i.e. the algorithm works with high probability for each input. Replacing $|j\rangle$ with $\exp(i\theta)|j\rangle$ does not change the success probability. Hence, we may assume that $|\langle j | \psi_k^{\mathcal{F}_j} \rangle| = \langle j | \psi_k^{\mathcal{F}_j} \rangle$ and thus

$$\|\psi_k^{\mathcal{F}_j} - j\|^2 = 2 - 2|\langle j | \psi_k^{\mathcal{F}_j} \rangle| \leq 2 - (1 - \delta).$$

By defining

$$E_k := \sum_{j=0}^{N-1} \|\psi_k^{\mathcal{F}_j} - j\|^2 \implies E_k \leq 2N(1 - \delta)$$

$$\mathcal{F}_k := \sum_{j=0}^{N-1} \|j - \psi_k\|^2$$

we see that

$$\begin{aligned} D_k &= \sum_{j=0}^{N-1} \|(\psi_k^{\mathcal{F}_j} - j) + (j - \psi_k)\|^2 \\ &\geq \sum_{j=0}^{N-1} \|(\psi_k^{\mathcal{F}_j} - j)\|^2 - \|(\psi_k^{\mathcal{F}_j} - j)\| \|(j - \psi_k)\| - \|(j - \psi_k)\|^2 \\ &= E_k + \mathcal{F}_k - 2 \left(\sum_{j=0}^{N-1} \|\psi_k^{\mathcal{F}_j} - j\|^2 \right)^{\frac{1}{2}} \left(\sum_{j=0}^{N-1} \|j - \psi_k\|^2 \right)^{\frac{1}{2}} \\ &= E_k + \mathcal{F}_k - 2\sqrt{E_k \mathcal{F}_k} = \left(\sqrt{E_k} - \sqrt{\mathcal{F}_k} \right)^2. \end{aligned}$$

Note that any state $|\Phi\rangle$ satisfies

$$\begin{aligned} \sum_{j=0}^{N-1} \|\Phi - j\|^2 &= \sum_{j=0}^{N-1} \|\Phi\|^2 - 2\langle \Phi | j \rangle + \underbrace{\|j\|^2}_{=1} \\ &\geq 2N - 2 \underbrace{\sqrt{\sum_{j=0}^{N-1} 1} \sqrt{\sum_{j=0}^{N-1} |\langle \Phi | j \rangle|^2}}_{=1} = 2(N - \sqrt{N}). \end{aligned}$$

This implies $\mathcal{F}_k \geq 2(N - \sqrt{N})$ and hence $\mathcal{F}_k \geq E_k$ for sufficiently large N and

$$\begin{aligned} \sqrt{\mathcal{F}_k} - \sqrt{E_k} &\geq \sqrt{2(N - \sqrt{N})} - \sqrt{2N(1 - \delta)} = \frac{2\delta N - 2\sqrt{N}}{\sqrt{2(N - \sqrt{N})} + \sqrt{2N(1 - \delta)}} \\ &\geq \frac{2\delta N - 2\sqrt{N}}{2\sqrt{2N}} = \frac{\delta}{\sqrt{2}}\sqrt{N} - \frac{1}{\sqrt{2}}. \end{aligned}$$

Therefore, $D_k \geq (\sqrt{\mathcal{F}_k} - \sqrt{E_k})^2 \approx \delta^2 N$ and since $D_k \leq 4k^2$, we have $k \approx \delta\sqrt{N}$. \square

Example 45. *Quantum composer example* The link to this quantum composer example can be found on the web page for this lecture. Let $N = 16$, $n = 4$,

$$\mathcal{F}(x) = \mathcal{F}(x_0, \dots, x_3) = x_0 \text{ AND } x_1 \text{ AND } x_2 \text{ AND } x_3 = \delta_{x=(1,1,1,1)}$$

and $t = 1$.

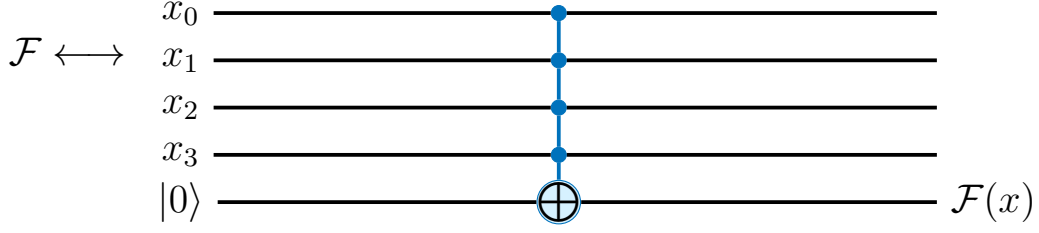


FIGURE 12. \mathcal{F} as a quantum circuit

Since

$$\frac{\arccos\left(\sqrt{\frac{1}{16}}\right)}{2 \arcsin\left(\sqrt{\frac{1}{16}}\right)} \approx 2.6083,$$

the optimal $r(N)$ is 3 and the probability of success is therefore at least $1 - 1/N = 15/16 \approx 0.9375$.

3.8. Numerical Quadrature.

3.8.1. *Problem.* Given $f : \{0, 1\}^n \rightarrow [-1, 1]$, compute $1/N \sum_{i=0}^{N-1} f(i)$.

3.8.2. *Quantum super sampling.* Consider the Oracle Q_f defined by

$$Q_f : |0\rangle \otimes |i\rangle \mapsto \left[\sqrt{1 - f(i)}|0\rangle + \sqrt{f(i)}|1\rangle \right] \otimes |i\rangle.$$

Algorithm 6. 1: Start with $|0^P\rangle \otimes |0\rangle \otimes |0^n\rangle$

2: Apply $\boxed{\mathbf{H}}^{\otimes P} \otimes \text{id} \otimes \boxed{\mathbf{H}}^{\otimes n}$ to obtain

$$\frac{1}{\sqrt{PN}} \sum_{i=0}^{N-1} \sum_{m=0}^{P-1} |m\rangle \otimes |0\rangle \otimes |i\rangle$$

3: Apply $\boxed{Q_{\mathcal{F}}}$ to obtain

$$\frac{1}{\sqrt{PN}} \sum_{i=0}^{N-1} \sum_{m=0}^{P-1} |m\rangle \otimes \left[\sqrt{1 - f(i)}|0\rangle + \sqrt{f(i)}|1\rangle \right]$$

4: Define

$$\begin{aligned} |G\rangle &:= \sqrt{\frac{\sum f(i)}{N}} \sum_{i=0}^{N-1} \sqrt{f(i)}|1\rangle \otimes |i\rangle \\ |B\rangle &:= \sqrt{\frac{\sum 1 - f(i)}{N}} \sum_{i=0}^{N-1} \sqrt{1 - f(i)}|0\rangle \otimes |i\rangle \\ \bar{f} &:= \frac{1}{N} \sum f(i) \end{aligned}$$

Note that

$$\begin{aligned} U_s &:= (1|0\rangle\langle 0| - \text{id}) \otimes \text{id} \\ U_s|B\rangle &= |B\rangle \\ U_s|\zeta\rangle &= -|G\rangle. \end{aligned}$$

Furthermore,

$$U_\psi = (2|\psi\rangle\langle\psi| - \text{id})$$

with $|\psi\rangle := \sqrt{1-\bar{f}}|B\rangle + \sqrt{\bar{f}}|\zeta\rangle$. Lines 1 to 3 provide a quantum circuit to implement the map $U : |0\rangle \oplus |0^n\rangle \mapsto |\psi\rangle$, and hence

$$U_\psi = U^{-1}(2|0^{n+1}\rangle\langle 0^{n+1}| - \text{id})U$$

as in Grover's algorithm.

5: For $\bar{f} := 1/N \sum f(i)$, the state reads

$$\frac{1}{\sqrt{P}} \sum_{m=0}^{P-1} |m\rangle \otimes \left[\sqrt{1-\bar{f}}|B\rangle + \sqrt{\bar{f}}|\zeta\rangle \right]$$

6: Apply Grover iteration $U_\psi U_s$ m -times to second $n+1$ qubits to obtain

$$\frac{1}{\sqrt{P}} \sum_{m=0}^{P-1} |m\rangle \otimes (U_\psi U_s)^n \left[\sqrt{1-\bar{f}}|B\rangle + \sqrt{\bar{f}}|\zeta\rangle \right]$$

Explanation: Let $\sin \theta = \sqrt{\bar{f}}$, then $|\psi\rangle = \cos \theta |B\rangle + \sin \theta |\zeta\rangle$ and $(U_\psi U_s)^m |\psi\rangle = \cos(2m+1)\theta |B\rangle + \sin(2m+1)\theta |\zeta\rangle$. U_s acts like a reflection across $|B\rangle$ as shown in Figure 13.

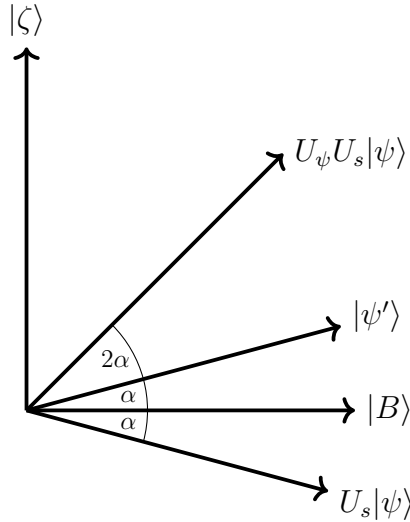


FIGURE 13. $U_\psi U_s$ and U_s acting on $|\psi\rangle$

Note that although the m -times application of $U_\psi U_s$ is as in Algorithm 3, the cost is in general $\mathcal{O}(Pm)$.

7: $|\psi\rangle$ rotates with rate 2θ in the $|B\rangle, |\zeta\rangle$ -plane. Use QFT to obtain the rate

$$\frac{1}{\sqrt{P}} \sum_{m=0}^{P-1} |m\rangle \otimes [\cos(2m+1)\theta|B\rangle + \sin(2m+1)\theta|\zeta\rangle].$$

Measure the last $n+1$ qubits in $\{|\zeta\rangle, |B\rangle\}$ -basis to obtain

$$\frac{1}{C} \sum_{m=0}^{P-1} \underbrace{\sin[(2m+1)\theta]}_{\hat{x}_m} |m\rangle$$

where $C = 1/P \sum_{m=0}^{P-1} \sin[(1m+1)\theta]^2$.

8: Under the assumption that $\theta = \pi\theta_0/P$, $\theta_0 \in k$, apply QFT to obtain

$$\frac{1}{C} \sum_{m=0}^{P-1} x_m |m\rangle,$$

where x_m is given by IFT of \hat{x}_m , i.e.

$$\begin{aligned} x_m &= \frac{1}{P} \sum_{k=0}^{P-1} \hat{x}_k e^{\frac{2\pi i m k}{P}} = \frac{1}{\sqrt{P}} \frac{P-1}{2} \frac{1}{2i} \begin{pmatrix} i(2k+1)\frac{\theta_0}{P} & -i(2k+1)\frac{\theta_0}{P} \\ e & -e \end{pmatrix} e^{\frac{2\pi i m k}{P}} \\ &= e^{i\frac{\theta_0}{P}} \frac{1}{2i} \begin{cases} \sqrt{P} & m = \theta_0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

9: Final measurement produces $|m\rangle = |\theta_0\rangle$ and hence $\bar{f} = \sin(\pi\theta_0/P)^2$.

Remark 46. The implementation of Q_f is not clear. If $f : \{0, 1\}^n \rightarrow \{0, 1\}$, we can use a controlled not-gate to implement the query as shown in Figure 14.

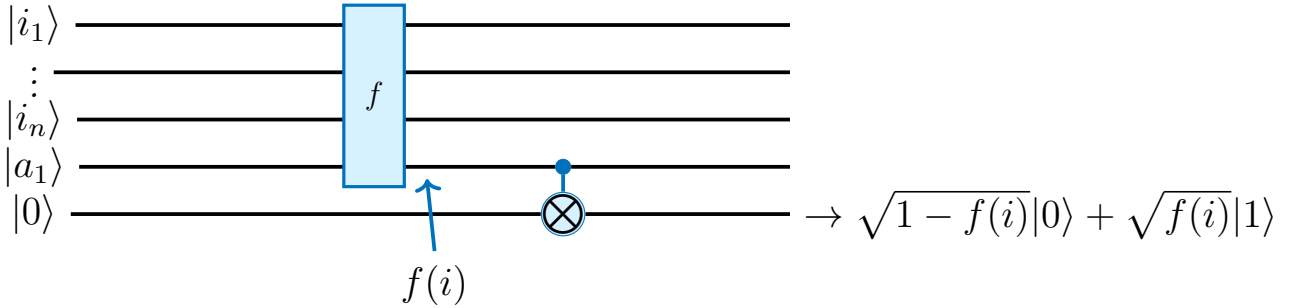


FIGURE 14. Circuit diagram for Q_f if $f : \{0, 1\}^n \rightarrow \{0, 1\}$

General integrals can always be split into R integration problems for precision 2^{-R} , i.e.,

$$\frac{1}{N} \sum_{i=0}^{N-1} f(i) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=1}^R 2^{-k} \underbrace{f_k(i)}_{\in \{0,1\}}.$$

4. LINEAR SYSTEMS OF EQUATIONS ON A QUANTUM COMPUTER

Let $A \in \mathbb{C}^{N \times N}$ be Hermitian and invertible and let $b \in \mathbb{C}^N$ for $N = 2^n$, $n \in \mathbb{N}$. The goal is to find $x \in \mathbb{C}^N$ with $Ax = b$.

If A is positive definite with sparsity s , we will use the classical conjugate gradient method with $\mathcal{O}(sNl)$ operations, where l is the number of iterations of the CG method.

Using the condition number $\kappa(A) = \|A\| \|A^{-1}\|$, we can estimate the error in the A -norm as

$$\|x - x_0\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^l \|x - x_0\|_A.$$

Given a relative error $\varepsilon_r \leq \varepsilon$ and defining

$$q := \left(1 - \frac{2}{\sqrt{\kappa} + 1} \right) \geq e^{-\frac{2}{\sqrt{\kappa}}},$$

it follows that

$$2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^l \leq \varepsilon$$

and thus

$$l \leq \frac{\log \frac{\varepsilon}{2}}{\log q} \leq \frac{1}{2} \sqrt{\kappa} \ln \frac{2}{\varepsilon}.$$

The computational complexity is therefore $\mathcal{O}(sN\sqrt{\kappa} \ln(2/\varepsilon))$.

The question arises whether an exponential speedup is possible on a quantum computer.

4.1. Quantum version of linear systems of equations. Let $A \in \mathbb{C}^{N \times N}$ be Hermitian with $\det A = 1$, $(b_i) = b \in \mathbb{C}^N$, and $(x_i) = x \in \mathbb{C}^N$ such that $Ax = b$ for some $N = 2^n$. Furthermore, let $|b\rangle$ be an n -qubit state given by

$$|b\rangle := \frac{\sum_i b_i |i\rangle}{\|\sum_i b_i |i\rangle\|} \quad (7)$$

and

$$|x\rangle := \frac{\sum_i x_i |i\rangle}{\|\sum_i x_i |i\rangle\|}. \quad (8)$$

The goal is to find $|\tilde{x}\rangle$ with $\| |x\rangle - |\tilde{x}\rangle \| \leq \varepsilon$ with probability $\Omega \geq 1/2$, for a given error $\varepsilon > 0$. This is sometimes (formal!) written as $A|x\rangle = |b\rangle$.

Remark 47.

- Normalization in Equations (7) and (8) is necessary to get a q -state.
- The quantum linear system is different from the linear system. The result is the state $|x\rangle$, not the vector x which solves $Ax = b$.
- If A is not Hermitian, consider

$$\tilde{A} = \begin{pmatrix} 0 & A^H \\ A & 0 \end{pmatrix} = |1\rangle\langle 0| \otimes A + |0\rangle\langle 1| \otimes A^H.$$

As we have doubled the dimension of the problem, we need 1 additional ancilla qubit. We solve $\tilde{A}|0x\rangle = |1b\rangle$.

- If $\det A \neq 1$, A can be rescaled such that $\det A = 1$.

Remark 48.

- We want an efficient algorithm for the quantum linear system of equations (polynomial in N).
- We implicitly assume that the state $|b\rangle$ can be provided in an efficient way and that A can be implemented on a quantum computer.

Remark 49. Reading coefficients of the state $|x\rangle$ has complexity $\mathcal{O}(N)$. Therefore, using a quantum linear system is only useful if quantum states are required. We will use subroutines for other problems.

4.2. HHL-algorithm (Harrow-Hassidim-Lloyd).

4.2.1. *Idea.* Given a Hermitian matrix $A \in \mathbb{C}^{N \times N}$, we use the spectral decomposition

$$A = \sum_{j=0}^{N-1} \lambda_j |u_j\rangle\langle u_j|$$

with the eigenvalue $\lambda_j \in \mathbb{R}$ and $|u_j\rangle \in \mathbb{R}^N$ being the eigenvectors of A . Now, as

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

it follows from the spectral theorem that

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

and

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

Note that the correct quantum operation is shown later.

4.2.2. *Problem.* The matrices A, A^{-1} are, in general, not unitary and hence $A^{-1}|b\rangle$ is not allowed.

4.2.3. *Solution.* The matrix $U = \exp(iA)$ is unitary and has the same eigenvectors as $A = XDX^{-1}$ since

$$e^{iA} = \sum_{n=0}^{\infty} \frac{(iA)^n}{n!} = \sum_{n=0}^{\infty} \frac{(iXDX^{-1})^n}{n!} = X \sum_{n=0}^{\infty} \frac{(iD)^n}{n!} X^{-1} = X e^{iD} X^{-1} = \sum_j e^{i\lambda_j} |u_j\rangle\langle u_j|.$$

We can therefore

- 1) compute eigenvalues of U with quantum phase estimation, which are the same as the eigenvalues of A .
- 2) Invert the eigenvalues of U with *controlled rotation*, $(\lambda_j \mapsto 1/\lambda_j)$ and
- 3) reverse quantum phase estimation.

4.3. Hamiltonian simulation. Implementing $\exp(iAt) = U$ directly has computational complexity $\mathcal{O}(N^3)$ and is therefore not favorable. How can we implement $\exp(iAt) = U$ and U^{2^k} efficiently?

For A Hamiltonian of U , this is a fundamental problem in quantum computing. Schrödinger's equation

$$i \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle \tag{9}$$

describes every q-system. Furthermore, the solution to Equation (9) is given by

$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle.$$

Thus, for an efficient simulation (*Hamiltonian simulation*), we have to implement e^{-iHt} in an efficient way. This can be done as a circuit up to an error ε .

Definition 50. A Hamiltonian H acting on n qubits can be implemented efficiently if

$$\forall t > 0, \varepsilon > 0 \exists \text{ a quantum circuit } U_H \text{ such that } \|U_H - \exp(iHt)\| \leq \varepsilon$$

where U_H consists of $\text{poly}(n, t, 1/\varepsilon)$ gates.

Remark 51. In general the minimal time required to simulate H at time t is $\mathcal{O}(\|H\|t)$ (no fast forwarding theorem).

For the general approximation problem, it is NP-hard to find a gate decomposition. Hence we will make assumptions to reduce the problems complexity.

4.3.1. *k-local Hamiltonians.* Assume that

$$H = \sum_{j=0}^{N-1} H_j, \quad m \sim \text{poly}(n)$$

and that H_j acts on $k = \mathcal{O}(1)$ qubits.

4.3.2. *Trotter-Suzuki splitting.* The implementation of $\exp(iH_j t)$ is easier for k -local Hamiltonians, as each H_j acts on k qubits. If H is diagonalizable ($H = TDT^H$), we can write $\exp(iHt)$ as

$$\exp(iHt) = T \exp(iDt) T^H.$$

If D_{ll} can be determined efficiently, we can load the entry (i), apply the phase (ii), and unload the entry (iii) to obtain

$$|i0\rangle \xrightarrow{(i)} |iD_{ll}\rangle \xrightarrow{(ii)} e^{iD_{ll}t} |iD_{ll}\rangle \xrightarrow{(iii)} e^{iD_{ll}t} |i0\rangle.$$

We have thus found an efficient diagonalization of H_j for k -local Hamiltonians.

Remark 52. In general

$$e^{iHt} \neq \prod_{j=1}^m e^{iH_j t},$$

as the equality only holds if all H_j commute as

$$e^{A+B} = \sum_{n=0}^{\infty} \frac{(A+B)^n}{n!} = \sum_{n=0}^{\infty} \sum_{k=0}^n \binom{n}{k} \frac{A^k B^{n-k}}{n!} \stackrel{\text{Cauchy}}{=} \sum_{n=0}^{\infty} \frac{A^n}{n!} \sum_{k=0}^n \frac{B^{n-k}}{(n-k)!} = e^A e^B$$

where we have used

$$\binom{n}{k} \frac{1}{n!} = \frac{1}{k!(n-k)!}.$$

Theorem 53. Trotter/Lie-product formula

Let $H = H_1 + H_2$ with H, H_1, H_2 Hermitian. Then

$$\lim_{l \rightarrow \infty} (e^{iH_1 t/l} e^{iH_2 t/l})^l = e^{iHt}$$

and

$$\|e^{iHt} - (e^{iH_1 t/L} e^{iH_2 t/L})^L\| \leq c \frac{t^2}{L}$$

with $c = c(\|H_1\|, \|H_2\|)$.

Proof. We start with a Taylor expansion of $\exp(iH_1 t/L)$:

$$e^{iH_1 t/L} = \text{id} + iH_1 \frac{t}{L} + \underbrace{\mathcal{O}(\|H_1\|^2 \frac{t^2}{L^2})}_{\text{bounded terms in } H_1, t, L}$$

which implies

$$e^{iH_1 t/L} - e^{iH_2 t/L} = \left(\text{id} + iH_1 \frac{t}{L} + \mathcal{O}\left(\|H_1\|^2 \frac{t^2}{L^2}\right) \right)^2 \left(\text{id} + iH_2 \frac{t}{L} + \mathcal{O}\left(\|H_2\|^2 \frac{t^2}{L^2}\right) \right)^2 \quad (10)$$

$$= \left(\underbrace{\text{id} + i(H_1 + H_2) \frac{t}{L}}_{=:A} + \underbrace{\mathcal{O}\left(\max(\|H_1\|, \|H_2\|)^2 \frac{t^2}{L^2}\right)}_{=:B} \right)^2. \quad (11)$$

Now we can use the binomial theorem to obtain

$$(A + B)^2 = A^2 + \sum_{j=0}^{L-1} A^{L-j-1} B A^j + \underbrace{\dots + B^2}_{\mathcal{O}(\|B\|^2)}$$

and hence Equation (11) reads

$$\begin{aligned} & e^{i(H_1+H_2)t} + L \mathcal{O}\left(\max(\|H_1\|, \|H_2\|)^2 \frac{t^2}{L^2}\right) \\ \implies & \|e^{i(H_1+H_2)t} - (e^{iH_1 t/L} e^{iH_2 t/L})^L\| \leq c \frac{t^2}{L}. \end{aligned}$$

Finally, we have

$$\lim_{L \rightarrow \infty} c \frac{t^2}{L} \rightarrow 0.$$

□

Remark 54.

- For efficient simulation $\max(\|H_1\|, \|H_2\|) = \mathcal{O}(\text{poly}(n))$.
- Given a maximal permissible error $\leq \varepsilon$, L is given by $L = \mathcal{O}(c(H_1, H_2)t^2/\varepsilon)$.
- In this case we use first order splitting. However, higher order splitting is also possible, e.g., Strang splitting (second order):

$$\|e^{i(H)t} - (e^{i(H_2)t/(2L)} e^{i(H_1)t/L} e^{i(H_2)t/(2L)})^L\| \leq c \frac{t^3}{L^2}$$

and thus

$$L = \mathcal{O}\left(c(H_1, H_2)t^{\frac{3}{2}}/\sqrt{\varepsilon}\right).$$

Splitting methods of order p are possible with

$$L = \mathcal{O}\left(t^{\frac{p+1}{p}} \varepsilon^{-\frac{1}{p}}\right)$$

which is almost linear in t .

- If k -local Hamiltonians with m -terms are used we obtain

$$\|e^{i(H)t} - (e^{i(H_1)t/L} \dots e^{i(H_m)t/L})^L\| \leq c \frac{m^2 t^2}{L}.$$

- In practice, we tend to over estimate the error. However, the advantages of this method are that it is simple and does not need additional qubits.

4.4. Graph coloring method. An essential question that arises is how can we find the decomposition

$$H = \sum_i H_i.$$

One possible answer is using *graph coloring methods*.

Definition 55. An undirected graph $G(V, E)$ is defined by the set of vertices V and the set of edges $E \subseteq V \times V$ which are unordered pairs of vertices.

Example 56. Let $V = \{1, \dots, 7\}$ and $E = \{(1, 2), (2, 3), (2, 6), (2, 7), (6, 7), (3, 4), (4, 5)\}$. The graph $G(V, E)$ is shown in Figure 15.

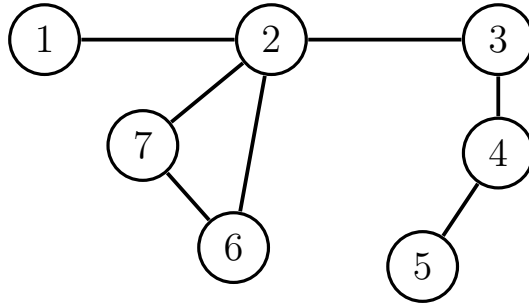


FIGURE 15. Graph corresponding to V and E in Example 56

Definition 57. Two vertices $v_1, v_2 \in V$ are called *adjacent* if $(v_1, v_2) \in E$. The *degree* of a vertex $v \in V$ is the number of adjacent vertices $\#\{v_i : (v, v_i) \in E\}$.

Let $|V| = n$. A graph G can be represented by its $n \times n$ adjacency matrix A given by

$$A_{ij} = \begin{cases} 1 & \text{if } v_i, v_j \in V : (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}.$$

Example 58. The adjacency matrix of Example 56 is given by

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

4.4.1. Graph coloring problem. Is it possible to color the edges of a graph such that no two adjacent edges have the same color, using k different colors?

4.4.2. Solution. The answer to this problem is given by the following theorem.

Theorem 59. Vizing's theorem For any graph $G = (V, E)$ with maximum degree d , the graph can be colored according to Section 4.4.1 with $k \leq d + 1$ colors.

Usage for Hamiltonian simulation:

- 1) Identify H with adjacency matrix A such that

$$A_{ij} = \begin{cases} 1 & \text{if } H_{ij} \neq 0 \\ 0 & \text{if } H_{ij} = 0. \end{cases}$$

We also obtain the associated graph G .

- 2) Find a coloring of G with k colors.
- 3) Split G (A) into k sub-graphs based on color

$$A = \sum_{c=1}^k A_c.$$

For arbitrary matrices A this is impossible to implement efficiently. We therefore require A to be sparse.

4.4.3. *Assumptions.* Let $A \in \mathbb{C}^{2^n \times 2^n}$ be s -sparse and let us have *sparse access* to entries of A , namely

- each row/column of A has at most s non-zero entries.
- We have the *query*

$$O_A : |ij\rangle|0\rangle \mapsto |ij\rangle|A_{ij}\rangle$$

where the back register is large enough to store the entries $A_{ij} \in \mathbb{C}$ exact (or with high enough precision) in binary representation.

- We have another Query

$$O_C : |jl\rangle \mapsto |j\nu(j, l)\rangle$$

where $\nu(j, l) \in \{1, \dots, N-1\}$ is the position of the l -th non-zero entry in the j -th row of A .

- We can execute O_A^{-1} and O_C^{-1} .

Remark 60. *Summary*

- With s -sparsity, the graph G has maximum degree $d \leq s$.
- Vizing's theorem states that G can be colored with at most $s+1$ colors.
- A coloring with s^2 colors can be computed efficiently.
- The A_c matrices are symmetric and 1-sparse. We can therefore efficiently simulate H_C since the matrices are diagonalizable with $\mathcal{O}(1)$ as w.o.l.g.:

$$H = \underbrace{\text{diag}(H)}_{\text{efficient}} + \text{remainder}.$$

- Total computational complexity for a naive implementation is

$$\mathcal{O}(s^2 t^2 \text{poly}(n)/\varepsilon).$$

4.4.4. *Sparse access implementation.* How can we implement *sparse access*?

Example 61. *Circulant matrix* Consider the 3-sparse circulant matrix A given by

$$A := \begin{pmatrix} \alpha & \gamma & 0 & \cdots & 0 & \beta \\ \beta & \alpha & \gamma & 0 & & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & 0 & \beta & \alpha & \gamma \\ \gamma & & 0 & 0 & \beta & \alpha \end{pmatrix}.$$

First, ν is given by $\nu(j, l) = j + l - 1 \bmod N$, for $j = 0, 1, 2$. Note that the matrix indices start with 0. Next, O_C is given by

$$O_C : |j\rangle \mapsto \begin{cases} |\bmod(j-1, N)\rangle & l = 0 \\ |j\rangle & l = 1 \\ |\bmod(j+1, N)\rangle & l = 2 \end{cases}$$

where the $\bmod(j \pm 1, N)$ operation can be implemented using shift permutations. Before we can implement the circuits required to implement O_A and O_C we consider the matrices

$$R = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & & 1 \\ 1 & 0 & \cdots & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 1 & & & 0 \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & 1 & 0 \end{pmatrix}.$$

[L] circuit:

For 3 qubits as in Figure 16 the bitflip X is given by

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

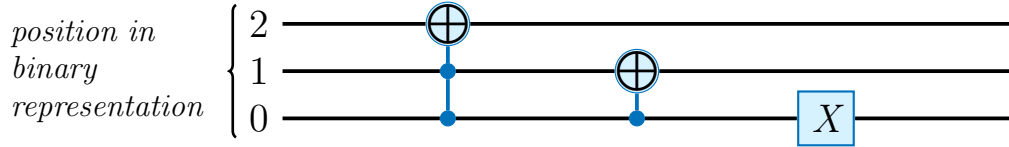


FIGURE 16. Circuit diagram for [L]

Example 62. For $2 \mapsto 3$ we have,

$$|\underbrace{0}_2 \underbrace{1}_1 \underbrace{0}_0\rangle \mapsto |\underbrace{0}_2 \underbrace{1}_1 \underbrace{1}_0\rangle$$

and for $7 \mapsto 0$

$$|111\rangle \mapsto |000\rangle.$$

[R] circuit:

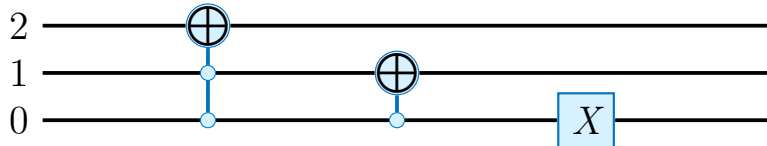


FIGURE 17. Circuit diagram for [R]

where



controlled NOT , active when control is 0 and

analogous CCNOT , active when both controls are 0.

We also need $\boxed{L^2}$ and $\boxed{R^2}$: For L^2 there holds

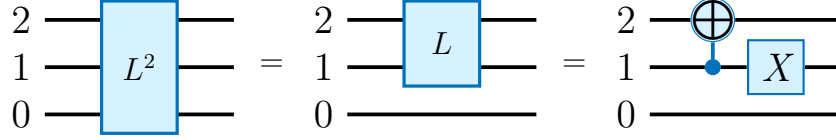


FIGURE 18. Circuit diagram for $\boxed{L^2}$

and the analogue is true for R .

O_C **circuit**:

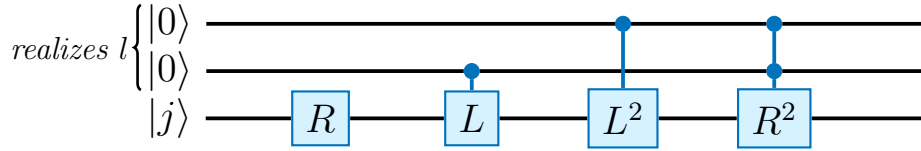


FIGURE 19. Circuit diagram for O_C

Depending on l we have the following cases:

$l = 0$: R -gate for all l , $j \mapsto \text{mod}(j - 1, N)$. This is OK for $l = 0$, hence $|00\rangle$.

$l = 1$: For $l = 1$, $j \mapsto j$ and we obtain $|01\rangle$. Here we can use the L -gate to reverse the R -gate.

$l = 2$: For $l = 2$, $j \mapsto \text{mod}(j + 1, N)$ and we obtain $|10\rangle$. We can use an L^2 -shift since RL^2 results in an L -shift.

$l = 3$: $l = 3 \simeq |11\rangle$ represents the 0 entries in the matrix. We can use the R^2 -shift to reverse the L^2 -shift.

O_A **circuit**:

In general, controlled rotations are unitary operations. Let $|\Theta\rangle$ be the control and $|0\rangle$ be the target. We define

$$U_\Theta : |\Theta\rangle|0\rangle \mapsto |\Theta\rangle(\cos(\pi\Theta)|0\rangle + \sin(\pi\Theta)|1\rangle)$$

where $\Theta \in [0, 1]$. Note that the binary representation of Θ is given by $\Theta = \Theta_0 2^{-1} + \Theta_1 2^{-2} + \dots + \Theta_{d-1} 2^{-d}$.

For $|\Theta\rangle = |0\rangle$ we obtain the output $|0\rangle|0\rangle$. For $|\Theta\rangle = |1\rangle$ we obtain a rotation of $\pi\Theta$ around the y -axis.

In matrix form, we have

$$R(2J) := \begin{pmatrix} \cos(J) & -\sin(J) \\ \sin(J) & \cos(J) \end{pmatrix}$$

which represents a 1-qubit rotation around the y -axis. We will write $\boxed{R(J)}$ for the controlled rotation in circuit diagrams.

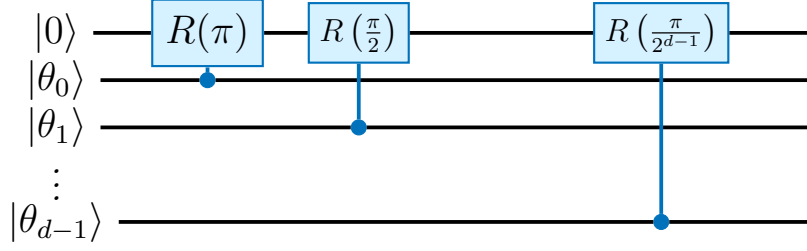


FIGURE 20. Circuit diagram for O_A

Finally, we generate the value $\alpha \in [0, 1]$ by controlled rotation of $\Theta_1 = 1/\pi \arccos \alpha$. Similarly, we generate β and γ as $\Theta_2 = 1/\pi \arccos \beta$ and $\Theta_3 = 1/\pi \arccos \gamma$ respectively.

4.5. Modification for 3-diagonal matrices. We will add two additional rotations $R(\Theta_4)$ and $R(\Theta_5)$ which rotate A_{1n} (A_{n1}) back to 0. We thus need an additional control qubit.

4.6. Loading the right hand side. Let $b \in \mathbb{R}^2$ be arbitrary but fixed. We want to load the state $|b\rangle := \sum_i b_i |i\rangle / \|b\|$ in $\mathcal{O}(\log N)$ (or $|\tilde{b}\rangle \simeq |b\rangle$ up to machine precision). A possible solution is *quantum RAM* (qRAM), e.g., a classical data structure which can be accessed with superpositions on q-states.

Let, without loss of generality, $\|b\| = 1$ which can be loaded in binary representation as it is a fixed scalar.

4.6.1. Idea. The idea we want to use is divide and conquer. We start by saving x in a binary tree B_x with root $\sum_{i=1}^N x_i^2 = 1$, depth $\log N = n$, and leafes $(x_i^2, \text{sgn } x_i)$ as shown in Figure 21.

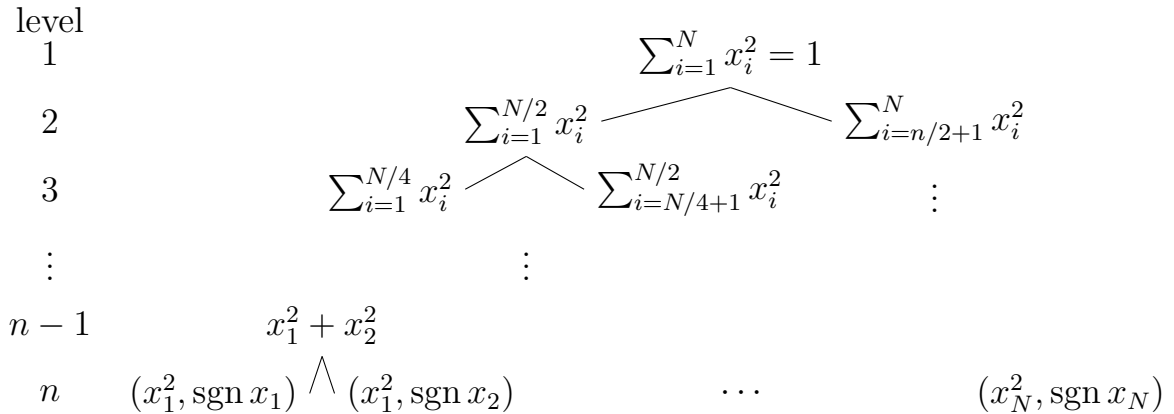


FIGURE 21. Divide and conquer tree for $\sum_{i=1}^N x_i^2 = 1$

Remark 63.

- Each node is the sum of its children. Additionally, each leaf has a sign bit.
- To load a vector, traverse the tree, starting from the root and add new registers if necessary. Controlled rotation is used to realize the value in a node.

- B_x has $\mathcal{O}(N)$ nodes. We assume that the node values are precomputed (classical) and are therefore not included in the complexity.

Algorithm 7. Loading a vector from qRAM

```

1: input:  $x \in \mathbb{R}^N$  represented by its binary tree  $B_x$ 
2: output:  $|x\rangle$ 
3: initialize  $n$ -qubits  $|0 \dots 0\rangle$  with  $|q_1\rangle, \dots, |q_n\rangle$ .
4:  $r = \text{root}(B_x)$ , call  $\text{PROCESSNODE}(r)$ 
5: procedure  $\text{PROCESSNODE}(\text{vertex } u)$ 
6:    $u_l, u_r$  are the children of  $u$ 
7:    $\theta_u = \arccos(\sqrt{u_l/u})$ 
8:   controlled rotation  $|q_k\rangle = \cos \theta |0\rangle + \sin \theta |1\rangle$ , if qubits  $|q_1\rangle, \dots, |q_k\rangle \simeq$  binary representation of  $u$ 
9:   if  $u_l, u_r$  are leafs then
10:      $|q_k\rangle = \text{PROCESSSIGN}(q_k, u_l u_r)$ 
11:     return
12:   else
13:      $\text{PROCESSNODE}(u_l)$ 
14:      $\text{PROCESSNODE}(u_r)$ 
15:   end if
16: end procedure
17: procedure  $\text{PROCESSSIGN}(q_k, u_l, u_r)$ 
18:   if  $\text{sgn}(u_l) = \text{sgn}(u_r) = 1$  then
19:     return  $q_k$ 
20:   else if  $\text{sgn}(u_l) = \text{sgn}(u_r) = -1$  then
21:     return  $-q_k$ 
22:   else if  $\text{sgn}(u_l) = 1, \text{sgn}(u_r) = -1$  then
23:     return  $Z q_k$ 
24:   else if  $\text{sgn}(u_l) = -1, \text{sgn}(u_r) = 1$  then
25:     return  $-Z q_k$ 
26:   end if
27: end procedure

```

Lemma 64. If $x \in \mathbb{R}^N$ is precomputed in B_x , then Algorithm 7 generates the state

$$|x\rangle = \sum_{i=1}^N x_i |i\rangle$$

in $\mathcal{O}(n) = \mathcal{O}(\log N)$.

Proof. First, we show that Algorithm 7 is correct and actually generates the state $|x\rangle$. Go through the tree, multiplying nodes u_k with $\sqrt{u_k/u_{k-1}}$ (for leaf i add an additional $\text{sgn}(x_i)$), where k is the current level. This way, we obtain

$$\prod_{k=1}^n \sqrt{u_k/u_{k-1}} \text{sgn}(x_i) = \underbrace{\sqrt{\frac{u_n}{u_1}}}_{\frac{x_i^2}{1}} \text{sgn}(x_i) = x_i.$$

It remains to show the complexity. There are 2^k rotations (in $\mathcal{O}(1)$) for each level k . This happens in parallel. Thus, a controlled operation on the same qubits. The control checks

the binary register of the parent. The complexity is therefore bounded with

$$\#levels = n = \log N.$$

□

Remark 65. *The assumption that B_x is precomputed is significant and can ruin the speedup. Nonetheless, if B_x is precomputed, Lemma 64 shows that $|b\rangle$ and copies of $|b\rangle$ can be generated in $\mathcal{O}(\log N)$.*

4.7. HHL revisited. Let us now revisit and improve the HHL algorithm. Consider the circuit shown in Figure 22.

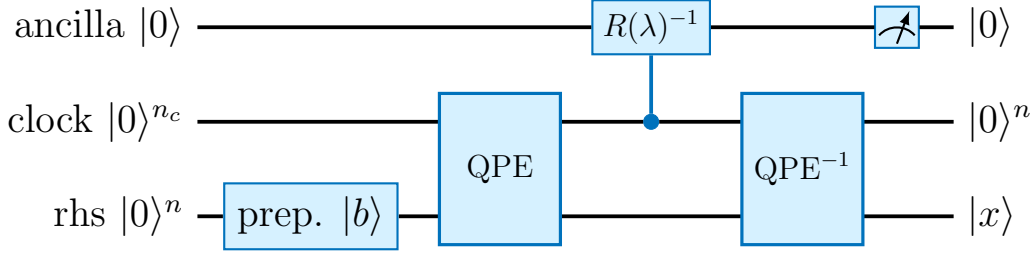


FIGURE 22. Revisited circuit for HHL

- 1) Determine $|b\rangle$ (e.g. qRAM) with $n = \log N$ qubits to represent $|b\rangle$
- 2) Apply quantum phase estimation to

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

where $U = \exp(iA)$ and $|0\rangle = |0^n\rangle^{n_c}$. We therefore need $H^{\otimes n_c}$, U^{2^j} for $j = 0, \dots, n_c - 1$ and QFT with $\mathcal{O}(n^2)$ gates to obtain an approximation for the eigenvalues of A (not U). The result is an approximation as the quantum phase estimation assumes λ_j has an exact binary representation in n_c qubits. The implementation of QPE as circuit is given by Figure 23

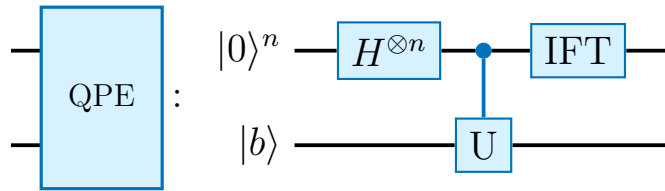



FIGURE 23. QPE circuit diagram

where

 implements $|j\rangle|psi\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{j\lambda_i} |j\rangle|\psi\rangle$ with eigenvalues λ and corresponding eigenvectors ψ .

We apply QPE to $|b\rangle$, i.e.,

$$|0\rangle|b\rangle \mapsto \sum_j \beta_j |\tilde{\lambda}_j\rangle |u_j\rangle$$

where $\tilde{\lambda}_j$ is the binary representation of λ_j .

- 3) Next, we add an ancilla qubit to use controlled rotation ($R_y(2J)$) to rotate $\theta/2 = \arcsin(c/\tilde{\lambda}_j)$ around the y -axis. We obtain

$$\sum_j \beta_j |\tilde{\lambda}_j\rangle |u_j\rangle \left(\sqrt{1 - \frac{c^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{c}{\tilde{\lambda}_j} |1\rangle \right)$$

where c is a constant such that $c \leq \min_j |\lambda_j| = \mathcal{O}(1/k)$.

Note that $\arcsin(\alpha)$ can be realized (approximated) with $\mathcal{O}(\text{poly}(n))$ elementary gates.

- 4) Apply the inverse QPE (*uncomputing*) to obtain

$$\sum_{j=0}^{N-1} \beta_j |0\rangle |u_j\rangle \left(\sqrt{1 - \frac{c^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{c}{\tilde{\lambda}_j} |1\rangle \right).$$

- 5) Measure the last qubit. If the result is $|1\rangle$ the state

$$c \sum_{j=0}^{N-1} \frac{\beta_j}{\tilde{\lambda}_j} |0\rangle |u_j\rangle$$

is proportional to $|\tilde{x}\rangle$.

The probability to measure $|1\rangle$ is

$$\frac{1}{\sqrt{\sum_{j=0}^{N-1} \frac{c^2 |\beta_j|^2}{|\tilde{\lambda}_j|^2}}}.$$

Remark 66. *The normalization factor cancels with c . It is therefore not present in the solution but in the probability of success.*

Example 67. Quantum composer *Let us consider the 1-qubit system:*

$$A = \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

with the solution to $Ax = b$ being

$$x = \frac{1}{8} \begin{pmatrix} 3 \\ 9 \end{pmatrix}.$$

The eigenvalues $\lambda_{1,2}$ and eigenvectors $u_{1,2}$ of A are given by

$$\lambda_0 = \frac{2}{3}, \quad \lambda_1 = \frac{4}{3}, \quad u_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad u_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

We now need $U = \exp(iAt)$ and $U^2 = \exp(i2At)$. Choosing $t = 3\pi/4$ the eigenvalues can be represented with 2 qubits in QPE. For this example, U is given by

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix} \underbrace{\begin{pmatrix} e^{i\lambda_0 t} & 0 \\ 0 & e^{i\lambda_1 t} \end{pmatrix}}_{\begin{pmatrix} i & 0 \\ 0 & -1 \end{pmatrix}} \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -1+i & 1+i \\ 1+i & -1+i \end{pmatrix}$$

and similarly

$$U^2 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

For the implementation with a 4-parameter unitary gate in IBM-Q given by

$$U = \begin{pmatrix} e^{i\gamma} \cos(\theta/2) & -e^{i(\gamma+\lambda)} \sin(\theta/2) \\ e^{i(\gamma+\phi)} \sin(\theta/2) & e^{i\gamma+\phi+\lambda} \cos(\theta/2) \end{pmatrix}$$

we obtain

$$\theta = \frac{\pi}{2}, \quad \phi = -\frac{\pi}{2}, \quad \lambda = \frac{\pi}{2}, \quad \gamma = \frac{3\pi}{4}$$

for U and

$$\theta = \pi, \quad \phi = \pi, \quad \lambda = 0, \quad \gamma = 0$$

for U^2 .

controlled rotation:

The eigenvalues from QPE (2-qubits) are $\tilde{\lambda}_j = N\lambda_j t/2\pi$ and hence

$$\tilde{\lambda}_0 = 1, \quad \tilde{\lambda}_1 = 2.$$

Choosing $c = 1$ we obtain

$$\begin{aligned} \theta_1 &= 2 \arcsin \left(\frac{1}{\tilde{\lambda}_1} \right) = \pi, \\ \theta_1 &= 2 \arcsin \left(\frac{1}{\tilde{\lambda}_2} \right) = \frac{\pi}{3}. \end{aligned}$$

results:

The output are normalized probability values

<i>state</i>	<i>P</i>
$ 00\rangle$	0.1875
$ 01\rangle$	0.0625
$ 10\rangle$	0.1875
$ 11\rangle$	0.5625

(12)

where the first qubit is $|b\rangle$ and the second qubit is the ancilla qubit. In this case, only the states where the ancilla qubit is $|1\rangle$ are relevant, i.e., $|01\rangle$ and $|11\rangle$. For this conditional measurement of the ancilla qubit we obtain $|0\rangle$ with $p = 1/10$ and $|1\rangle$ with $p = 9/10$. The normalized solution is thus given by

$$|x\rangle = \frac{3}{\sqrt{90}}|0\rangle + \frac{9}{\sqrt{90}}|1\rangle.$$

4.8. Error and complexity analysis. Until now we assumed that all scalars have an exact binary representation. However, in general this is not feasible for all eigenvalues of A . Quantum phase estimation outputs eigenvalues $\tilde{\lambda}_j$ which are approximations of λ_j with error

$$|\tilde{\lambda}_j \lambda_j| \leq \delta$$

with probability $1 - 1/\text{poly}(n)$. The run time of QPE is $\mathcal{O}(T_u \text{poly}(n)/\delta)$ where T_u is the time required to implement $U = \exp(iA)$.

4.8.1. *problem.* The next problem is the numerical stability of

$$\lambda_j \mapsto \frac{1}{\lambda_j}$$

with controlled rotation. If $\lambda_j \approx 0$ a small error $\tilde{\lambda}_j + \varepsilon$ has a large impact on the result. The effect of the small perturbation ε is significantly larger than ε itself as

$$\left| \frac{1}{\lambda_j} - \frac{1}{\tilde{\lambda}_j} \right| = \left| \frac{\varepsilon}{\lambda_j(\lambda_j + \varepsilon)} \right| \approx \frac{\varepsilon}{\lambda_j^2}.$$

The problem has a bad condition number $\kappa(A)$.

4.8.2. *solution.* The solution are filter functions. We only invert eigenvalues λ if $\lambda \geq 1/\kappa$. Accordingly, we define f as

$$f(\lambda) := \begin{cases} 0 & \text{for } \lambda < \frac{1}{2\kappa} \\ \frac{1}{2\kappa\lambda} & \text{for } \lambda \geq \frac{1}{\kappa} \\ \frac{1}{2} \sin\left(\frac{\pi}{2}(2\kappa\lambda - 1)\right) & \text{otherwise.} \end{cases}$$

Note that for $1/(2\kappa) \leq \lambda < 1/\kappa$, f is interpolating between 0 and $1/(2\kappa\lambda)$ and is thus continuous.

Similarly, we define the reverse filter g as

$$g(\lambda) := \begin{cases} 0 & \text{for } \lambda \geq \frac{1}{\kappa} \\ \frac{1}{2\kappa\lambda} & \text{for } \lambda < \frac{1}{2\kappa} \\ \frac{1}{2} \cos\left(\frac{\pi}{2}(2\kappa\lambda - 1)\right) & \text{otherwise.} \end{cases}$$

Note that g is also continuous and $f^2(\lambda) + g^2(\lambda) \leq 1$ for all λ .

Remark 68. *The choice of f and g is not unique, other options are possible. The cut-off $1/(2\kappa)$ can also be chosen differently.*

Instead of controlled rotation, add a 3-qubit register:

$$|h(\tilde{\lambda}_j)\rangle := \sqrt{1 - f(\tilde{\lambda}_j)^2 - g(\tilde{\lambda}_j)^2} |\text{nothing}\rangle + f(\tilde{\lambda}_j) |\text{well}\rangle + g(\lambda_j) |\text{ill}\rangle$$

where $|\text{nothing}\rangle$ is the part where no inversion was performed, $|\text{well}\rangle$ is the part where the eigenvalues have been inverted and $|\text{ill}\rangle$ is the part of $|b\rangle$ inside the ill-conditioned subspace of A .

Finally, we apply $(QPE)^{-1}$ and measure $|\text{well}\rangle$.

The result is roughly

$$\sum_{j, \lambda_j \geq 1/\kappa} \lambda_j^{-1} \beta_j |u_j\rangle |\text{well}\rangle + \sum_{j, \lambda_j < 1/\kappa} \beta_j |u_j\rangle |\text{ill}\rangle.$$

Lemma 69. *The map $\lambda \mapsto |h(\lambda)\rangle$ is Lipschitz continuous with Lipschitz constant $L = \mathcal{O}(\kappa)$, namely*

$$\| |h(\lambda_i)\rangle - |h(\lambda_j)\rangle \|_2 \leq c\kappa |\lambda_i - \lambda_j|. \quad (13)$$

Proof. After estimating the derivatives of f and g , Equation (13) can be verified explicitly. \square

4.8.3. *Goal.* Our next Goal is an error estimation for the inexact QPE with filtered inversion of the eigenvalues. The exact state and the approximated state are given by

$$\begin{aligned} \text{exact:} \quad & |\Psi\rangle := \sum_i \beta_i |u_i\rangle |h(\lambda_i)\rangle \\ \text{approximation:} \quad & |\tilde{\Psi}\rangle := \sum_i \beta_i |u_i\rangle |h(\tilde{\lambda}_i)\rangle. \end{aligned}$$

Hence the error in the 2-norm is

$$\| |\Psi\rangle - |\tilde{\Psi}\rangle \|_2^2 = \| |\Psi\rangle \|_2^2 + \| |\tilde{\Psi}\rangle \|_2^2 - 2 \left(1 - \underbrace{\Re\langle\Psi|\tilde{\Psi}\rangle}_{\in[0,1], \text{ (C.S.)}} \right). \quad (14)$$

For u_i ONB there holds

$$\Re\langle\Psi|\tilde{\Psi}\rangle = \sum_{i=1}^N |\beta_i|^2 \langle h(\lambda_i) | h(\tilde{\lambda}_i) \rangle$$

and using ??

$$\Re\langle\Psi|\tilde{\Psi}\rangle \geq 1 - \frac{c^2 \kappa^2}{2} |\lambda_i - \tilde{\lambda}_i|^2 \geq 1 - \frac{c^2 \kappa^2 \delta^2}{2}. \quad (15)$$

Thus the error per eigenvalue is bound by δ from above. Plugging into Equation (14) we obtain

$$\| |\Psi\rangle - |\tilde{\Psi}\rangle \|_2^2 \leq \underbrace{\sum_{i=1}^N |\beta_i|^2}_{=1} c^2 \kappa^2 \delta^2 = \delta^2 c^2 \kappa^2$$

and therefore

$$\| |\Psi\rangle - |\tilde{\Psi}\rangle \|_2 \leq c \kappa \delta.$$

An error of $\mathcal{O}(\varepsilon)$ is achieved with an QPE error $\mathcal{O}(\varepsilon/\kappa)$ and run time $\mathcal{O}(\kappa \text{poly}(n)/\varepsilon)$.

For the measurement we want the ancilla qubit to be $|1\rangle$ for well conditioned A and $|well\rangle$ for arbitrary A . The probability of success is (using $\sum |\beta_i|^2 / |\lambda_i|^2 \simeq |A^{-1}b|$)

$$p \geq \sum_{i: \lambda_i \geq \frac{1}{\kappa}} |\beta_i|^2 \left| \frac{1}{\tilde{\lambda}_i \kappa} \right|^2 = \mathcal{O} \left(\frac{1}{\kappa^2} \right).$$

Using *amplitude amplification* (similar to Algorithm 5) we can increase the probability of success to $\mathcal{O}(1/\kappa)$. Hence, $\mathcal{O}(\kappa)$ is required for the procedure to get $|well\rangle$ with arbitrarily high probability.

4.8.4. Total computational complexity.

- state preparation: $\mathcal{O}(n)$, if QRAM is precomputed
- Hamiltonian simulation: If A is s -sparse, $\exp(iAt)$ can be implemented (up to an error ε in $\mathcal{O}(nst \text{poly}(\log(st/\varepsilon)))$.
Note: this is the best result shown in literature. The simple method shown in the lecture has a complexity of $\mathcal{O}(ns^2 t^2 / \varepsilon)$.
- QPE: $\mathcal{O}(\text{poly}(n) \kappa / \varepsilon \cdot T_u)$
- Amplitude amplification: $\mathcal{O}(\kappa)$

In total, the complexity is $\mathcal{O}(\text{poly}(n)\kappa^2 s/\varepsilon \cdot \text{poly}(\log(s/\varepsilon)))$. Compared to the CG method's $\mathcal{O}(\exp(n)s\sqrt{\kappa} \ln(2/\varepsilon))$, we see an exponential speed-up in n but a slowdown in κ, ε . Can we improve the complexity in κ, ε ?

4.9. Improvements on the HHL-algorithm. Comparing the CG method and the HHL algorithm we see that

CG: complexity vs. exactness $\mathcal{O}(\log(1/\varepsilon))$

HHL: Due to QPE only $\mathcal{O}(1/\varepsilon)$

4.9.1. *Goal.* Our goal is therefore to derive q-linear system algorithms which also have a logarithmic dependence on ε^{-1} but retain the exponential speedup in n .

4.9.2. *Idea.* Use a direct approximation of A^{-1} similar to Caley-Hamilton theorem. Let p be the characteristic polynomial of A with $p(A) = 0$

$$\begin{aligned} {}^G\mathcal{P}_N &\implies A^N + \alpha_{N-1}A^{N-1} + \dots + \alpha_1A + \alpha_0\text{id} = 0, \quad \alpha_i \in \mathbb{C} \\ \iff A^{-1} &= -\frac{1}{\alpha_0} (A^{N-1} + \alpha_{N-1}A^{N-2} + \dots + \alpha_1\text{id}) = p_{N-1}(A). \end{aligned}$$

Determining α_i is too expensive but there might exist a polynomial $q_m \in \mathcal{P}_m$ with $m \ll N$ such that

$$A^{-1} \approx q_m(A)$$

and q_m can be implemented efficiently.

There are two possibilities to find q_m ,

- 1) using trigonometric polynomials with a Fourier approximation which is unitary and
- 2) using Chebyshev polynomials and minimizing $\|q_m(A)\|_2$.

We will now examine how the approximation of A^{-1} effects the final state.

Lemma 70. *Let B be a Hermitian matrix with $\|B^{-1}\| \leq 1$ and D such that $\|B - D\| \leq \varepsilon < 1/2$. Then the states*

$$|x\rangle := \frac{B|\psi\rangle}{\|B|\psi\rangle\|}, \quad |\tilde{x}\rangle := \frac{D|\psi\rangle}{\|D|\psi\rangle\|}$$

satisfy

$$\| |x\rangle - |\tilde{x}\rangle \| \leq 4\varepsilon.$$

Proof. Using the triangle inequality we obtain

$$\begin{aligned} \| |x\rangle - |\tilde{x}\rangle \| &= \left\| \frac{B|\psi\rangle}{\|B|\psi\rangle\|} - \frac{D|\psi\rangle}{\|D|\psi\rangle\|} \right\| \\ &\leq \left\| \frac{B|\psi\rangle}{\|B|\psi\rangle\|} - \frac{B|\psi\rangle}{\|D|\psi\rangle\|} \right\| + \frac{1}{\|D|\psi\rangle\|} \| \|B|\psi\rangle\| - \|D|\psi\rangle\| \| \\ &\leq \frac{\|D|\psi\rangle\| - \|B|\psi\rangle\|}{\|D|\psi\rangle\|} + \frac{1}{\|D|\psi\rangle\|} \| \|B|\psi\rangle\| - \|D|\psi\rangle\| \|. \end{aligned}$$

By assumption $1 \leq \|B|\psi\rangle\|$, hence using the triangle inequality once more we end up with

$$1 \leq \|B|\psi\rangle\| \leq \|D|\psi\rangle\| + \|(B - D)|\psi\rangle\| \leq \|D|\psi\rangle\| + \varepsilon$$

and therefore

$$\frac{\|D|\psi\rangle\| - \|B|\psi\rangle\|}{\|D|\psi\rangle\|} \leq \frac{\varepsilon}{\|D|\psi\rangle\|} + \frac{\varepsilon}{\|D|\psi\rangle\|} \leq 2\frac{\varepsilon}{1-\varepsilon} \leq 4\varepsilon.$$

□

We can apply Lemma 70 to $B = A^{-1}$ and the approximation $D \approx A^{-1}$.

4.9.3. *Fourier approach.* We will now approximate A^{-1} as a linear combination of unitary operators, namely

$$A^{-1} \approx \sum_j \alpha_j e^{-iAt_j}.$$

We are therefore interested in

- how well can A^{-1} be approximated by a Fourier series and
- can it be implemented efficiently.

Without loss of generality, let $\alpha_j > 0$ for all j as the phase cannot be measured.

4.9.4. *Goal.* Our goal is the implementation of

$$M = \sum_j \alpha_j U_j$$

where U_j are unitary operators. Note that M itself does not have to be unitary.

Lemma 71. *Let $M = \sum_j \alpha_j U_j$ with $\alpha_j > 0$ and U_j unitary. Furthermore, let V be the map defined by*

$$V|0^m\rangle := \frac{1}{\sqrt{\alpha}} \sum_j \sqrt{\alpha_j} |j\rangle$$

with $\alpha = \sum_j \alpha_j$ and

$$U := \sum_j |j\rangle\langle j| \otimes U_j.$$

Then $W := V^H U V$ satisfies

$$W|0^m\rangle|\psi\rangle = \frac{1}{\alpha} M|\psi\rangle + |\Phi^\perp\rangle$$

for all states $|\psi\rangle$ with $\pi|\Phi^\perp\rangle := (|0^m\rangle\langle 0^m| \otimes \text{id})|\Phi^\perp\rangle = 0$.

Here the first term realizes M (later A^{-1}) and the second term is orthogonal to $|0^m\rangle$ inside the first register. Thus we can implement M by measuring $|0^m\rangle$ in the first register.

Proof.

$$\begin{aligned} W|0^m\rangle|\psi\rangle &= V^H U \left(\frac{1}{\sqrt{\alpha}} \sum_j \sqrt{\alpha_j} |j\rangle |\psi\rangle \right) \stackrel{\text{select } U}{=} V^H \left(\frac{1}{\sqrt{\alpha}} \sum_j \sqrt{\alpha_j} |j\rangle U_j |\psi\rangle \right) \\ &= \Pi V^H \left(\frac{1}{\sqrt{\alpha}} \sum_j \sqrt{\alpha_j} |j\rangle U_j |\psi\rangle \right) + (\text{id} - \Pi) V^H \left(\frac{1}{\sqrt{\alpha}} \sum_j \sqrt{\alpha_j} |j\rangle U_j |\psi\rangle \right) \\ &= (|0^m\rangle \otimes \text{id}) \left(\frac{1}{\sqrt{\alpha}} \sum_j \sqrt{\alpha_j} |j\rangle U_j |\psi\rangle \right) + |\Phi^\perp\rangle \end{aligned}$$

where we used the fact that $\Pi(\text{id} - \Pi) = 0$ for the last equality. Since Π projects the first register onto $|0^m\rangle$, and with the definition of V^H we get

$$W|0^m\rangle|\psi\rangle = \frac{1}{\alpha}|0^m\rangle \sum_j \alpha_j U_j |\psi\rangle + |\Phi^\perp\rangle.$$

□

Remark 72.

- *Select U chooses U_i using a control register.*
- *The probability of success is $\|M|\psi\rangle\|^2/\alpha^2$. In our use case, $M = A^{-1}$, $|\psi\rangle = |b\rangle$ with QRAM, multiple $|b\rangle$ preparation is possible (rotation of $|b\rangle$), facilitating a quadratic speedup with high probability of $\mathcal{O}(\alpha/\|M|\psi\rangle\|)$ iterations.*
- *V is unitary and hence easy to implement.*
- *If the unitary operators U_i are easy to implement and the decomposition has $\mathcal{O}(\log N)$ terms, M can be implemented efficiently.*
- *The query-complexity of U is approximately the query complexity of the most expensive U_i . This is, in general, not desirable for the gate-complexity. In this special case however it is as all $U_j = \exp(-iA)^{t_j}$ have the same complexity.*
- *Using a diagonalization of $A = T^H D T$ we obtain*

$$T^H D^{-1} T = A^{-1} \stackrel{!}{\approx} \sum_j \alpha_j e^{-iA t_j} = T^H \sum_j \alpha_j e^{-iD t_j} T$$

and since $D = \text{diag}(\lambda_j)$ we only need an expansion of

$$f(x) = \frac{1}{x} \stackrel{!}{\approx} \sum_j \alpha_j e^{-i x t_j}$$

for $x \in \sigma(A)$ or $x \in \Omega \supseteq \sigma(A)$.

Similar to HHL, we only consider eigenvalues with good condition number ($\lambda_j \geq 1/\kappa$) and a scaling such that $\max_j \lambda_j = 1$. We look for an approximation of f on $D_k := [-1, 1] \setminus [-1/\kappa, 1/\kappa]$.

4.9.5. *Idea.* To approximate f we will

- 1) smooth f in a neighborhood of 0,
- 2) apply a Fourier expansion (integral),
- 3) and truncate the expansion by replacing the integral with a finite sum.

We end up with a function h of the form $\sum_j \alpha_j e^{-i x t_j}$ with

$$\sup_{x \in D_k} |f(x) - h(x)| \leq C\varepsilon.$$

?? implies that $h(A)|b\rangle$ (with normalization) is an approximation of $|x\rangle$.

Lemma 73. *The function*

$$h(x) := \frac{i}{\sqrt{2\pi}} \sum_{j=0}^{J-1} \sum_{k=-K}^K \Delta_y \Delta_z z_k e^{-\frac{z_k^2}{2}} e^{-i x y_j z_k}$$

where

$$y_j := j \Delta_y, \quad z_k := k \Delta_z, \quad J = \mathcal{O}\left(\frac{\kappa}{\varepsilon} \log \frac{\kappa}{\varepsilon}\right), \quad K = \mathcal{O}\left(\kappa \log \frac{\kappa}{\varepsilon}\right),$$

$$\Delta_y = \mathcal{O}\left(\frac{\varepsilon}{\sqrt{\log(\kappa/\varepsilon)}}\right), \text{ and } \Delta_z = \mathcal{O}\left(\kappa \frac{1}{\sqrt{\log(\kappa/\varepsilon)}}\right)$$

satisfies

$$\sup_{x \in D_k} |h(x) - \frac{1}{x}| \leq C\varepsilon.$$

Proof. Defining $g(y) := y \exp(-y^2/2)$, and using substitution we find

$$\frac{1}{x} = \int_0^\infty g(xy) dy$$

since

$$\int_0^\infty g(x) dy = 1.$$

We choose g in this way as it decays fast, is smooth, and because

$$\mathcal{F}(g) = -ig$$

g is an eigenfunction of the Fourier expansion with eigenvalue $-i$. This implies

$$g = -i\mathcal{F}(g) = \frac{i}{\sqrt{2\pi}} \int_{\mathbb{R}} z e^{-\frac{z^2}{2}} e^{-iyz} dz \quad (16)$$

$$\implies \frac{1}{x} = \int_0^\infty \int_{\mathbb{R}} z e^{-\frac{z^2}{2}} e^{-ixyz} dz dy. \quad (17)$$

Note that $h(x)$ is a Riemann sum for the truncated integral in Equation (16).

1. Step Summation/integration in y : Using the geometric series it follows that

$$h(x) = \frac{i\Delta_y}{\sqrt{2\pi}} \sum_{k=-K}^K \Delta_z z_k e^{-\frac{z_k^2}{2}} \frac{1 - e^{-ixy_j z_k}}{1 - e^{-ix\Delta_y z_k}} \quad (18)$$

and

$$\frac{1}{x} = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{x} \int_{\mathbb{R}} e^{-\frac{z^2}{2}} dz.$$

2. Step Approximation of $1/x \int_{\mathbb{R}} e^{-\frac{z^2}{2}} dz$ with an infinite Riemann-sum: Using Poisson's summation formula

$$\sum_{k=-\infty}^{\infty} f(k) = \sqrt{2\pi} \sum_{k=-\infty}^{\infty} \hat{f}(2\pi k)$$

we obtain

$$\frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \Delta_z e^{-\frac{z_k^2}{2}} = \sum_{k=-\infty}^{\infty} e^{-\left(\frac{2\pi k}{\Delta_z}\right)^2/2} = 1 + \sum_{|k| \geq 1} e^{-\left(\frac{2\pi k}{\Delta_z}\right)^2/2} \quad (19)$$

with mesh size Δ_z and therefore

$$\left| \frac{1}{x} \left(1 - \frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \Delta_z e^{-\frac{z_k^2}{2}} \right) \right| \stackrel{(19)}{\leq} \frac{1}{|x|} \cdot 2 \sum_{k=1}^{\infty} e^{-\frac{2\pi^2 k^2}{\Delta_z^2}} = \frac{1}{|x|} \frac{2}{e^{2\pi^2 \Delta_k^2} - 1} \leq CK \cdot \varepsilon e^{-\kappa^2 \log \kappa}.$$

After choosing Δ_z and with $1/|x| \leq \kappa$ we have

$$CK \cdot \varepsilon e^{-\kappa^2 \log \kappa} \leq \tilde{C}\varepsilon.$$

3. Step Truncating the series expansion: By adding $(1 - \exp(-ixy_j z_k))$, using the triangle inequality, Poisson summation formula, and estimating the rest of the series by choosing K we obtain

$$\left| \frac{1}{\sqrt{2\pi}x} \left(\sum_{k=-\infty}^{\infty} \Delta_z e^{-\frac{z_k^2}{2}} - \sum_{k=-K}^K \Delta_z e^{-\frac{z_k^2}{2}} (1 - e^{-ixy_j z_k}) \right) \right| \leq C\varepsilon. \quad (20)$$

Deriving Equation (20) is a lengthy calculation and is therefore omitted at this point. It can, however, be found in the literature.

4. Step Since

$$\left| \frac{1}{1 - e^{-ix}} - \frac{1}{x} \right| < 1 \quad \forall x \in [-1, 1]$$

we have

$$\frac{i\Delta_y z_k}{1 - e^{-ix\Delta_y z_k}} - \frac{1}{x} \leq \Delta_y |z_k|$$

and hence

$$\begin{aligned} |h(x) - \frac{1}{\sqrt{2\pi}x} \sum_{k=-K}^K \Delta_z e^{-\frac{z_k^2}{2}} (1 - e^{-ixy_j z_k})| &\stackrel{(18)}{\leq} \frac{2}{\sqrt{2\pi}} \left| \sum_{k=-K}^K \left(\frac{i\Delta_y z_k}{1 - e^{-ix\Delta_y z_k}} - \frac{1}{x} \right) \cdot \Delta_z e^{-\frac{z_k^2}{2}} \right| \\ &\leq \sqrt{\frac{2}{\pi}} \Delta_y \underbrace{\sum_{k=-K}^K \Delta_z e^{-\frac{z_k^2}{2}}}_{\leq \int_0^\infty \exp(-z^2/4) dz} = C\Delta_y \Delta_z \leq C\varepsilon \end{aligned}$$

up to log-terms.

5. Step By combining the steps 2-4 with the triangle inequality leads to

$$|h(x) - \frac{1}{x}| \leq C\varepsilon.$$

□

4.9.6. Complexity.

Theorem 74. *The quantum linear system of equations can be solved for $\exp(-iAt)$ with $t = \mathcal{O}(\kappa \log(\kappa/\varepsilon))$ and accuracy $\mathcal{O}(\varepsilon/(\kappa \sqrt{\log(\kappa/\varepsilon)}))$, by applying Hamiltonian simulation $\mathcal{O}(\kappa \sqrt{\log(\kappa/\varepsilon)})$ times. The gate-complexity is given by*

$$\mathcal{O} \left(s\kappa^2 \log^{2.5} \left(\frac{\kappa}{\varepsilon} \left(\log N + \log^{2.5} \left(\frac{\kappa}{\varepsilon} \right) \right) \right) \right).$$

Proof. At this point, we will only give a sketch of the proof. The full proof can be found in the literature.

We have to implement U and V . For sparse A we can use Hamiltonian simulation with

$$\mathcal{O} \left(st \left(\log N + \log^{2.5} \left(\frac{t}{\varepsilon} \right) \right) \log \frac{t}{\varepsilon} \right)$$

for U and select U_i . The choice of t implies a certain accuracy. Lastly, Hamiltonian simulation has to be applied $\mathcal{O}(\kappa \sqrt{\log(\kappa/\varepsilon)})$ times. □

4.9.7. Chebychev approach.

4.9.8. *Idea.* In this section we will replace the Fourier expansion by Chebychev polynomials.

4.9.9. *Problem.* The chebychev polynomials are not unitary.

4.9.10. *Solution.* The solution is *block-encoding* ($n + 1$ -qubit block encoding). Let $A \in \mathbb{C}^{2^n \times 2^n}$ with $\|A\| \leq 1$ and assume there exists a unitary matrix $U \in \mathbb{C}^{2^{n+1} \times 2^{n+1}}$ such that

$$U = \begin{pmatrix} A & * \\ * & * \end{pmatrix}$$

where $* \in \mathbb{C}^{2^n \times 2^n}$ are arbitrary but unitary. Now, for all states $|\psi\rangle$, consider $|0\rangle|\psi\rangle \simeq (\psi \ 0)^T$. Then

$$U(|0\rangle|\psi\rangle) = |0\rangle A|\psi\rangle + \underbrace{|1\rangle|\Phi^\perp\rangle}_{\text{unimportant}}$$

and thus, $A|\psi\rangle$ can be obtained by measuring $|0\rangle$ in the first qubit as depicted in Figure 24.

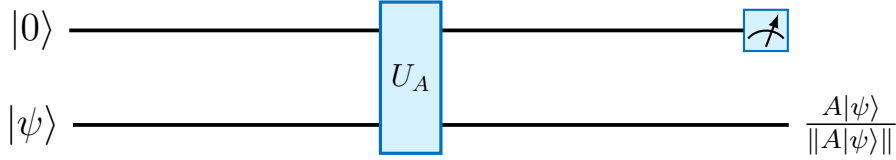


FIGURE 24. Circuit to obtain $A|\psi\rangle$ for Chebychev approach

Remark 75.

- *LCU (without amplitude amplification) is a special case of block-encoding.*
- *Block encoding for an arbitrary A is too expensive. However, for a sparse A it is efficiently feasible.*

4.9.11. *Goal.* Our goal is to implement block-encoding for A^{-1} .

Theorem 76. Let $p \in \mathcal{P}_d$ with $\|p\|_{\infty, [-1, 1]} \leq 1/4$ be a polynomial of degree $\leq d$ and U block-encoding of A with $n + a$ -qubits.

A block-encoding of $P(A)$ with $n + \mathcal{O}(a)$ qubits can be achieved with d applications of U, U^{-1} , one application of controlled- U and $\mathcal{O}(ad)$ elementary gates (2-qubits).

Example 77. Application of Theorem 76 To approximate $f(x) = 1/x$ by $p(x)$ we use the fact that for $\tilde{d} = \mathcal{O}(\kappa^2 \log(\kappa/\varepsilon))$ we have

$$\sup_{x \in D_k} \left| \underbrace{\frac{1 - (1 - x^2)^{\tilde{d}}}{x}}_{\in \mathcal{P}_{2\tilde{d}-1}} - \frac{1}{x} \right| \leq \frac{\varepsilon}{2}.$$

Now using the Chebychev polynomials T_j we get

$$\frac{1 - (1 - x^2)^{\tilde{d}}}{x} = \sum_{j=0}^{2\tilde{d}-1} \alpha_j T_j.$$

Furthermore, truncating the expansion at $d = \mathcal{O}(\kappa \log(\kappa/\varepsilon))$ results in an error $\leq \varepsilon/2$. Theorem 76 provides an efficient implementation of $P_d(A)$. Through the use of block-encoding we obtain the state $|\tilde{x}\rangle$ with $||\tilde{x}\rangle - |x\rangle| \leq C\varepsilon$.

4.9.12. *Complexity.* In total, the gate-complexity is given by

$$\mathcal{O}\left(s\kappa^2 \log^2\left(\frac{s\kappa}{\varepsilon}\right) \left(\log N + \log^{2.5}\left(\frac{s\kappa}{\varepsilon}\right)\right)\right),$$

which is better than the Fourier approach. However, the Chebychev approach needs direct sparse access which is not possible for arbitrary matrices. Using Hamiltonian simulation instead of the Fourier approach is therefore more general.

5. SOLVING LINEAR DIFFERENTIAL EQUATIONS

Every linear differential equation can be reduced to a first order system

$$x'(t) = Ax(t) + b(t) \quad \in \mathbb{R}^{N_x} x(0) = x_0$$

where $A \in \mathbb{R}^{N_x \times N_x}$ is sparse, $b \in \mathbb{R}^{N_x}$, $x_0 \in \mathbb{R}^{N_x}$ and $s \in \mathbb{N}$.

Idea 1: Lie-Trotter approach:

- exponential cost in t
- no inhomogeneous equations

Idea 2: Space-time approach [1] Feynman's clock: Encode time in basis states $|j\rangle$ and produce output state

$$|\psi(t)\rangle = \sum_{j=0}^{N_t} |j\rangle |x_j\rangle$$

where $x_j \approx x(t_j)$ and

$$\begin{array}{ll} N_t = \frac{T}{\tau} \dots & \text{number of time steps} \\ T \dots & \text{final time} \\ \tau \dots & \text{time step size.} \end{array}$$

Since the probability of measuring $x(T) \approx x_{N_t}$ is small, we can extend the ODE beyond T with

$$A(t) := \text{id}, \quad b(t) := 0 \quad \forall (T, 2T].$$

Thus, increasing the probability as $x(t) = x(T)$ for $t \in [T, 2T]$.

Example 78. *Forward Euler A simple example is the forward Euler method given by*

$$\frac{x_{j+1} - x_j}{\tau} = A(t_j)x_j + b(t_j) \quad \forall t \in [0, 2T].$$

We define the vectors \bar{x} , \bar{b} and the matrix $\underline{A} \in \mathbb{R}^{2N_t N_x \times 2N_t N_x}$ as

$$\bar{x} := \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{2N_t} \end{pmatrix}, \quad \bar{b} := \begin{pmatrix} x_0 \\ b_0\tau \\ b_1\tau \\ \vdots \\ b_{N_t}\tau \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \underline{A} := \begin{pmatrix} \text{id} & 0 & \dots & 0 \\ -(\text{id} + A\tau) & \text{id} & & \\ & \ddots & & \vdots \\ & -(\text{id} + A\tau) & \text{id} & \\ & & -\text{id} & \text{id} \\ & & & \ddots & 0 \\ & & & & -\text{id} & \text{id} \end{pmatrix}$$

leads to the formulation $\underline{A}\bar{x} = \bar{b}$.

The local Euler error $\approx \tau^2$ and hence the error at end-time is $2N_t\tau^2 \approx T^2/N_t$. To achieve

error ε , we need $N_t \approx T^2/\varepsilon$. Using the HHL-algorithm requires a bounded condition number. Consider

$$\begin{pmatrix} 1 & & & 0 \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{pmatrix} \underbrace{\begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix}}_x = \underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_b$$

with

$$\|\underline{A}^{-1}\| \geq \frac{\|x\|}{\|b\|} = \frac{\sqrt{\sum_{i=1}^n i^2}}{\sqrt{\sum_{i=1}^n 1}} \approx \frac{n^{\frac{3}{2}}}{n^{\frac{1}{2}}} = n.$$

Hence the condition number κ of \underline{A} is $\kappa \approx 2N_t N_x \approx T^2$ and thus the HHL algorithm requires at least $\kappa^2 \approx T^4$ operations.

5.1. Multi-step methods. The idea is to use the information of multiple time steps to increase the accuracy. The multistep method is given by

$$\sum_{l=0}^k d_l x_{j+l} = \tau \sum_{l=0}^k \beta_l (A(t_{j+l})x_{j+l} + b(t_{j+l}))$$

and its stability by

$$\rho(\xi) = \sum_{j=0}^k \alpha_j \xi^j, \quad \sigma(\xi) = \sum_{j=0}^k \beta_j \xi^j.$$

Let $R_j(\mu)$ denote the roots of $\rho(\xi) - \mu\sigma(\xi) = 0$ and define

$$S := \left\{ \mu \in \mathbb{C} : \begin{array}{l} \text{all roots of } R_j(\mu) \text{ satisfy } |R_j(\mu)| \leq 1 \\ \text{multiple roots } R_j(\mu) \text{ satisfy } |R_j(\mu)| < 1 \end{array} \right\}.$$

If all roots of σ are elements of S , the method is stable at infinity. In matrix form the method reads

$$\underline{A} = \begin{cases} \underline{A}_{j,j} = & 0 \leq j \leq k, \quad N_t < j \leq 2N_t \\ \underline{A}_{j,j-1} = -(\text{id} + A\tau) & 1 \leq j < k \\ \underline{A}_{j,j-k+l} = \alpha_l \text{id} - \beta_l A\tau & k \leq j \leq N_t, \quad 0 \leq l \leq k \\ \underline{A}_{j,j-1} = -\text{id} & N_t < j \leq 2N_t \end{cases}$$

$$\bar{b} = \begin{cases} \bar{b}_0 = x_0 \\ \bar{b}_j = bh & 1 \leq j < k \\ \bar{b}_j = \sum_{l=0}^k \beta_l bh & k \leq j \leq N_t \\ \bar{b}_j = 0 & N_t < j \leq 2N_t. \end{cases}$$

Assume there are oracles O_A and $O_{\mathcal{F}}$ such that $O_A|j, l\rangle|z\rangle = |j, l\rangle z \oplus A_{j,l}$ and $O_{\mathcal{F}}|j, l\rangle = |j, f(j, l)\rangle$ where $A_{j,l}$ is in binary representation and $f(j, l)$ is the l -th non-zero in column j . A similar oracle for the l -th non-zero in row j is also needed.

Lemma 79. *There holds $\|\underline{A}\| \lesssim 1$ if $\tau \lesssim 1/\|A\|$.*

Proof. We write \underline{A} as sum of block diagonal matrices

$$\underline{A} = \sum_{k=0}^l \underline{A}_k,$$

with norm

$$\begin{aligned}\|\underline{A}_0\| &\leq \max\{1, |\alpha_k| + |\beta_k|h\|A\|\}, \\ \|\underline{A}_1\| &\leq \max\{1 + h\|A\|, |\alpha_{k-1}| + |\beta_{k-1}|h\|A\|\}, \\ \|\underline{A}_l\| &\leq |\alpha_l| + |\beta_l|h\|A\|, \quad \forall 2 \leq l \leq k.\end{aligned}$$

Thus the norm of \underline{A} is bounded by

$$\|\underline{A}\| \leq \sum_{l=0}^k \|\underline{A}_l\| \lesssim k \lesssim 1.$$

□

Lemma 80. Assume that $A = VDV^{-1}$ with eigenvalues λ_i such that $|\arg(-\lambda_i)| \leq \alpha$. Furthermore, assume the multi-step method is $A(\alpha)$ -stable ($S \supseteq \{\lambda \in \mathbb{C} : |\arg(-\lambda)| < \alpha, \lambda \neq 0\}$). Then,

$$\|\underline{A}^{-1}\| \lesssim N_t \kappa_V,$$

where $\kappa_V := \|V\|\|V^{-1}\|$ is the condition number of V .

Proof. Let \underline{V} denote the block diagonal matrix

$$\underline{V} = \begin{pmatrix} V & & & \\ & V & & \\ & & \ddots & \\ & & & V \end{pmatrix}$$

and \underline{D} the matrix \underline{A} where we replace A by D . Then, $\underline{A} = \underline{V}\underline{D}\underline{V}^{-1}$ and $\|\underline{A}^{-1}\| \leq \kappa_V \|\underline{D}^{-1}\|$.

It remains to estimate $\|\underline{D}^{-1}\|$. We write \underline{D} as

$$\underline{D} = \sum_{l=0}^k \underline{D}_l$$

with (off-diagonal) block-matrices \underline{D}_l . Depending on l we have

$l = 1$:

$$\underline{D}^{-1} = (\underline{D}_0 + \underline{D}_1)^{-1} = \underline{D}_0^{-1}(\text{id} + \underline{D}_1\underline{D}_0^{-1})^{-1} = \underline{D}_0^{-1} \sum_{k=0}^{\infty} (-\underline{D}_1\underline{D}_0^{-1})^k$$

Note that $\underline{D}_1\underline{D}_0^{-1}$ is of the form

$$\underline{D}_1\underline{D}_0^{-1} = \begin{pmatrix} 0 & & & \\ \star & \ddots & & \\ & \ddots & \ddots & \\ & & \star & 0 \end{pmatrix}$$

and therefore $\|\underline{D}_1\underline{D}_0^{-1}\| = 0$ for $k \geq 2N_t$. Furthermore, the off-diagonal entries have the form

$$(1 + C\tau)^k \lesssim 1, \quad \forall k \leq \frac{1}{\tau} = N_t$$

and hence, $\|\underline{D}^{-1}\| \lesssim N_t$.

$l > 1$: (sketch)

$\underline{D}y = r$ corresponds to the discretization of the system

$$y^{(j)'}(t) = \lambda_j y^{(j)}(t) + r^{(j)}(t).$$

Since the method is α -stable, the numerical approximation $y_i^{(j)}$ cannot grow unless forced by r^j . Let

$$\left(y_i^{(j,l)}\right)_{i=1}^{N_t}$$

denote the solution with right hand side

$$\left(r_i^{(j)}\delta_{ik}\right)_{i=1}^{N_t}$$

and initial condition $x_0\delta_{k0}$, i.e.,

$$y_i^{(j)} = \sum_{k=0}^{N_t} y_i^{(j,k)}.$$

Stability shows $|y_i^{(j,k)}| \lesssim |r_k^{(j)}|$ for $i \geq k$ and hence

$$\|y^{(j)}\| \lesssim \sum_{k=0}^{N_t} \sqrt{N_t - k} |r_k^{(j)}| \lesssim N_t \underbrace{\sqrt{\sum_{k=0}^{N_t} |r_k^{(j)}|^2}}_{\|r^{(j)}\|}$$

and

$$\|y\| \lesssim N_t \|r\|.$$

□

Theorem 81. *Under the above assumptions, the HHL-algorithm produces a state proportional to*

$$|\psi\rangle = \sum_{j=0}^{N_t} |j\rangle |x_j\rangle$$

with error ε in

$$\mathcal{O}\left(\log N_x s^{\frac{q}{2}} (\|A\|T)^{2+\frac{2}{p}} \kappa_V^5 \left(\|x_0\| + \frac{\|b\|}{\|A\|}\right) \varepsilon^{-2}\right)$$

calls to the oracles for A, b , and x_0 .

Theorem 82. [2] *Suppose $A = VDV^{-1}$ with $D \leq 0$. Assume $A(t) = A$ and $b(t) = b$. There exists a quantum algorithm that produces*

$$\frac{x(T)}{\|x(T)\|}$$

up to an error ε with

$$\mathcal{O}\left(\kappa_V s \rho T \|A\| \text{poly}\left(\log\left(\kappa_V s \rho \beta T \frac{\|A\|}{\varepsilon}\right)\right)\right)$$

query calls, where

$$\rho := \max_{t \in [0, T]} \frac{\|x(t)\|}{\|x(T)\|}$$

$$\beta := \frac{\|x_0\| + T\|b\|}{\|x(T)\|}.$$

REFERENCES

- [1] Dominic W Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47(10):105301, February 2014.
- [2] Dominic W. Berry, Andrew M. Childs, Aaron Ostrander, and Guoming Wang. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Communications in Mathematical Physics*, 356(3):1057–1081, October 2017.