Quantum Computing Notes

Table of contents

1 Notation **1**

2 Basics of Quantum Mechanics **1**
2.1 Quantum State . . . 1
2.2 Evolution of a quantum state . . . 2
2.3 Observables and measurements . . . 2

3 Quantum Computer **3**
3.1 Qubit . . . 3
3.1.1 Pauli gates . . . 3
3.1.2 Measurement of one qubit . . . 4
3.1.3 Other important one-qubit gates . . . 4
3.2 Collection of many qubits or the quantum register . . . 4
3.3 No-Cloning Theorem . . . 5
3.4 Gates . . . 6
3.5 Summary of the common quantum gates . . . 6
3.5.1 Constant gates . . . 6
3.5.2 Parametrised gates . . . 7
3.6 Quantum circuit . . . 8
3.6.1 Qubits and Wires . . . 8
3.6.2 Gates . . . 8
3.6.3 Measurement . . . 9
3.6.4 Complexity of a quantum algorithm . . . 9

4 Quantum Fourier Transform **10**
4.1 Discrete Fourier Transform . . . 10
4.2 Quantum Fourier transform . . . 11
4.3 Implementation . . . 11
4.3.1 Controlled rotation . . . 11
4.3.2 QFT circuit . . . 12

5 Quantum Phase Estimation **14**
5.1 Circuit . . . 15
5.2 Limitations and sources of error . . . 17

1 Notation

$\|\cdot\|$ denotes the Euclidean norm on $\mathbb{C}^n$. * denotes the complex conjugate.

Definition 1.1 (Kronecker product)**.** For $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{p\times q}$, the matrix $A \otimes B$ is the Kronecker product of A and B defined by

$$(A \otimes B)_{ij,k\ell} = A_{ik}B_{j\ell}, \quad \text{for } 1 \le i \le m, 1 \le j \le p, 1 \le k \le n, 1 \le \ell \le q.$$

Proposition 1.1. *Let A, B, C, D be matrices of compatible sizes. The following assertions are true*

- *for $\lambda_A, \lambda_B, \mu_C, \mu_D \in \mathbb{C}$,*

$$(\lambda_A A + \lambda_B B) \otimes (\mu_C C + \mu_D D) = \lambda_A\mu_C A \otimes C + \lambda_A\mu_D A \otimes D + \lambda_B\mu_C B \otimes C + \lambda_B\mu_D B \otimes D$$

- $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$
- $(A \otimes B)^* = A^* \otimes B^*$
- *if A and B are unitary matrices, then $A \otimes B$ is a unitary matrix*
- *if (U_A, Σ_A, V_A^*) and (U_B, Σ_B, V_B^*) be singular value decompositions of A and B, then $(U_A \otimes U_B, \Sigma_A \otimes \Sigma_B, V_A^* \otimes V_B^*)$ is a singular value decomposition of $A \otimes B$.*

The first three statements are consequences of the definition of a Kronecker product. The last two statements directly follow from the second and third statements.

Notation: for $A \in \mathbb{C}^{m\times n}$ and $p \in \mathbb{N}$, we will write $A^{\otimes p} = \underbrace{A \otimes \cdots \otimes A}_{p \text{ times}}$.

2 Basics of Quantum Mechanics

2.1 Quantum State

Definition 2.1 (Quantum state)**.** Let $N \in \mathbb{N}$. Let $\sim$ be the equivalence relation on $\mathbb{C}^N$ defined by

$$\forall\, \phi, \psi \in \mathbb{C}^N, \phi \sim \psi \Leftrightarrow \exists \lambda \neq 0 \in \mathbb{C}, \psi = \lambda\phi.$$

Let $\mathrm{P}\mathbb{C}^N$ be the quotient set $\mathbb{C}^N$ by $\sim$. A quantum state is an element of $\mathrm{P}\mathbb{C}^N$.

Remark 2.1 (Dirac notation). For $\psi \in \mathbb{C}^N, \psi \neq 0$, the corresponding quantum state is denoted by $|\psi\rangle$. For $\phi \in (\mathbb{C}^N)^*, \phi \neq 0$, the dual quantum state is denoted by $\langle\phi|$. The scalar product between $|\phi\rangle$ and $|\psi\rangle$ is denoted by $\langle\phi|\psi\rangle$. If $\phi, \psi \in \mathbb{C}^N$ are such that $\|\phi\| = \|\psi\| = 1$ then $\langle\phi|\psi\rangle = \sum_{i=1}^N \phi_i^*\psi_i$.

In the following, we will identify $|\psi\rangle$ to a normalised representing vector $\psi \in \mathbb{C}^N$. Hence we will write the coordinates of $|\psi\rangle$ in $\mathbb{C}^N$ as $\begin{bmatrix}\psi_1\\ \vdots \\ \psi_N\end{bmatrix}$, where $\sum_{i=1} |\psi_i|^2 = 1$. Note that by definition of the equivalence class, all vector representations of the form $\begin{bmatrix}e^{\mathrm{i}\theta}\psi_1\\ \vdots \\ e^{\mathrm{i}\theta}\psi_N\end{bmatrix}$, for $\theta \in \mathbb{R}$, are equivalent.

Remark 2.2. For $|\phi\rangle, |\psi\rangle$ quantum states, $|\psi\rangle\langle\phi|$ denotes the projection onto ψ parallel to ϕ and is identified with a matrix $\mathbb{C}^{N\times N}$.

2.2 Evolution of a quantum state

For $|\psi\rangle$, $|\phi\rangle$ quantum states, there exists a matrix $U \in \mathbb{C}^{N\times N}$ such that $|\psi\rangle = U|\phi\rangle$.

Definition 2.2 (Schrödinger equation)**.** Let $(H(t))_{t\geq 0}$ be a family of Hermitian matrices, and $|\psi_0\rangle$ a quantum state. The Schrödinger evolution of a quantum state $t \mapsto \psi(t)$ is defined as the solution to the equation

$$\begin{cases} \mathrm{i}\dfrac{\partial}{\partial t}\psi(t) = H(t)\psi(t), \quad t \geq 0 \\ \psi(0) = \psi_0 \end{cases}$$

One can check that if $\|\psi_0\| = 1$ then for all $t \geq 0$, $\|\psi(t)\| = 1$.

If $t \mapsto H(t)$ is constant and equal to H, then the solution to the Schrödinger equation is $\psi(t) = e^{\mathrm{i}Ht}\psi_0$. This shows that for Hermitian matrices H, the matrix $e^{\mathrm{i}H}$ is a unitary matrix. Conversely, one can show that any unitary matrix U can be written as $e^{\mathrm{i}H}$ for some Hermitian matrix H.

2.3 Observables and measurements

Definition 2.3 (Observable)**.** An observable $O \in \mathbb{C}^{N\times N}$ is a Hermitian matrix. For $|\psi\rangle$ a quantum state, the expectation value denoted by $\langle\psi|O|\psi\rangle$ is equal to $\langle\psi, O\psi\rangle$.

Let O be an observable. Then O can be written in the basis of its eigenvectors as $O = \sum_{m=1}^{N} \lambda_m P_m$, where for $1 \leq m \leq N$, P_m orthogonal projector, $P_m = |u_m\rangle\langle u_m|$. A quantum state $|\psi\rangle$ can be written in the basis (u_m) as $\psi = \sum_{m=1}^{N} p_m u_m$, with $\sum_{m=1}^{N} |p_m|^2 = 1$. Then the expectation value of the observable is $\langle\psi|O|\psi\rangle = \mathrm{Tr}(|\psi\rangle\langle\psi|O) = \sum_{m=1}^{N} \lambda_m |p_m|^2$.

Definition 2.4 (Measurement)**.** Let $|\psi\rangle$ be a quantum state on $\mathbb{C}^N$ and $O \in \mathbb{C}^{N\times N}$ be an observable. Let (u_m) be an orthonormal basis of eigenvectors of O. Let $(p_m) \in \mathbb{C}^N$ such that $|\psi\rangle = \sum_{k=1}^{N} p_m u_m$ (with $\sum_{k=1}^{N} |p_m|^2 = 1$). The measurement operator $\mathcal{M}$ for the observable O is the operator such that

$$\mathcal{M}|\psi\rangle = u_m, \quad \text{with probability } |p_m|^2.$$

The measurement is, for a general quantum state $|\psi\rangle$ and observable O, a nonlinear operator. The output of the measurement is probabilistic and depends on the expansion of the quantum state ψ in the basis of eigenvectors of the observable O.

Measurements are the only way to have access to the information contained in the quantum state $|\psi\rangle$. For efficient quantum computations, it is thus important that the quantum state obtained after quantum computations has contributions in a few eigenvectors of the measured observable.

3 Quantum Computer

3.1 Qubit

Classical bit: $b \in \{0, 1\}$

Definition 3.1 (Quantum bit (qubit))**.** A quantum bit (or qubit) is a quantum state on $\mathbb{C}^2$.

The canonical basis associated to the space of quantum bits on $\mathbb{C}^2$ is denoted by $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$.

A quantum bit $|\psi\rangle \in \mathbb{C}^2$ satisfies $|\psi\rangle = \psi_1|0\rangle + \psi_2|1\rangle$ with $|\psi_1|^2 + |\psi_2|^2 = 1$ (up to a global phase).

A qubit can be parametrised by two angles Representation on the Bloch sphere:

$$|\psi\rangle = e^{i\gamma}(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle),$$

where $0 \leq \theta \leq \pi$, $0 \leq \varphi < 2\pi$ (recall that the global phase $e^{i\gamma}$ is not relevant).

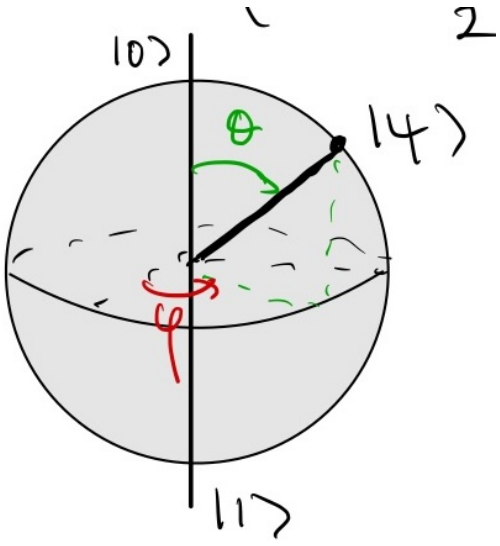


Figure 1: Bloch sphere

3.1.1 Pauli gates

Definition 3.2. Let $X, Y, Z \in \mathbb{C}^{2\times 2}$ be the Pauli matrices defined by

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Notice that the Pauli matrices X, Y, Z are Hermitian. They satisfy the commutation relations $[X, Y] = iZ, [Y, Z] = iX, [Z, X] = iY$. One can show that (I, X, Y, Z) is in fact a basis of the $\mathbb{R}$-vector space of Hermitian matrices. This means that unitary evolutions of quantum states can be written as $e^{\mathrm{i}\theta X + \mathrm{i}\phi Y + \mathrm{i}\gamma Z}$ (as the global phase can be ignored).

This motivates the introduction of parametrised gates

$$e^{\mathrm{i}\theta/2X} = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \quad e^{\mathrm{i}\theta/2Y} = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \quad e^{\mathrm{i}\theta/2(Z-I)} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}.$$

As the matrices X, Y, Z do **not** commute, we do **not** have $e^{\mathrm{i}\theta X + \mathrm{i}\phi Y + \mathrm{i}\gamma Z} = e^{\mathrm{i}\theta X} e^{\mathrm{i}\phi Y} e^{\mathrm{i}\gamma Z}$. However, using what is called *Trotter splitting*, we have that

$$e^{\mathrm{i}\theta X + \mathrm{i}\phi Y + \mathrm{i}\gamma Z} = \lim_{n\to\infty} \prod_{i=1}^{n} e^{\mathrm{i}\theta/nX} e^{\mathrm{i}\phi/nY} e^{\mathrm{i}\gamma/nZ},$$

which motivates the use of the parametrised gates introduced above.

3.1.2 Measurement of one qubit

Measurements of a qubit are taken for the observable Z (except otherwise stated). The computational basis $|0\rangle, |1\rangle$ is the basis that diagonalises Z, thus the measurement of a quantum state $|\psi\rangle = \psi_0|0\rangle + \psi_1|1\rangle$, gives $|0\rangle$ with probability $p_0 = |\psi_0|^2$ and $|1\rangle$ with probability $p_1 = |\psi_1|^2 = 1 - p_0$.

This means that the measurement is a Bernoulli random variable. The variance related to the estimation of the probability p_0 is equal to $p_0(1 - p_0)$.

To estimate p_0, one uses the empirical estimator $\widehat{p}_0 = \frac{\text{number of measured } |0\rangle}{\text{total number of samples}}$. If $\mathcal{N}$ is the total number of samples, and assuming that each measurement is independent, the variance of the empirical estimator is $\frac{p_0(1-p_0)}{\mathcal{N}}$. To guarantee that the standard deviation of the empirical estimator is below some threshold ϵ, we must have $\frac{p_0(1-p_0)}{\mathcal{N}} \leq \epsilon^2$, thus $\mathcal{N} \geq \frac{p_0(1-p_0)}{\epsilon^2}$.

For a small probability p_0, an accurate estimation of the probability p_0 requires to have $\epsilon \ll p_0$, which means that the number of samples needs to be much larger than $\frac{1}{p_0}$. This is another important restriction in designing efficient algorithms in quantum computing.

3.1.3 Other important one-qubit gates

Function	Description	Matrix Representation	Usage
H	Hadamard gate	$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$	`H(qubit)`
S	Phase gate (or S gate)	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	`S(qubit)`
T	T gate	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$	`T(qubit)`

3.2 Collection of many qubits or the quantum register

A classical register is a collection of classical bits $b = (b_1, \dots, b_n)$, $b_i \in \{0, 1\}$, $i = 1, \dots, n$.

Definition 3.3 (Quantum register)**.** A quantum register on n qubits is a quantum state on $\bigotimes_{i=1}^{n} \mathbb{C}^2$.

The space $\bigotimes_{i=1}^{n} \mathbb{C}^2$ can be identified with $\mathbb{C}^{2^n}$. Its canonical basis, also called the *computational basis* is the set of the canonical basis of $\mathbb{C}^{2^n}$. These vectors are denoted by $|q_1 \dots q_n\rangle$, where $(q_k) \in \{0, 1\}$. They correspond to the j-th canonical vector, $j \in \{0, \dots, 2^n - 1\}$, where $j = \sum_{k=0}^{n-1} q_{k+1} 2^k$ (*i.e.* (q_k) is the binary decomposition of j). It is also customary to denote the quantum state $|j\rangle$.

Example 3.1 (Kronecker products of qubits)**.** If $|\psi_1\rangle \in \mathbb{C}^2$, $|\psi_2\rangle \in \mathbb{C}^2$, $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ is a 2-qubit state.

$$|0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Remark 3.1 (Entanglement)*.* The quantum states seen so far are given as Kronecker products. The power of quantum computing lies in the manipulation of *entangled* states, that **cannot** be obtained from Kronecker products of quantum states. For example, the *Bell state* $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$ cannot be written as a Kronecker product of one qubit state.

Remark 3.2 (Superposition and entanglement)*.* Entangled states are linear combinations of several Kronecker products of quantum states. Linear combinations of tensor product states are often called *superpositions*. A superposition of product states is not necessarily entangled. The state

$$(H|0\rangle) \otimes \cdots \otimes (H|0\rangle) = \frac{1}{2^{n/2}} \sum_{q_0, \dots, q_{n-1} \in \{0,1\}} |q_0 \dots q_n\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^n - 1} |k\rangle,$$

is a superposition of product states but is not entangled because it can be written as a product state (but in a different basis.)

3.3 No-Cloning Theorem

Theorem 3.1 (No cloning). *Let $|s\rangle \in \mathbb{C}^n$ be a quantum state. There is no unitary matrix $U \in \mathbb{C}^{n^2 \times n^2}$ such that for any $|\psi\rangle \in \mathbb{C}^n$, we have*

$$U|\psi\rangle \otimes |s\rangle = |\psi\rangle \otimes |\psi\rangle.$$

Proof. Let U be such that for any $|\psi\rangle \in \mathbb{C}^n$, we have

$$U|\psi\rangle \otimes |s\rangle = |\psi\rangle \otimes |\psi\rangle.$$

Let $|\phi\rangle \in \mathbb{C}^n$ such that $\langle\phi|\psi\rangle \neq 0$ or 1. Then $U|\phi\rangle \otimes |s\rangle = |\phi\rangle \otimes |\phi\rangle$. Thus we have

$$\begin{aligned} \langle\phi| \otimes \langle\phi|\psi\rangle \otimes |\psi\rangle &= \langle s| \otimes \langle\phi|U^*U|s\rangle \otimes |\psi\rangle \\ \langle\phi|\psi\rangle^2 &= \langle\phi|\psi\rangle. \end{aligned}$$

This means that either $\langle\phi|\psi\rangle = 0$ or $\langle\phi|\psi\rangle = 1$: contradiction. □

This theorem has a crucial consequence in the design of quantum algorithms: a vast majority of classical algorithms are iterative (Newton, root finding, conjugate-gradient…) which requires to store a copy of some iterate. In quantum computing, as it is impossible to copy arbitrary quantum states, the design of quantum algorithms cannot be a simple transposition of efficient classical algorithms.

Following the proof of the no-cloning theorem, it is however possible to copy *some* quantum states:

- for known quantum states $|\psi\rangle$, $|s\rangle$, if a unitary U_ψ such that $U_\psi|s\rangle = |\psi\rangle$, then $(I \otimes U_\psi)(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle$;
- for quantum state $|j\rangle$ in the computational basis, then any state $|k\rangle$ in the computational basis can be copied, *i.e.* for any $0 \leq j \leq n-1$, there is U such that for any $0 \leq k \leq n-1$, we have

  $$U|k\rangle \otimes |j\rangle = |k\rangle \otimes |k\rangle.$$

  For $n = 4$ and $j = 0$, this unitary is the CNOT gate: $\text{CNOT}(|k\rangle \otimes |0\rangle) = |k\rangle \otimes |k\rangle$. One can check that if $|\psi\rangle = \psi_0|0\rangle + \psi_1|1\rangle$ with $\psi_0\psi_1 \neq 0$, then $\text{CNOT}|\psi\rangle \otimes |0\rangle = \psi_0|00\rangle + \psi_1|11\rangle \neq |\psi\rangle \otimes |\psi\rangle$.

3.4 Gates

Linear operations on quantum registers are called *gates*. These linear operations are necessarily unitary operators.

In classical computing, any operation on classical registers can be implemented using combinations of only a few gates (for example, using only NAND or NOR gates).

In quantum computing, any operation can be implemented using only one or two-qubit gates (*i.e.* gates that operate only on one or two qubits).

It is possible to further restrict this set, if we give up on the *exact* representation of the unitary U, *i.e.* for a given accuracy $\epsilon > 0$, we want to find $(U_1, \ldots, U_m)$ such that

- $\|U - U_m U_{m-1} \cdots U_1\| < \epsilon$,
- $(U_k)_{1 \leq k \leq m}$ belongs to a (small) set of universal gates.

There are multiple choices of universal gates such as $\{H, T, \text{CNOT}\}$ or $\{H, \text{Toffoli}\}$ (see their definitions below). A natural question is to check whether there are sets of universal gates that are better than others. The answer is given by the following theorem.

Theorem 3.2 (Solovay-Kitaev). *Let $\mathcal{S}, \mathcal{T}$ be two sets of universal gates that are closed under inverses. Then any m-gate circuit using the gate set $\mathcal{S}$ can be implemented to precision ϵ using a circuit of $\mathcal{O}(m \,\text{polylog}(m/\epsilon))$ gates from the gate set $\mathcal{T}$.*

Asymptotically, the Solovay-Kitaev theorem states that any choice of sets of universal gates leads to a comparable number of quantum operations.

3.5 Summary of the common quantum gates

Here is a list of common quantum gates as well as the corresponding command in MyQLM. A quantum algorithm is simply a series of quantum gates applied to a initial quantum state (which is generally the state $|0\rangle$).

3.5.1 Constant gates

Function	Description	Matrix Representation	Usage
X	Pauli-X gate, NOT gate	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	`X(qubit)`
Y	Pauli-Y gate	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	`Y(qubit)`
Z	Pauli-Z gate	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	`Z(qubit)`
H	Hadamard gate	$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$	`H(qubit)`
S	Phase gate (or S gate)	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	`S(qubit)`
T	T gate	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$	`T(qubit)`
CNOT	CNOT (Controlled NOT) gate	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	`CNOT(control_qubit, target_qubit)`
CCNOT	Toffoli gate (or CCNOT gate).	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	`CCNOT(control_qubit1, control_qubit2, target_qubit)`
CSIGN	Controlled Sign or C-Z gate	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	`CSIGN(control_qubit, target_qubit)`
SWAP	SWAP gate	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	`SWAP(qubit1, qubit2)`
SQRTSWAP	Square Root of SWAP gate. It creates a superposition of swapped and non-swapped states.	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1+i) & \frac{1}{2}(1-i) & 0 \\ 0 & \frac{1}{2}(1-i) & \frac{1}{2}(1+i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	`SQRTSWAP(qubit1, qubit2)`

Function	Description	Matrix Representation	Usage
ISWAP	iSWAP gate. It swaps the states of two qubits with a phase factor of i.	$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & i & 0\\ 0 & i & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	`ISWAP(qubit1, qubit2)`

In this list, the CNOT, CSIGN, SWAP, SQRTSWAP and ISWAP are unitary operators acting on two qubits. It can be proved that these unitary operations cannot be written as the Kronecker product of two one-qubit gates.

The Toffoli gate is a 3-qubit gate that also cannot be written as a Kronecker product of one-qubit gates.

3.5.2 Parametrised gates

Function	Description	Matrix Representation	Usage
RX(θ)	Rotation around the X-axis by an angle θ.	$\begin{bmatrix}\cos(\theta/2) & -i\sin(\theta/2)\\ -i\sin(\theta/2) & \cos(\theta/2)\end{bmatrix}$	`RX(theta)(qubit)`
RY(θ)	Rotation around the Y-axis by an angle θ.	$\begin{bmatrix}\cos(\theta/2) & -\sin(\theta/2)\\ \sin(\theta/2) & \cos(\theta/2)\end{bmatrix}$	`RY(theta)(qubit)`
RZ(θ)	Rotation around the Z-axis by an angle θ.	$\begin{bmatrix}e^{-i\theta/2} & 0\\ 0 & e^{i\theta/2}\end{bmatrix}$	`RZ(theta)(qubit)`
PH(φ)	Phase gate that leaves $\|0\rangle$ unchanged and maps $\|1\rangle$ to $e^{i\varphi}\|1\rangle$.	$\begin{bmatrix}1 & 0\\ 0 & e^{i\varphi}\end{bmatrix}$	`PH(phi)(qubit)`

3.6 Quantum circuit

Quantum circuits are graphical representations of quantum algorithms. They show how quantum gates are applied to qubits over time.

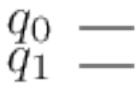
3.6.1 Qubits and Wires

- **Qubits**: each qubit is represented by one horizontal line (a wire) in a circuit diagram.
- **Initial State**: qubits start in the $|0\rangle$ state unless otherwise specified.

Example in MyQLM

The following piece of code displays a quantum register initialised (by default) at $|00\rangle$.

```
from qat.lang.AQASM import Program
from qat.core.printer import plot_in_notebook
import matplotlib.pyplot as plt
my_program = Program()
qregister = my_program.qalloc(2) #allocates 2 qubits
circuit = my_program.to_circ()
plot_in_notebook(circuit,fmt='pdf') #circuit.display() works too
```



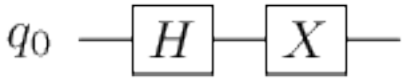
3.6.2 Gates

Gates are represented by boxes or dots in the quantum circuit. Their applications are read from left to right.

Example for one qubit in MyQLM

The following piece of code displays the quantum algorithm $XH|0\rangle$.

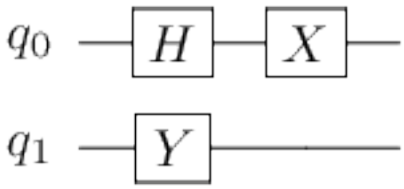
```
from qat.lang.AQASM import Program
from qat.lang.AQASM import H, X #import the Hadamard gate and the CNOT gate
my_program = Program()
qregister = my_program.qalloc(1) #allocates 2 qubits
H(qregister[0]) #first apply Hadamard gate
X(qregister[0]) #then apply X gate
circuit = my_program.to_circ()
plot_in_notebook(circuit,fmt='png') #circuit.display() works too
```



Example for two qubits in MyQLM

The following piece of code displays the quantum algorithm $(I \otimes Y)(X \otimes I)(H \otimes I)|00\rangle$.

```
from qat.lang.AQASM import Program
from qat.lang.AQASM import H, X, Y #import the Hadamard gate and the CNOT gate
my_program = Program()
qregister = my_program.qalloc(2) #allocates 2 qubits
H(qregister[0]) #first apply Hadamard gate on the first qubit
X(qregister[0]) #then apply X gate on the first qubit
Y(qregister[1]) #then apply X gate on the first qubit
circuit = my_program.to_circ()
plot_in_notebook(circuit,fmt='pdf') #circuit.display() works too
```



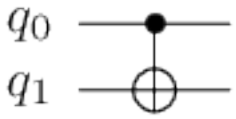
Note that since $(I \otimes Y)$ and $(X \otimes I)(H \otimes I)$, the quantum operations are squashed to the left.

CNOT gate in MyQLM

The following piece of code displays the quantum algorithm CNOT$|00\rangle$.

```
from qat.lang.AQASM import Program
from qat.lang.AQASM import CNOT #import the Hadamard gate and the CNOT gate
```

```
my_program = Program()
qregister = my_program.qalloc(2) #allocates 2 qubits
CNOT(qregister[0],qregister[1]) #first apply the CNOT gate with the 1st qubit as the control and 2nd as th
circuit = my_program.to_circ()
plot_in_notebook(circuit,fmt='pdf') #circuit.display() works too
```



SWAP gate in MyQLM

The following piece of code displays the quantum algorithm SWAP$|00\rangle$.

```
from qat.lang.AQASM import Program
from qat.lang.AQASM import SWAP #import the Hadamard gate and the CNOT gate
my_program = Program()
qregister = my_program.qalloc(2) #allocates 2 qubits
SWAP(qregister[0],qregister[1]) #first apply the CNOT gate with the 1st qubit as the control and 2nd as th
circuit = my_program.to_circ()
plot_in_notebook(circuit,fmt='pdf') #circuit.display() works too
```

3.6.3 Measurement

3.6.4 Complexity of a quantum algorithm

Remark 3.3 (Clifford gates). Clifford gates are quantum gates that stabilises the group formed by Kronecker products of Pauli matrices (*i.e.* if C is a Clifford gate and P a Kronecker product of Pauli matrices, then CPC^* is a Kronecker product of Pauli matrices). It can be checked that the Hadamard gate, the CNOT gate are Clifford gates, but not the T gate. Since these gates stabilise the group of Kronecker product of Pauli matrices, they can be classically simulated in polynomial time.

The complexity of quantum algorithms is estimated in three different ways:

- the depth of the circuit: the depth of the circuit is the maximal number of gates along any path from an input and an output. It is a reasonable depiction of the total run time of a quantum simulation. One of the main challenge in the design of physical quantum computers is in maintaining the *coherence time* of the quantum system. The coherence time is the longest period during which the quantum system accurately preserves the outcome of a quantum algorithm. Nowadays, the maximal coherence time is of the order of 0.1 ms and the total number of operations that can be performed is of the order $10^2, 10^3$.
- the number of two qubit gates in the quantum algorithm: CNOT gates have a probability of failure of about 1%, whereas one-qubit gates typically have failure rates of about 0.1% (or sometimes lower than that). The accuracy of a quantum algorithm depends heavily on the number of two-qubit gates.
- the number of T gates in the quantum algorithm: to a lesser extend, the complexity of a quantum algorithm can also be estimated by counting the number of T gates. If a quantum algorithm is expressed in the universal gate set $\{H, T, \text{CNOT}\}$, the only non-Clifford gate is the T gate. As Clifford gates are classically easy to emulate, the quantum advantage can be estimated as the total number of T gates.

4 Quantum Fourier Transform

The quantum Fourier transform is the quantum implementation of the discrete Fourier transform on n qubits.

4.1 Discrete Fourier Transform

We recall the definition of the discrete Fourier transform.

Definition 4.1. For $N \in \mathbb{N}^*$, the **Discrete Fourier Transform (DFT)** $\mathcal{F}_N$ is defined by:

$$\forall 0 \leq j \leq N-1, \quad (\mathcal{F}_N x)_j = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} e^{2i\pi \frac{qj}{N}} x_q \quad \text{where} \quad x = \begin{bmatrix} x_0 \\ \vdots \\ x_{N-1} \end{bmatrix}.$$

In the canonical basis, the DFT is represented by the matrix:

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix}, \quad \text{where} \quad \omega = e^{2i\pi/N}.$$

Proposition 4.1. *The matrix F_N satisfies $F_N^* F_N = I_N$ (identity), since:*

$$(F_N^* F_N)_{ij} = \frac{1}{N} \sum_{q=0}^{N-1} (\omega^*)^{iq} \omega^{jq} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Thus, F_N is an unitary matrix.

For $x \in \mathbb{C}^N$, $F_N x$ can be computed in $O(N \log N)$ using the **FFT algorithm** (faster than the naive $O(N^2)$ matrix-vector multiplication of a dense matrix).

4.2 Quantum Fourier transform

Definition 4.2 (Quantum Fourier Transform (QFT))**.** The quantum Fourier transform on n qubits is defined as the quantum gate such that for a quantum state $|\psi\rangle = \sum_{j=0}^{2^n-1} c_j |j\rangle$:

$$\mathrm{QFT}_n |\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \left(\sum_{j=0}^{2^n-1} c_j e^{2\pi i jk/2^n} \right) |k\rangle,$$

where $(|j\rangle)_{0 \leq j \leq 2^n-1}$ is the computational basis of $\mathbb{C}^{2^n}$.

The QFT is exactly the implementation in the computational basis of an n-qubit quantum computer of the discrete Fourier transform.

For a quantum state $|j\rangle$ of the computational basis, we have

$$\mathrm{QFT}_n |j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i jk/2^n} |k\rangle.$$

Example 4.1. For one qubit, the QFT is the matrix $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. This coincides with the Hadamard matrix.

4.3 Implementation

4.3.1 Controlled rotation

Definition 4.3 (Controlled rotation)**.** Let $U \in \mathbb{C}^{2^n \times 2^n}$ be a unitary matrix. The controlled rotation for the unitary U, denoted C-U $\in \mathbb{C}^{2^{n+1} \times 2^{n+1}}$ is the matrix such that

$$\text{C-U} = |0\rangle\langle 0| \otimes I_{2^n} + |1\rangle\langle 1| \otimes U.$$

The first qubit is called the *control* qubit and the other qubits are called the *target* qubits.

One can check that the matrix C-U is a unitary matrix, and as such defines a quantum gate. One way to see that is to write the matrix in the computational basis

$$\text{C-U} = \begin{bmatrix} I_{2^n} & \\ & U \end{bmatrix},$$

where it is clear that C-U is unitary.

Remark 4.1. The CNOT gate is a controlled X gate.

Another way to see the controlled rotation is when it is applied to $|q\rangle \otimes |\psi\rangle$, where $|q\rangle$ is either $|0\rangle$ or $|1\rangle$. Then

$$\text{C-U}|q\rangle \otimes |\psi\rangle = |q\rangle \otimes U^q|\psi\rangle.$$

Proposition 4.2. *Let $U \in \mathbb{C}^{2^n \times 2^n}, V \in \mathbb{C}^{2^m \times 2^m}$ be unitary matrices. The controlled rotation* C-U $\otimes$ V $\in \mathbb{C}^{2^{n+m+1} \times 2^{n+m+1}}$ *is the matrix product of the controlled rotations*

$$\text{C-(U} \otimes \text{V)} = \text{C-(I}_{2^n} \otimes \text{V)C-(U} \otimes \text{I}_{2^m}) = \text{C-(U} \otimes \text{I}_{2^m})\text{C-(I}_{2^n} \otimes \text{V}).$$

Proof. The equality is clear from the block decomposition of

$$\text{C-(U} \otimes \text{V)} = \begin{bmatrix} I_{2^{n+m}} & \\ & \text{U} \otimes \text{V} \end{bmatrix} = \begin{bmatrix} I_{2^{n+m}} & \\ & \text{I}_{2^n} \otimes \text{V} \end{bmatrix} \begin{bmatrix} I_{2^{n+m}} & \\ & \text{U} \otimes \text{I}_{2^m} \end{bmatrix}.$$

□

The first qubit is the control in the definition of the controlled qubit. This is arbitrary and controlled rotation can be defined for other qubits by sandwiching the controlled rotation with appropriate SWAPs.

Definition 4.4. Let $U \in \mathbb{C}^{2^n \times 2^n}$ be a unitary matrix. The controlled rotation with control qubit j for the unitary U, denoted C_j-U $\in \mathbb{C}^{2^{n+1} \times 2^{n+1}}$ is the matrix such that

$$\text{C}_j\text{-U} = \text{SWAP}_{1j}(\text{C-U})\text{SWAP}_{1j},$$

where SWAP_{1j} is the SWAP gate between qubits 1 and j defined by

$$\text{SWAP}_{1j}(|q_1 \dots q_{j-1} q_j q_{j+1} \dots q_n\rangle) = |q_j \dots q_{j-1} q_1 q_{j+1} \dots q_n\rangle, \quad \text{for any } q_1, \dots, q_n = 0, 1.$$

Suppose that U is a one-qubit gate, in general C-U $\neq$ C_2-U. An important case where it holds is the parametrised Z-rotation.

Proposition 4.3. *Let* $R_Z(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$. *Then* C-$R_Z(\theta) = \text{C}_2$-$R_Z(\theta)$.

Proof. We have $\text{SWAP}_{12} = \begin{bmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{bmatrix}$ (in the $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ basis). In the same basis, we have C-$R_Z(\theta) = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{i\theta} \end{bmatrix}$, thus the SWAP_{12} gate thus not affect the C-$R_Z(\theta)$ matrix. □

Definition 4.5 (Multi-controlled rotation)**.** Let $U \in \mathbb{C}^{2^n \times 2^n}$ be a unitary matrix. The multi-controlled rotation over m qubits, denoted C^m-U $\in \mathbb{C}^{2^{n+m} \times 2^{n+m}}$ is the matrix such that

$$\text{C}^\text{m}\text{-U} = \sum_{j=0}^{2^m-2} |j\rangle\langle j| \otimes I_{2^n} + |2^m - 1\rangle\langle 2^m - 1| \otimes U.$$

In the computational basis, a multi-controlled gate is written

$$\text{C}^\text{m}\text{-U} = \begin{bmatrix} I_{2^{m+n}-2^n} & \\ & U \end{bmatrix}$$

Remark 4.2. The Toffoli gate is a multi-controlled X gate over 2 qubits.

4.3.2 QFT circuit

Let $0 \leq j, k \leq 2^{n+1} - 1$. Let $(j_n, \dots, j_0)$ and $(k_n, \dots, k_0)$ be the binary representations of j, k (*i.e.* $j = \sum_{\ell=0}^n j_\ell 2^\ell$ and $k = \sum_{\ell=0}^n k_\ell 2^\ell$). Let $\overline{j_0} = \sum_{\ell=1} 2^{\ell-1} j_\ell$ and $\overline{k_n} = \sum_{\ell=0}^{n-1} k_\ell$.

We have

$$jk = (j_0 + 2\overline{j_0})(\overline{k_n} + k_n) = j_0 2^n k_n + j_0\overline{k_n} + 2^{n+1}\overline{j_0}k_n + 2\overline{j_0 k_n}.$$

We compute the (j, k) entry of the discrete Fourier transform:

$$\begin{aligned} \exp(\frac{2i\pi jk}{2^{n+1}}) &= \exp(i\pi j_0 k_n) \exp(\frac{2i\pi j_0\overline{k_n}}{2^{n+1}}) \exp(2i\pi\overline{j_0}k_n) \exp(\frac{2i\pi\overline{j_0}\overline{k_n}}{2^n}) \\ &= (-1)^{j_0 k_n} \exp(\frac{2i\pi j_0\overline{k_n}}{2^{n+1}}) \exp(\frac{2i\pi\overline{j_0}\overline{k_n}}{2^n}), \end{aligned}$$

where we have used that $j_0, \overline{j_0}, k_n, \overline{k_n}$ are integers.

We realise that the last coefficient is up to a normalisation constant the $(\overline{j_0 k_n})$ coefficient of the discrete Fourier transform on $\mathbb{C}^{2^n}$. This motivates the introduction of the matrix $\widetilde{F}_{2^n} = \widetilde{F}_{2^n} S$, where S is the flip of the matrix of the flip of the indices. This matrix can be implemented using $\lfloor n/2 \rfloor$ SWAP.

If $j_0 = 0$, for $k_n = 0$ or 1, we see that $(\widetilde{F}_{2^{n+1}})_{0\overline{j_0};k_n\overline{k_n}} = \frac{1}{\sqrt{2}}(\widetilde{F}_{2^n})_{\overline{j_0};\overline{k_n}}$.

Let $P \in \mathbb{C}^{2^n \times 2^n}$ be the diagonal and unitary matrix such that $P_{kk} = \exp(\frac{2i\pi k}{2^{n+1}})$. Then for $j_0 = 1$, we have that $(\widetilde{F}_{2^{n+1}})_{1\overline{j_0};k_n\overline{k_n}} = \frac{1}{\sqrt{2}}(-1)^{k_n}(\widetilde{F}_{2^n})_{\overline{j_0};\overline{k_n}} P$.

The matrix $\widetilde{F}_{2^{n+1}}$ has the following block structure in the $(|j_0\rangle|\overline{j_0}\rangle; |k_n\rangle|\overline{k_n}\rangle)$ basis

$$\widetilde{F}_{2^{n+1}} = \frac{1}{\sqrt{2}} \begin{bmatrix} \widetilde{F}_{2^n} & \widetilde{F}_{2^n} \\ \widetilde{F}_{2^n} P & -\widetilde{F}_{2^n} P \end{bmatrix}.$$

This block structure can be factorised in

$$\frac{1}{\sqrt{2}}\begin{bmatrix} \widetilde{F}_{2^n} & \widetilde{F}_{2^n} \\ \widetilde{F}_{2^n}P & -\widetilde{F}_{2^n}P \end{bmatrix} = \begin{bmatrix} \widetilde{F}_{2^n} & \\ & \widetilde{F}_{2^n} \end{bmatrix} \begin{bmatrix} I_{2^n} & \\ & P \end{bmatrix} \begin{bmatrix} I_{2^n}/\sqrt{2} & I_{2^n}/\sqrt{2} \\ I_{2^n}/\sqrt{2} & -I_{2^n}/\sqrt{2} \end{bmatrix}.$$

The first matrix is simply $I_2 \otimes \widetilde{F}_{2^n}$, the second matrix is a controlled-P gate and the last one is simply $H \otimes I_{2^n}$.

It suffices to determine the circuit for the rotation P in order to find the circuit of the QFT. As $P_{kk} = \exp(\frac{2i\pi k}{2^{n+1}})$, using the binary writing $(k_0, \dots, k_{n-1})$ of the integer k, we have that

$$P_{kk} = \prod_{\ell=0}^{n-1} \exp(\frac{i\pi k_\ell}{2^{n-\ell}}).$$

It is thus a Kronecker product of one qubit parametrised Z-rotations. Using Proposition 4.2, we deduce that the circuit can be written as

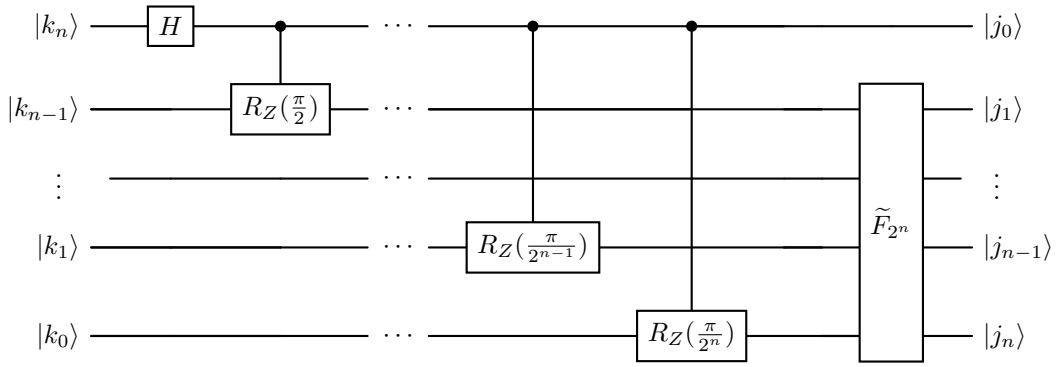


Figure 2: QFT circuit: first step

As the controlled rotations are parametrised-Z rotations, we have another (more common) writing of the algorithm as

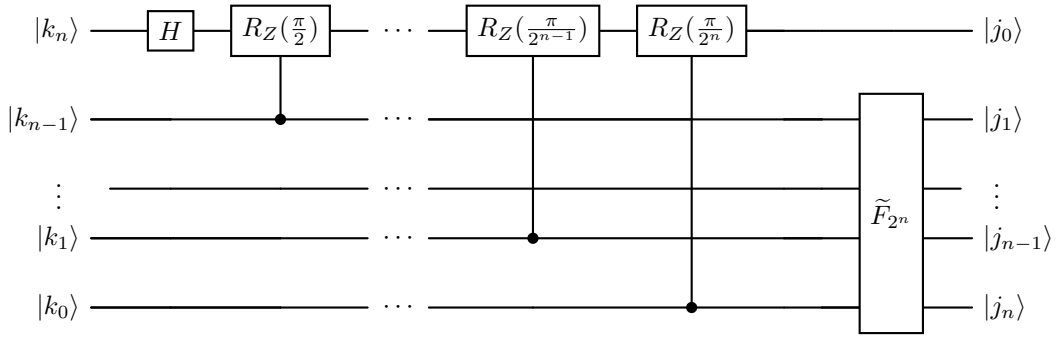


Figure 3: QFT circuit: first step

Performing the induction steps yields the full QFT circuit (not forgetting the SWAPs at the end):

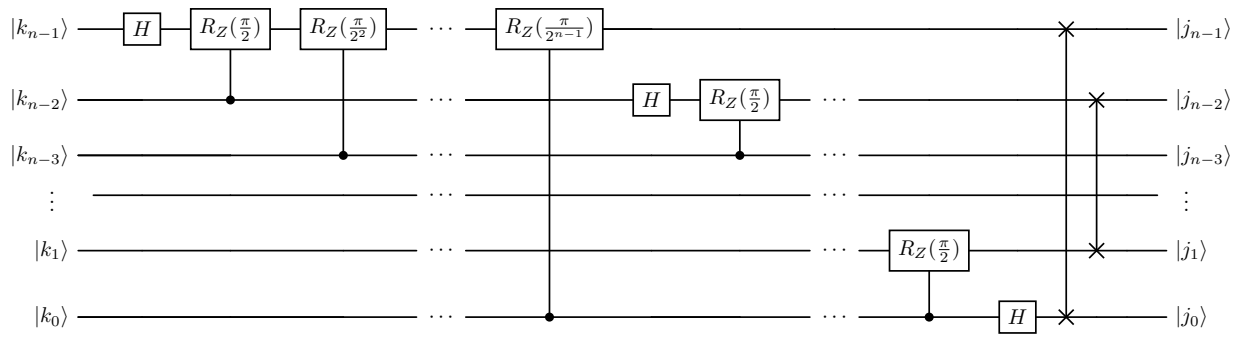


Figure 4: Full QFT circuit

Complexity: the QFT circuit has $\sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}$ controlled gates and $\frac{n}{2}$ SWAPS, hence the complexity of applying a QFT is of the order n^2. This means that the discrete Fourier transform on $\mathbb{C}^{2^n}$ is exponentially faster (in terms of elementary operations) on a quantum computer, compared to its classical version.

5 Quantum Phase Estimation

Assumptions: the unitary matrix $U \in \mathbb{C}^{2^n \times 2^n}$ has an eigenvalue $e^{2i\pi\theta}$ for a normalised eigenvector ψ such that there is $m \in \mathbb{N}$ such that $\theta = \frac{k}{2^m}$, with $0 \leq k \leq 2^m - 1$.

The goal of the quantum phase estimation (QPE) algorithm is to compute the eigenvalue $e^{2i\pi\theta}$.

A typical example of a unitary U for which the eigenvalue is sought are operators of the form e^{iHt} for a Hermitian matrix H.

Let $\mathcal{U} \in \mathbb{C}^{2^{n+m} \times 2^{n+m}}$ be the matrix defined by

$$\mathcal{U} = \sum_{j=0}^{2^m-1} |j\rangle\langle j| \otimes U^j.$$

A quick calculation shows that $\mathcal{U}$ is a unitary matrix.

The implementation of the matrix $\mathcal{U}$ can be simplified by realising that

$$\begin{aligned}
\mathcal{U} &= \sum_{j=0}^{2^m-1} |j\rangle\langle j| \otimes U^j \\
&= \sum_{j_0,\cdots,j_{m-1}} |j_0 \cdots j_{m-1}\rangle\langle j_0 \cdots j_{m-1}| \otimes U^{\sum_{\ell=0}^{m-1} j_\ell 2^\ell} \\
&= \sum_{j_0=0,1} |j_0\rangle\langle j_0| \otimes \cdots \otimes \sum_{j_{m-1}=0,1} |j_{m-1}\rangle\langle j_{m-1}| \otimes \prod_{\ell=0}^{m-1} U^{j_\ell 2^\ell}.
\end{aligned}$$

Definition 5.1 (Strong Kronecker product)**.** For $A \in \mathbb{C}^{mp\times mp}$, $B \in \mathbb{C}^{np\times np}$, we define the *strong Kronecker product* as the matrix denoted by $A \bowtie B \in \mathbb{C}^{mnp\times mnp}$ with

$$(A \bowtie B)_{abc,def} = \sum_{g=1}^{p} A_{ac;dg} B_{bc;gf}.$$

The strong Kronecker product is a mix of a Kronecker product with respect to the first index and a matrix-vector product in the second index.

Using this notation, we see that the expression of $\mathcal{U}$ can be written

$$\begin{aligned}\mathcal{U} &= (\sum_{j_0=0,1} |j_0\rangle\langle j_0| \otimes U^{j_0}) \bowtie \cdots \bowtie (\sum_{j_{m-1}=0,1} |j_{m-1}\rangle\langle j_{m-1}| \otimes U^{2^{m-1}j_{m-1}}) \\ &= (\text{C-}U) \bowtie (\text{C-}U^2) \bowtie \cdots \bowtie (\text{C-}U^{2^{m-1}}).\end{aligned}$$

The corresponding circuit is then given by

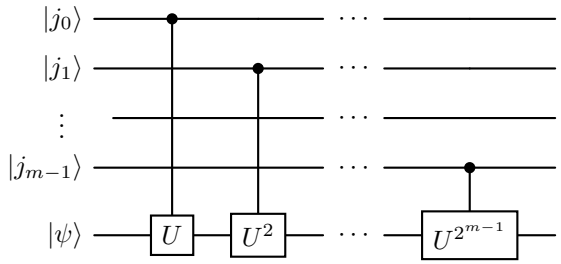


Figure 5: Circuit for $\mathcal{U}$

5.1 Circuit

Let $|\psi\rangle$ be an eigenvector of U for the eigenvalue $e^{2i\pi\theta}$. Then for any $|\phi\rangle = \sum_{j=0}^{2^m-1} \phi_j|j\rangle$ (with $\sum_{j=0}^{2^m-1} |\phi_j|^2 = 1$), we have

$$\begin{aligned}\mathcal{U}|\phi\rangle \otimes |\psi\rangle &= \sum_{j=0}^{2^m-1} \phi_j|j\rangle \otimes U^j|\psi\rangle \\ &= \sum_{j=0}^{2^m-1} e^{2i\pi\theta j}\phi_j|j\rangle \otimes |\psi\rangle.\end{aligned}$$

The coefficients of the quantum state in the first m qubits are given by the vector $(e^{2i\pi\theta j}\phi_j)$. It can be viewed as the discretisation of a function $x \mapsto e^{2^{m+1}i\pi\theta x}\phi(x)$ on the grid $x_j = \frac{j}{2^m}, j = 0, ..., 2^m - 1$ (with $\phi(x_j) = \phi_j$).

As we are looking for the value θ, it is natural to consider the Fourier transform as if $x \mapsto e^{2^{m+1}i\pi\theta x}$ is 1-periodic, then its inverse discrete Fourier transform is the canonical vector e_k where k is the periodicity of the function $x \mapsto e^{2^{m+1}i\pi\theta x}$.

The QPE circuit is given below

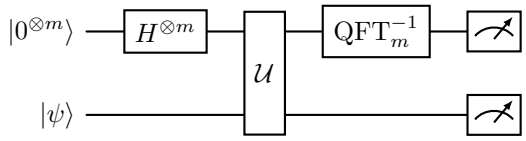


Figure 6: QPE circuit

Remark 5.1. For $m = 1$, the QPE circuit reduces to the Hadamard test $(H \otimes I_{2^n})(\text{C-}U)(H \otimes I_{2^n})$, which has the following circuit:

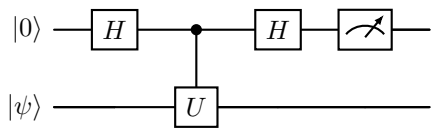


Figure 7: Hadamard test

As $H^{\otimes m}|0\rangle^{\otimes m} = \text{QFT}_m|0\rangle^{\otimes m}$, the circuit can also be written with a QFT gate instead of the tensor product of Hadamard gates.

Theorem 5.1 (Quantum phase estimation)**.** *Assume that ψ is an eigenvector of the unitary U with eigenvalue $e^{2i\pi\theta}$ such that $\theta = \frac{k}{2^m}$ for some $0 \leq k \leq 2^m - 1$.*

Then the resulting state of the QPE algorithm in Figure 6 is $|k\rangle \otimes |\psi\rangle$, i.e. *the result of the QPE algorithm is deterministic and only requires one measurement.*

Proof. By assumption on the eigenvalue, the coefficients of the quantum state in the first m qubits are the inverse discrete Fourier transform of the periodic function $x \mapsto e^{2^{m+1}i\pi\theta x}$. Hence it is localised in one coefficient.

We can also compute exactly the output of the circuit and find the same conclusion. By definition of the QFT, we have $\text{QFT}_m|0\rangle^{\otimes m} = \frac{1}{2^{m/2}} \sum_{j=0}^{2^m-1} |j\rangle$. Thus we have

$$\begin{aligned}\mathcal{U}(\text{QFT}_m|0\rangle^{\otimes m} \otimes |\psi\rangle) &= \frac{1}{2^{m/2}} \sum_{j=0}^{2^m-1} |j\rangle \otimes U^j|\psi\rangle \\ &= \frac{1}{2^{m/2}} \sum_{j=0}^{2^m-1} e^{2i\frac{kj}{2^m}}|j\rangle \otimes |\psi\rangle.\end{aligned}$$

Thus applying the last QFT_m^{-1} gate, we have

$$(\text{QFT}_m^{-1} \otimes I_{2^n})(\frac{1}{2^{m/2}} \sum_{j=0}^{2^m-1} e^{2i\frac{kj}{2^m}}|j\rangle \otimes |\psi\rangle) = \frac{1}{2^m} \sum_{\ell=0}^{2^m-1} \sum_{j=0}^{2^m-1} e^{2i\frac{(k-\ell)j}{2^m}}|\ell\rangle^{\otimes m} \otimes |\psi\rangle.$$

Now $\sum_{j=0}^{2^m-1} e^{2i\frac{(k-\ell)j}{2^m}} = 0$ if $\ell \neq k$, thus the resulting state of the QPE is $|k\rangle \otimes |\psi\rangle$. □

5.2 Limitations and sources of error

There are two main limitation in the algorithm:

- an exact eigenvector ψ of the unitary matrix U needs to be implemented in the quantum computer: in practice, it is unreasonable to assume that such an eigenvector can be achieved exactly as a starting point for the QPE. One can only hope for a starting quantum state $|\widetilde{\psi}\rangle$ which has a significant overlap with the target quantum state $|\psi\rangle$. In practice, it can be a challenge to even prepare such a state.
- another limitation in the algorithm is the m-digit representability of the angle θ associated to the eigenvalue $e^{2i\pi\theta}$.

Let us investigate this point. As being stated above, the coefficients are the inverse discrete Fourier transform of the function $x \mapsto e^{2^{m+1} i\pi x}$ on the grid points $(\frac{j}{2^m})_{0\leq j\leq 2^m-1}$. The k-th coefficient is:

$$\begin{aligned}\frac{1}{2^{m/2}}\sum_{\ell=0}^{2^m-1} e^{2i\pi\theta\ell}e^{-2i\pi\frac{k\ell}{2^m}} &= \frac{1}{2^{m/2}}\sum_{\ell=0}^{2^m-1} e^{2i\pi\frac{(2^m\theta-k)\ell}{2^m}} \\ &= \frac{1}{2^{m/2}}\frac{1-e^{2^{m+1}i\pi\theta}}{1-e^{2i\pi(\theta-\frac{k}{2^m})}} \\ &= \frac{1}{2^{m/2}}\frac{\sin(2^m\pi\theta)}{\sin((\theta-\frac{k}{2^m})\pi)}.\end{aligned}$$

The squared coefficients define a discrete probability law with parameter θ. Measurements $(|k\rangle)_{k\in K}$ of the first m-qubits are independent samples of a random variable following the probability law $(\frac{1}{2^m}\frac{\sin(2^m\pi\theta)^2}{\sin((\theta-\frac{k}{2^m})\pi)})^2$. Standard parametric estimations can be used in order to evaluate θ, and hence the eigenvalue. In Figure 8, we plot the result of the *maximum likelihood estimation. The estimated parameter $\widehat{\theta}$ is defined as the maximum of the loss function

$$\mathcal{L}(\theta, K) = \sum_{k\in K} \log\Big|\frac{\sin(2^m\pi\theta)}{\sin((\theta-\frac{k}{2^m})\pi)}\Big|.$$

This step can be done on a classical computer and does not require to run additional calculations on a quantum computer.
The result as well as the estimated variance are given in the following plot for $m = 4$:

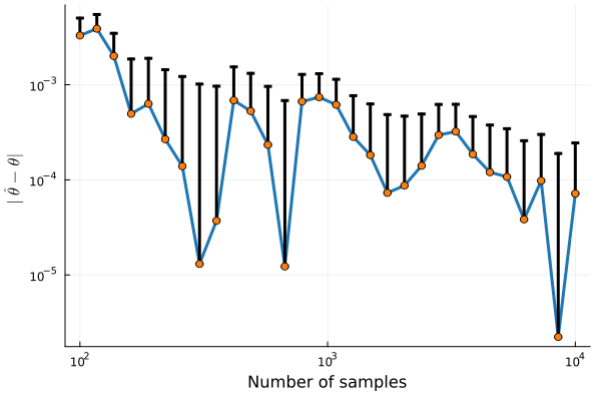


Figure 8: Maximum likelihood estimation

Although the number of qubits to estimate θ is small, the parametric estimation of θ is accurate up to the 4-th digit.