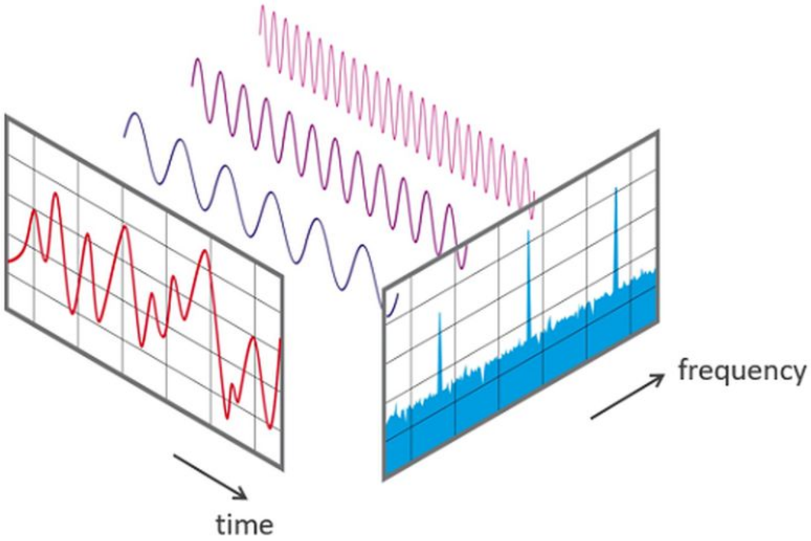
This is CS4048!

GCR:wzj3vua

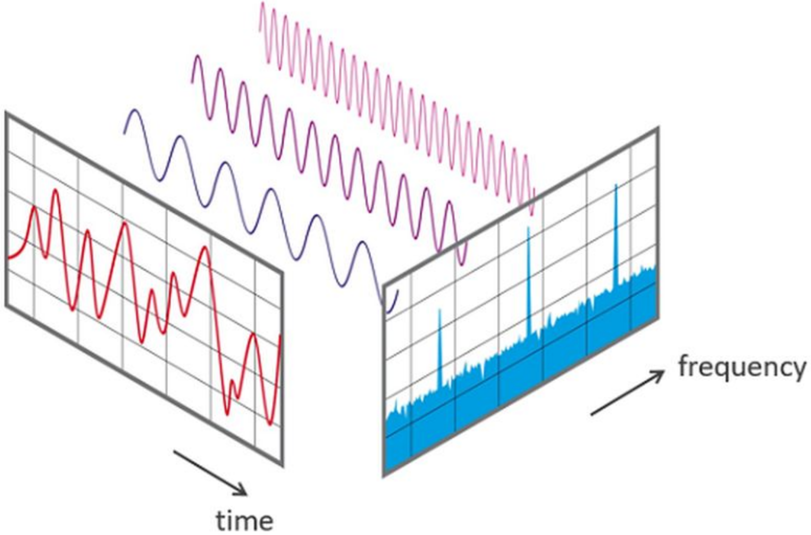
Quantum Fourier transform



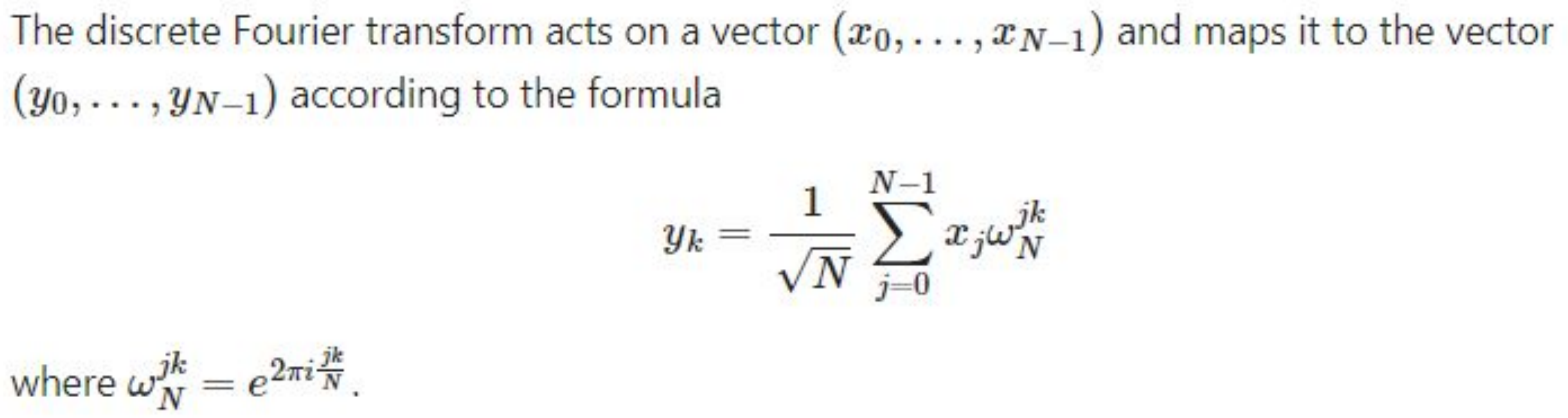
Classical Fourier Transform

To transform a signal in the time domain f(t) to the frequency domain ℱ(ω), or vice versa

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$$



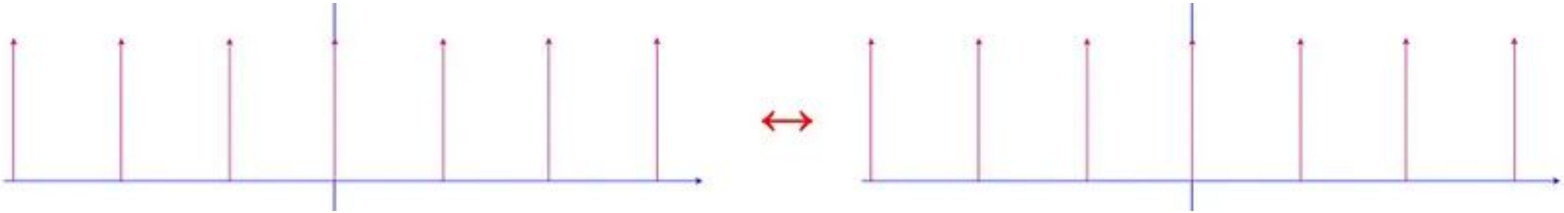
Discrete Fourier Transform



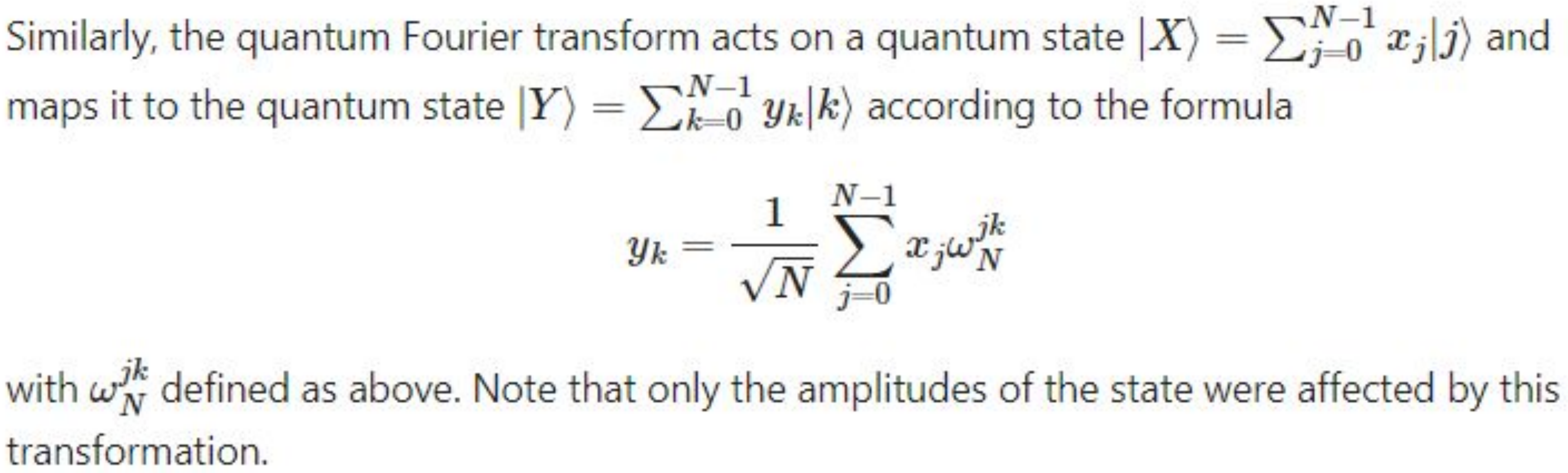
The discrete Fourier transform acts on a vector $(x_0, \ldots, x_{N-1})$ and maps it to the vector $(y_0, \ldots, y_{N-1})$ according to the formula

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{jk}$$

where $\omega_N^{jk} = e^{2\pi i \frac{jk}{N}}$.



Quantum Fourier Transform

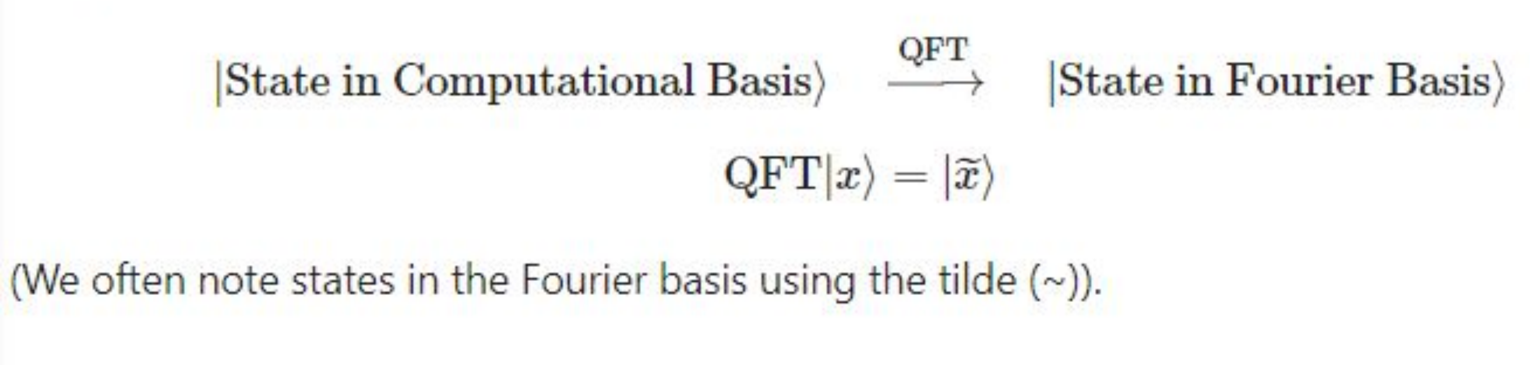


Similarly, the quantum Fourier transform acts on a quantum state $|X\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$ and maps it to the quantum state $|Y\rangle = \sum_{k=0}^{N-1} y_k |k\rangle$ according to the formula

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{jk}$$

with ω_N^{jk} defined as above. Note that only the amplitudes of the state were affected by this transformation.

Intuition



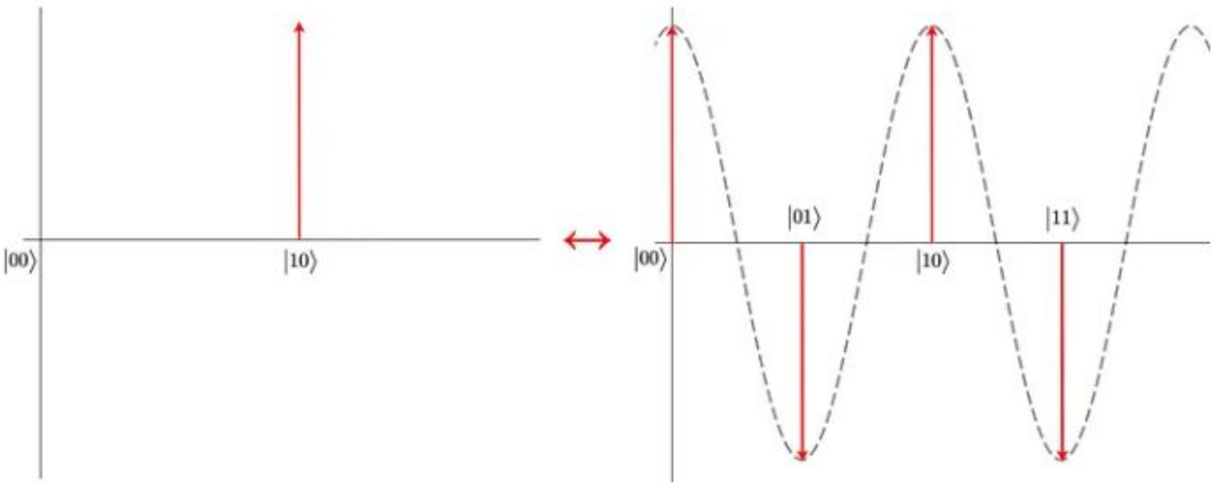
$$|\text{State in Computational Basis}\rangle \quad \xrightarrow{\text{QFT}} \quad |\text{State in Fourier Basis}\rangle$$

$$\text{QFT}|x\rangle = |\widetilde{x}\rangle$$

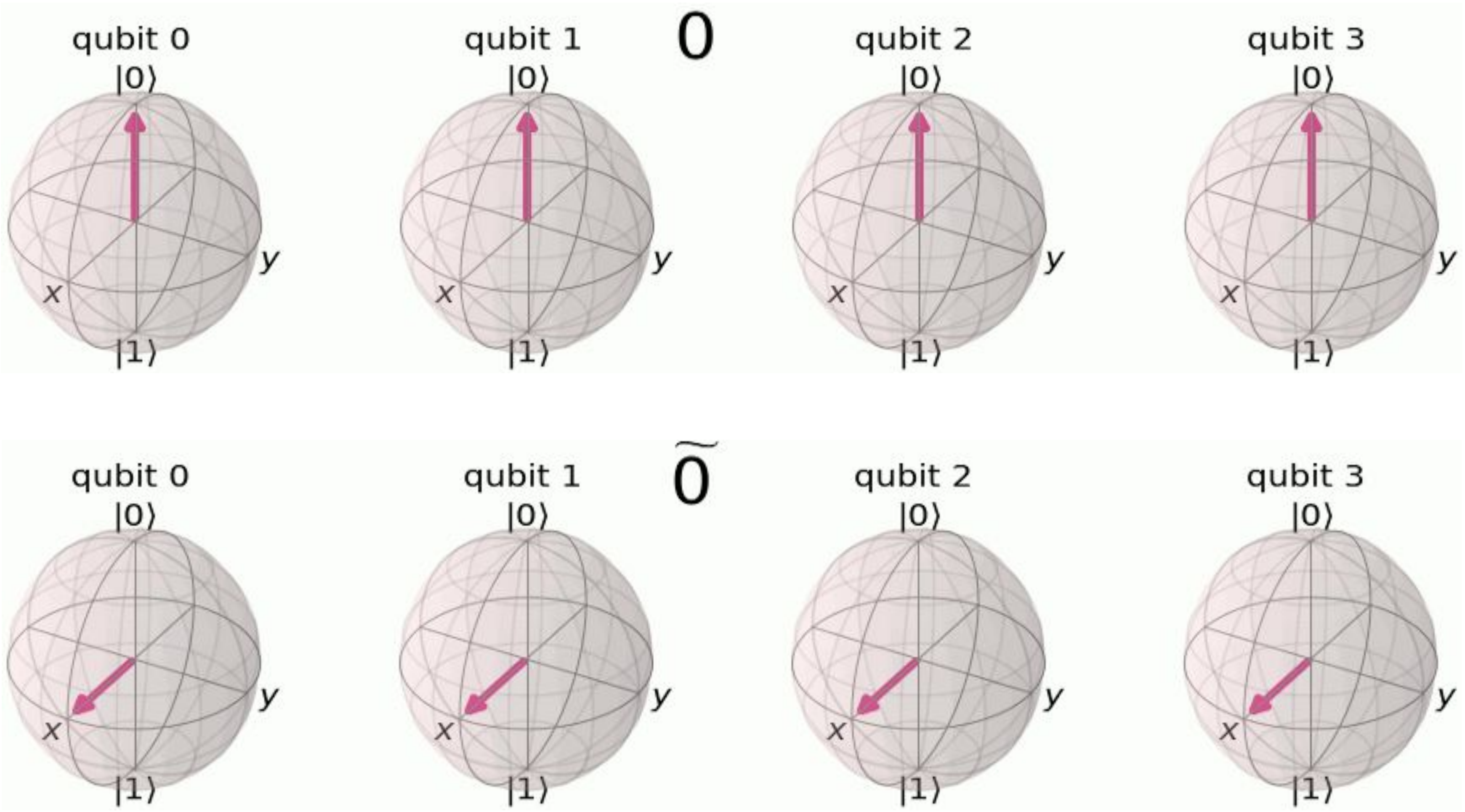
(We often note states in the Fourier basis using the tilde (~)).

Quantum Fourier Transform

i.e., |10⟩ will transform into the superposition below:



> *“Quantum Fourier Transform is an important part of many quantum algorithms, most notably Shor’s factoring algorithm and Quantum Phase Estimation”*
>
> IBM Quantum


0
qubit 0
qubit 1
qubit 2
qubit 3
|0⟩
|1⟩
x
y
$\tilde{0}$
qubit 0
qubit 1
qubit 2
qubit 3
|0⟩
|1⟩
x
y

QFT Formula

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

QFT Matrix Representation

$$QFT = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2N-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3N-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \omega^{3N-3} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix}$$

Quantum Fourier Transform – Home Task

Apply QFT to the basis state |10⟩ and find the new quantum state.

$$QFT = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2N-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3N-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \omega^{3N-3} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix}$$

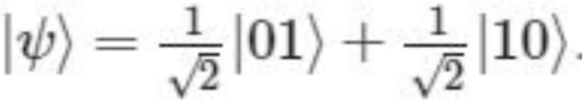
$$y = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$$

Quantum Fourier Transform

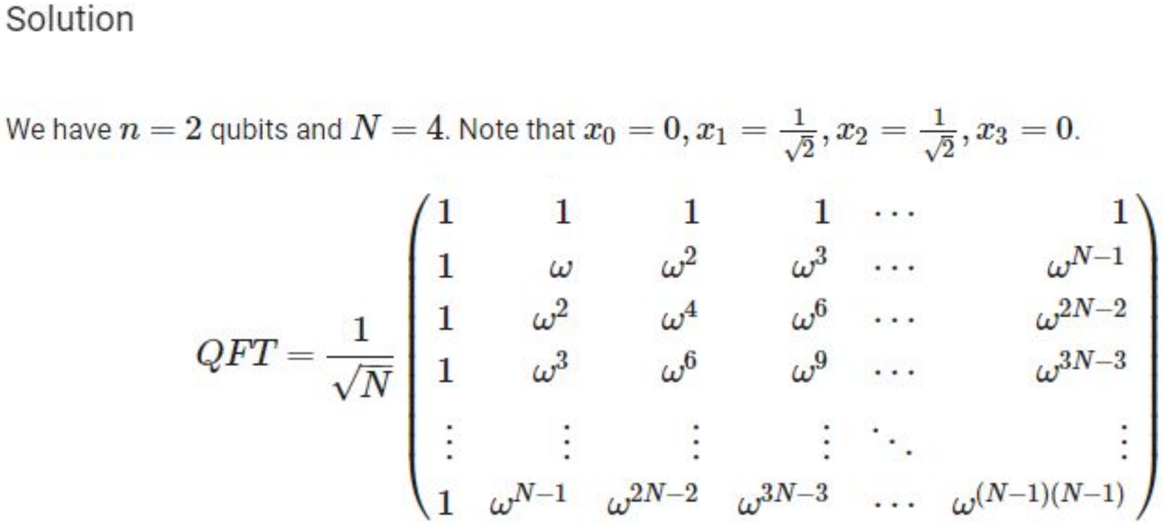
Apply QFT to the quantum state $|\psi\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle.$

Quantum Fourier Transform

Apply QFT to the quantum state



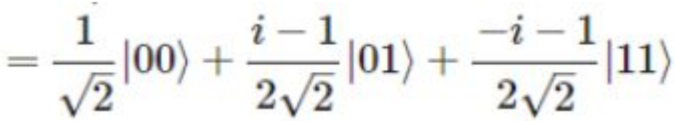
$$|\psi\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle.$$



Solution

We have $n = 2$ qubits and $N = 4$. Note that $x_0 = 0, x_1 = \frac{1}{\sqrt{2}}, x_2 = \frac{1}{\sqrt{2}}, x_3 = 0$.

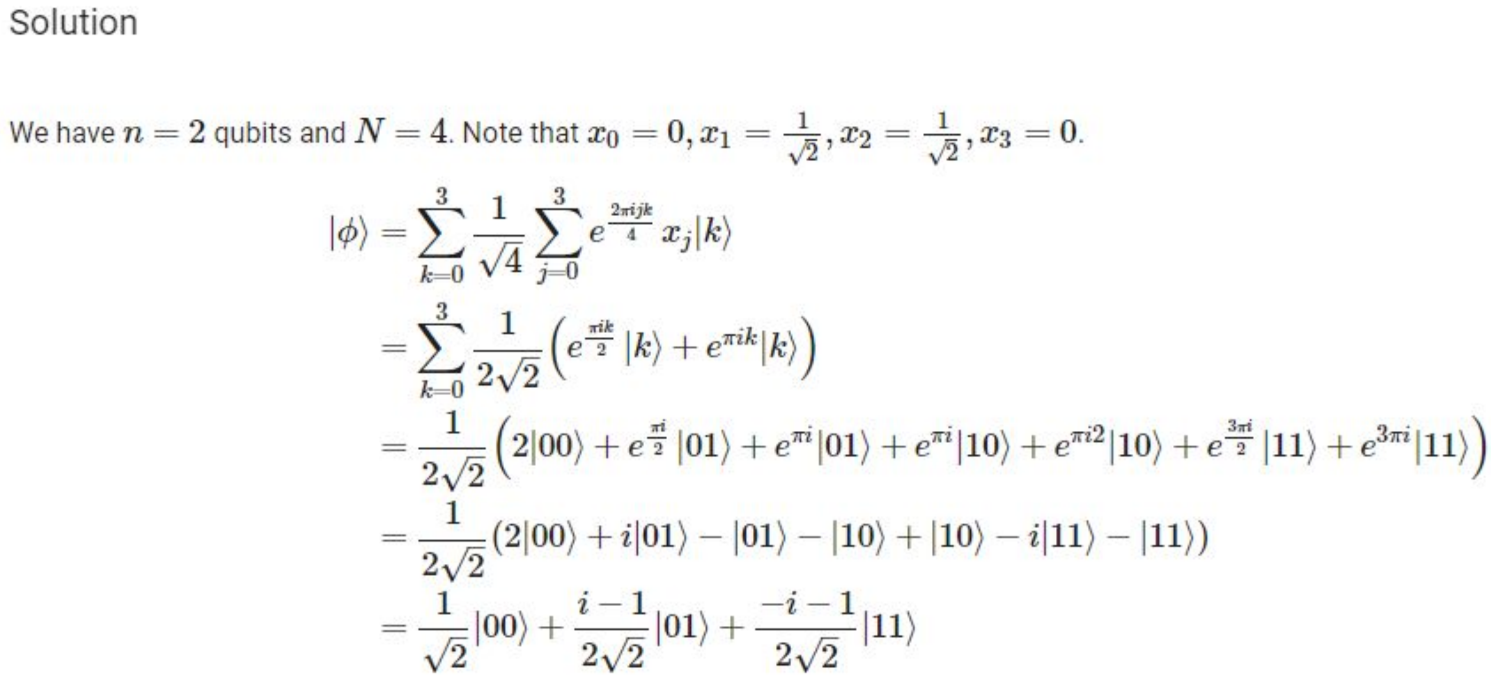
$$QFT = \frac{1}{\sqrt{N}}\begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2N-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3N-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \omega^{3N-3} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix}$$



$$= \frac{1}{\sqrt{2}}|00\rangle + \frac{i-1}{2\sqrt{2}}|01\rangle + \frac{-i-1}{2\sqrt{2}}|11\rangle$$

Quantum Fourier Transform - Alternate Method

Apply QFT to the quantum state $|\psi\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle.$



Solution

We have $n = 2$ qubits and $N = 4$. Note that $x_0 = 0, x_1 = \frac{1}{\sqrt{2}}, x_2 = \frac{1}{\sqrt{2}}, x_3 = 0$.

$$
\begin{aligned}
|\phi\rangle &= \sum_{k=0}^{3} \frac{1}{\sqrt{4}} \sum_{j=0}^{3} e^{\frac{2\pi ijk}{4}} x_j |k\rangle \\
&= \sum_{k=0}^{3} \frac{1}{2\sqrt{2}} \left(e^{\frac{\pi ik}{2}} |k\rangle + e^{\pi ik} |k\rangle \right) \\
&= \frac{1}{2\sqrt{2}} \left(2|00\rangle + e^{\frac{\pi i}{2}} |01\rangle + e^{\pi i}|01\rangle + e^{\pi i}|10\rangle + e^{\pi i2}|10\rangle + e^{\frac{3\pi i}{2}} |11\rangle + e^{3\pi i}|11\rangle \right) \\
&= \frac{1}{2\sqrt{2}} (2|00\rangle + i|01\rangle - |01\rangle - |10\rangle + |10\rangle - i|11\rangle - |11\rangle) \\
&= \frac{1}{\sqrt{2}}|00\rangle + \frac{i-1}{2\sqrt{2}}|01\rangle + \frac{-i-1}{2\sqrt{2}}|11\rangle
\end{aligned}
$$

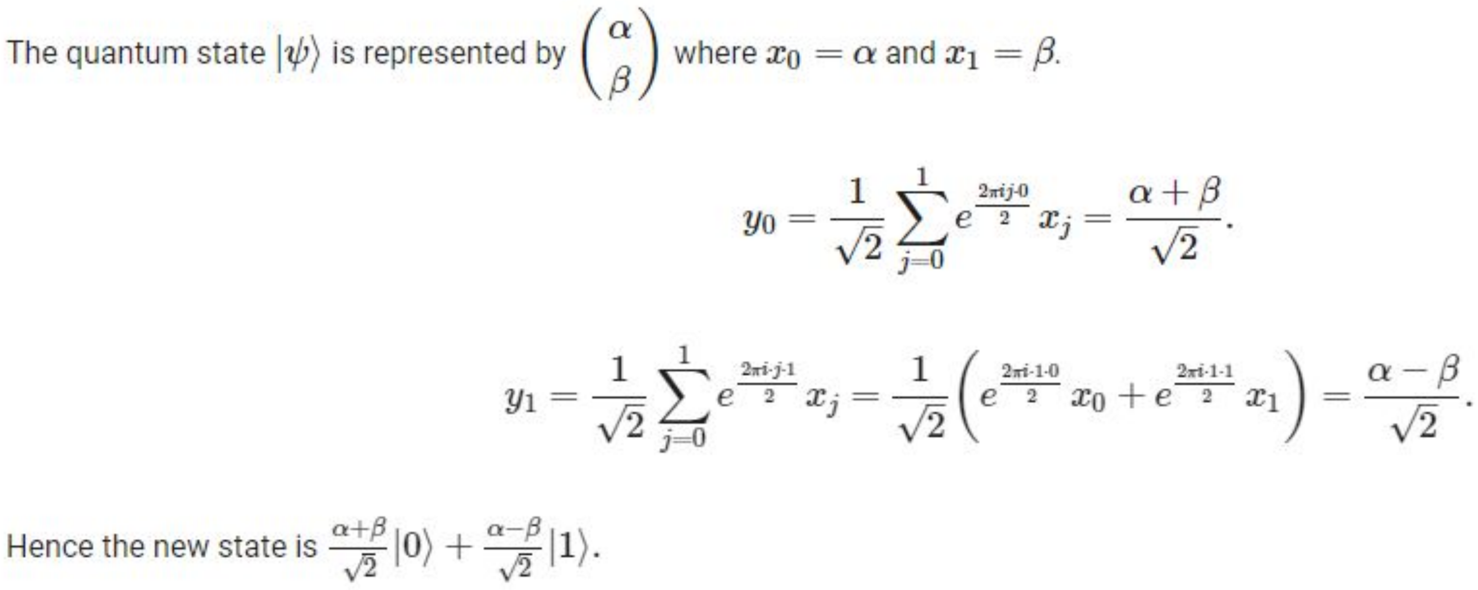
Quantum Fourier Transform

Apply QFT to the quantum state $|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$ and find the new quantum state.

Matrix Expansion:

Quantum Fourier Transform

Apply QFT to the quantum state |ψ⟩=α|0⟩+β|1⟩ and find the new quantum state.



The quantum state $|\psi\rangle$ is represented by $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ where $x_0 = \alpha$ and $x_1 = \beta$.

$$y_0 = \frac{1}{\sqrt{2}} \sum_{j=0}^{1} e^{\frac{2\pi i j \cdot 0}{2}} x_j = \frac{\alpha + \beta}{\sqrt{2}}.$$

$$y_1 = \frac{1}{\sqrt{2}} \sum_{j=0}^{1} e^{\frac{2\pi i \cdot j \cdot 1}{2}} x_j = \frac{1}{\sqrt{2}} \left(e^{\frac{2\pi i \cdot 1 \cdot 0}{2}} x_0 + e^{\frac{2\pi i \cdot 1 \cdot 1}{2}} x_1 \right) = \frac{\alpha - \beta}{\sqrt{2}}.$$

Hence the new state is $\frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle$.

Quantum Fourier Transform

Apply QFT to the quantum state $|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$ and find the new quantum state.

Conclude that applying 1 qubit QFT is equivalent to applying Hadamard gate.

Quantum Fourier transform

The Quantum Fourier Transform can also be decomposed into several basic quantum gates, acting on one or two qubits.

This permits us to easily represent graphically the operator as a quantum circuit.

Indeed, the QFT can be defined as a combination and succession of three different gates: the Hadamard gate, the SWAP gate, and the controlled-Rk .

Controlled R Quantum gate

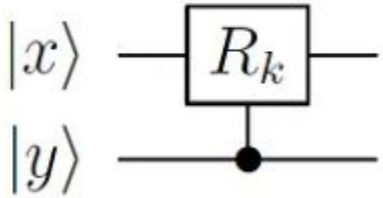
A controlled-R quantum gate applies a relative phase change to |1>. The matrix form of this operator is:

$$\hat{R}_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

Circuit representation Of QFT

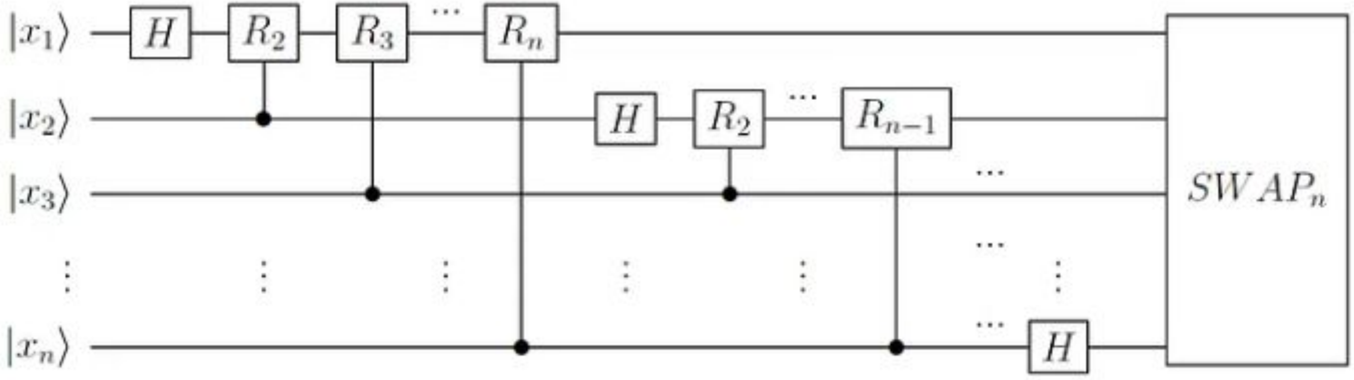
A controlled-R quantum gate applies a relative phase change to |1>. The matrix form of this operator is:

$$cR_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\frac{2i\pi}{2^k}} \end{pmatrix}$$

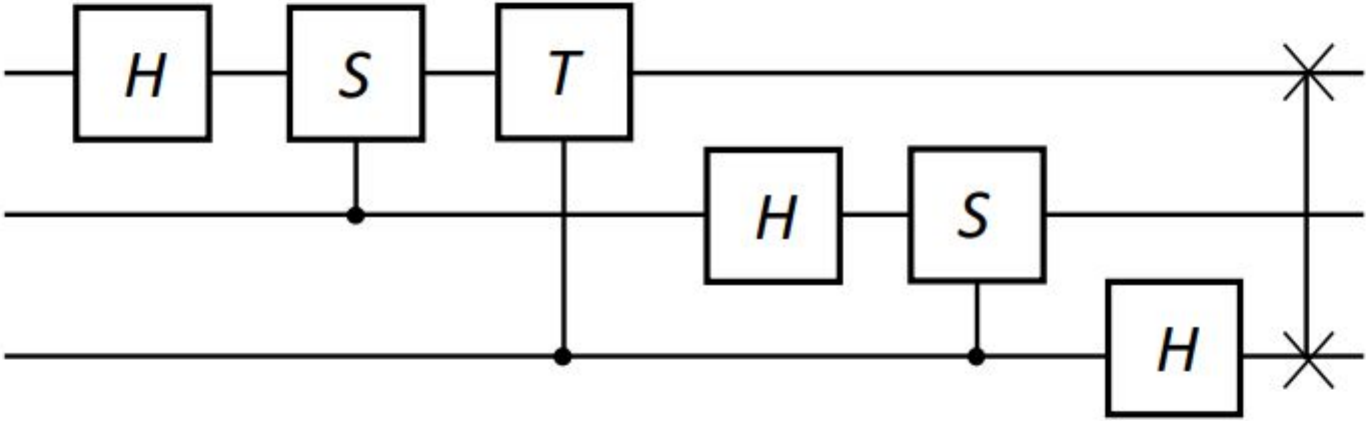


Circuit representation Of QFT

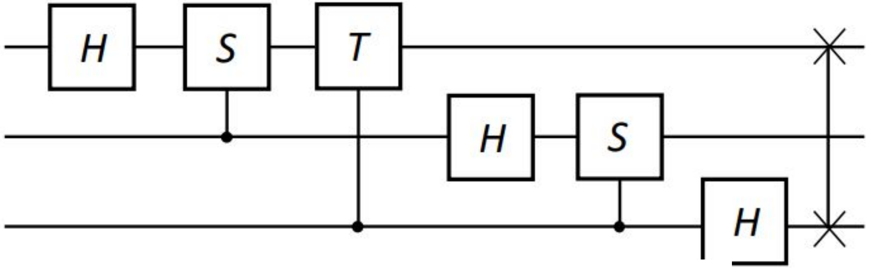
The complete circuit of the QFT is represented below, and the last gate is SWAPn, the n-qubit gate consisting in swapping the first qubit with the last, the second with the (n-1)-th qubit, and so on.



three-qubit quantum Fourier transform



THREE-QUBIT QFT



Prior to the SWAP gate we have:

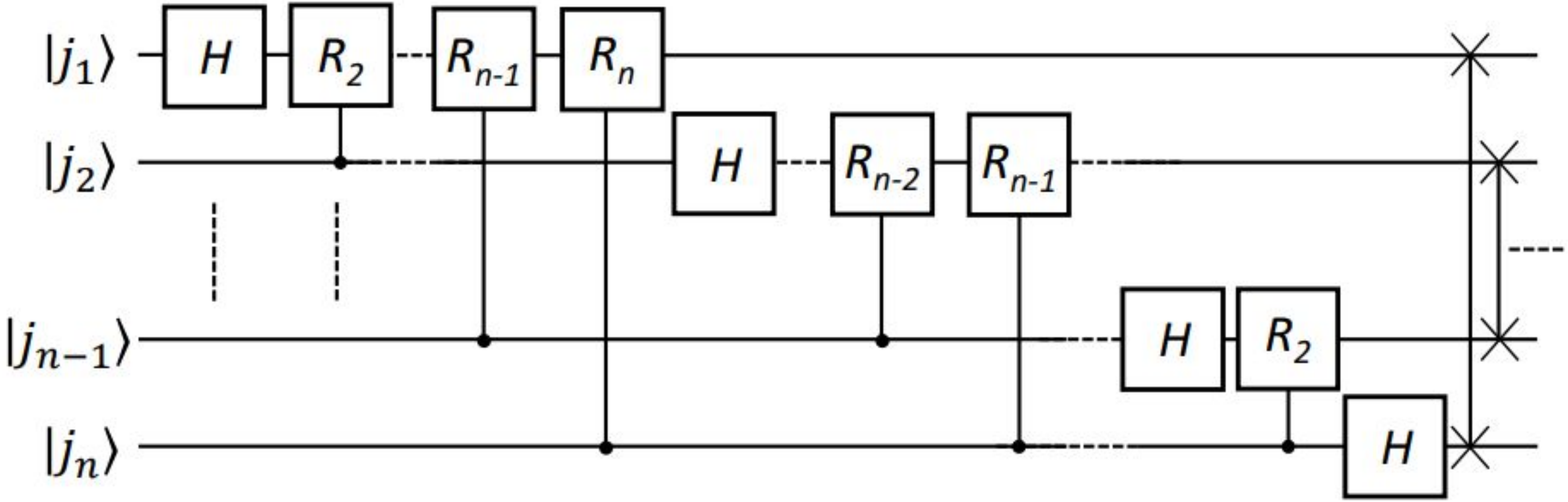
$$|j_1\rangle \rightarrow \frac{|0\rangle + e^{\pi i j_1}|1\rangle}{\sqrt{2}} \rightarrow \frac{|0\rangle + e^{\pi i j_1} e^{\pi i j_2/2}|1\rangle}{\sqrt{2}} \rightarrow \frac{|0\rangle + e^{\pi i j_1} e^{\pi i j_2/2} e^{\pi i j_3/4}|1\rangle}{\sqrt{2}}$$

$$|j_2\rangle \rightarrow \frac{|0\rangle + e^{\pi i j_2}|1\rangle}{\sqrt{2}} \rightarrow \frac{|0\rangle + e^{\pi i j_1} e^{\pi i j_3/2}|1\rangle}{\sqrt{2}}$$

$$|j_3\rangle \rightarrow \frac{|0\rangle + e^{\pi i j_3}|1\rangle}{\sqrt{2}}$$

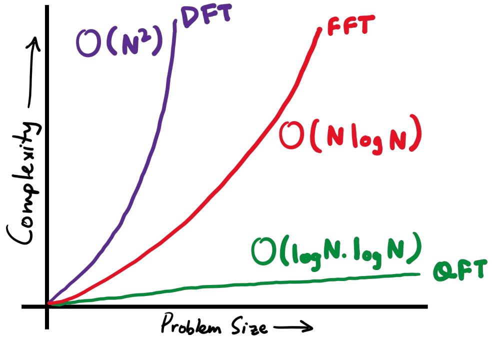
We can thus express the state after the SWAP gate:

$$\frac{1}{2\sqrt{2}}\left(|0\rangle + e^{\pi i j_3}|1\rangle\right)\left(|0\rangle + e^{\pi i (j_2 + (j_3/2))}|1\rangle\right)\left(|0\rangle + e^{\pi i (j_1 + (j_2/2) + (j_3/4))}|1\rangle\right)$$



Where $R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$ is a single qubit unitary *rotation* gate.

Circuit representation Of QFT



In total, the maximum number of quantum gates applied is equal to

$$\frac{n(n+1)}{2} + \frac{n}{2}$$

This implies a complexity of O(n²) in the number of operations, which is a complexity of O(log(N)²), with N the number of basis states. This quadratic complexity of the QFT is one of the key reasons why Shor’s algorithm is polynomial.

The inverse quantum Fourier transform

To invert the QFT, we must run the circuit in reverse, with the inverse of each gate in place to achieve the transform:

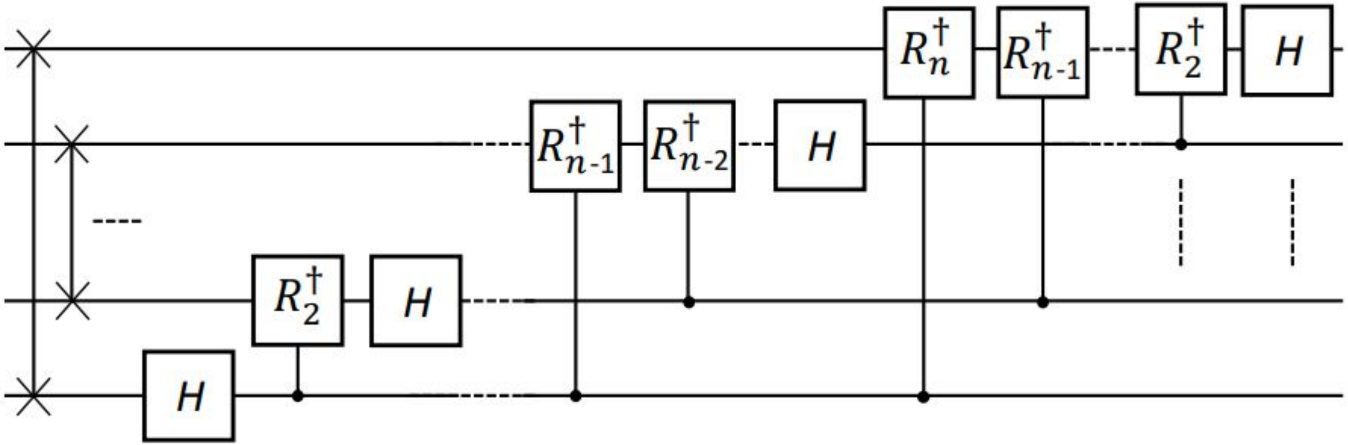
$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \to |j\rangle$$

We have already seen that the Hadamard gate is self-inverse, and the same is clearly true for the SWAP gate; the inverse of the rotations gate R_k is given by:

$$R_k^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i/2^k} \end{bmatrix}$$

The inverse quantum Fourier transform circuit

Thus we can express the inverse QFT circuit:



If you wanna dive deep

The quantum Fourier transform

We can see that the general quantum Fourier transform is thus (where $N = 2^n$):

$$\frac{1}{\sqrt{N}}\left(|0\rangle + e^{\pi i j_n}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{\pi i (j_2 + (j_3/2) + \cdots + (j_n/(2^{n-1})))}|1\rangle\right) \otimes \left(|0\rangle + e^{\pi i (j_1 + (j_2/2) + \cdots + (j_n/(2^n)))}|1\rangle\right)$$

To rearrange further, we will use binary decimals (also termed binary fractions, that is we can express: $\frac{a_1}{2} + \frac{a_2}{4} + \cdots + \frac{a_n}{2^n}$ as the binary decimal $0.a_1 a_2 \cdots a_n$. So we can thus express the QFT:

$$\frac{1}{\sqrt{N}}\left(|0\rangle + e^{2\pi i (0.j_n)}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i (0.j_2 j_3 \cdots j_n)}|1\rangle\right)\left(|0\rangle + e^{2\pi i (0.j_1 j_2 \cdots j_n)}|1\rangle\right)$$

Furthermore, as $e^{2m\pi i} = 1$ for any integer m, this equals:

$$\frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left(|0\rangle + e^{2\pi i j 2^{-l}}|1\rangle\right)$$

The quantum Fourier transform (continued)

We now rearrange the quantum Fourier transform into standard form:

$$\begin{aligned}
\frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left(|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \right) =& \frac{1}{\sqrt{N}} \bigotimes_{l=1}^{n} \left(\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right) \\
=& \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \\
=& \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} |k_1 \cdots k_n\rangle \\
=& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle
\end{aligned}$$

Resources for Quantum Fourier Transform

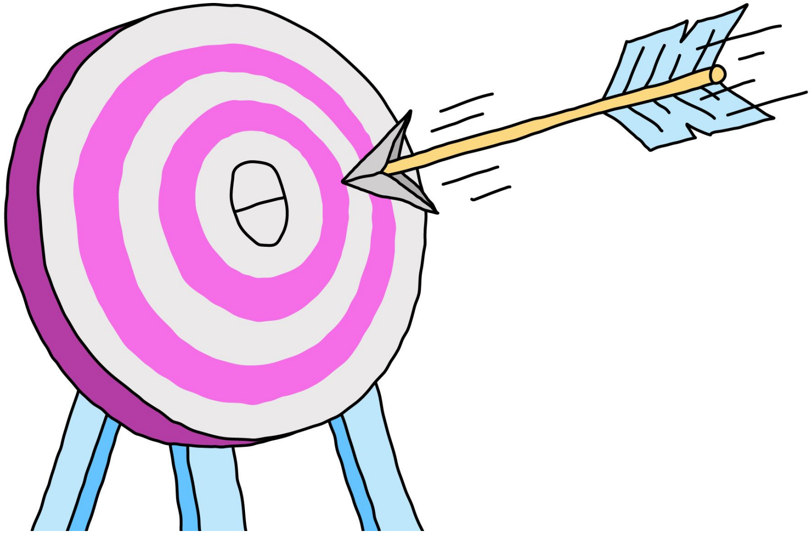
https://courses.edx.org/c4x/BerkeleyX/CS191x/asset/chap5.pdf

https://jonathan-hui.medium.com/qc-quantum-fourier-transform-45436f90a43

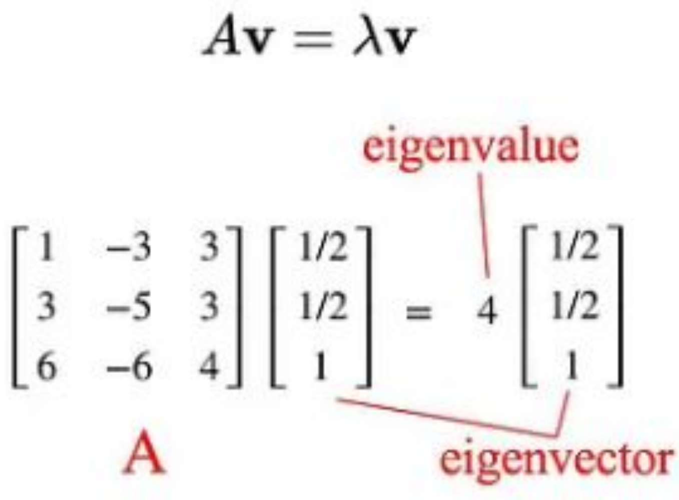
https://medium.com/colibritd-quantum/getting-to-know-quantum-fourier-transform-ae60b23e58f4

https://www.cl.cam.ac.uk/teaching/1920/QuantComp/Quantum_Computing_Lecture_9.pdf

Phase Estimation



EigenValues & EigenVector



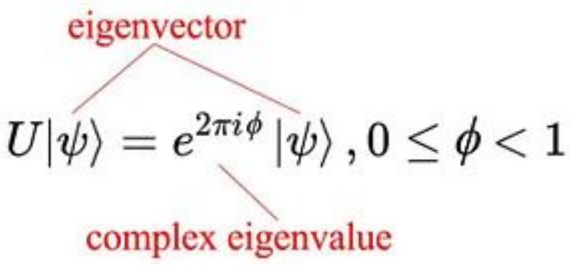
$$A\mathbf{v} = \lambda\mathbf{v}$$

$$\underbrace{\begin{bmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 1/2 \\ 1/2 \\ 1 \end{bmatrix}}_{\text{eigenvector}} = \overset{\text{eigenvalue}}{4} \underbrace{\begin{bmatrix} 1/2 \\ 1/2 \\ 1 \end{bmatrix}}_{\text{eigenvector}}$$

Quantum Phase Estimation (QPE)

Here ϕ is the phase of the eigenvalue (remember, unitaries have eigenvalues with an absolute value of 1). The goal is to estimate ϕ,hence the name phase estimation.

Our challenge is to design a quantum algorithm to solve this problem. How would that work?



eigenvector

$$U|\psi\rangle = e^{2\pi i\phi}\,|\psi\rangle\,, 0 \leq \phi < 1$$

complex eigenvalue

Relative Vs Global Phase

relative phase

$$|\psi\rangle = e^{i\theta}\left(\alpha|0\rangle + \beta e^{i\phi}|1\rangle\right)$$

global phase

Global phases do not affect measurement outcome probabilities at all and, as such, are unobservable; whereas relative phases can affect measurement outcomes if measured in a different basis.

But what if we had a situation where we did want to figure out the global phase. Could we do so?

Finding Global Phase

This problem corresponds to figuring out the eigenvalue of a unitary operator corresponding to a given eigenvector.

The action of a unitary operator U on its eigenvector |ψ⟩ is

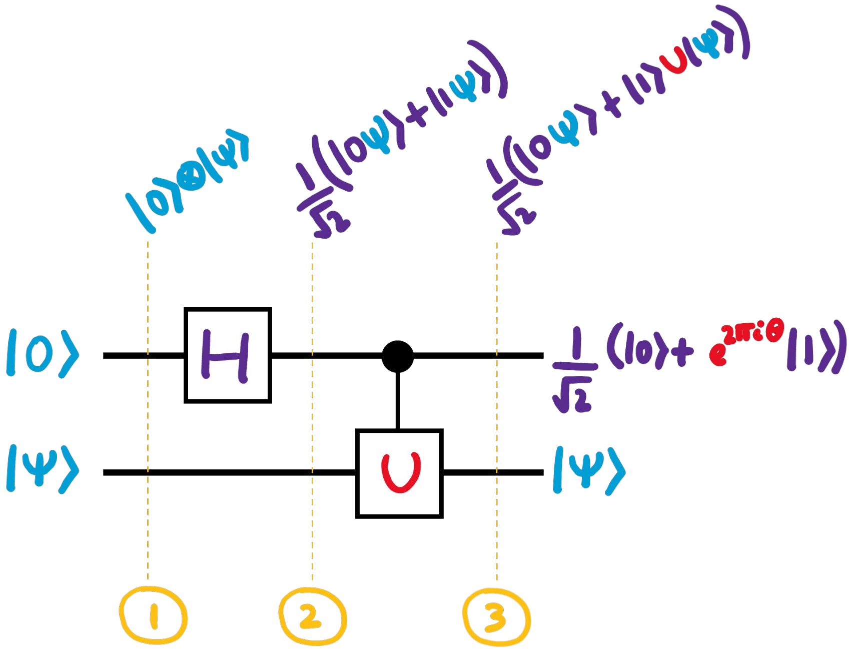
$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle.$$

$e^{2\pi i\theta}$ is an eigenvalue with a unit norm, which appears as a global phase in the output.

Phase Kickback

A controlled unitary operation, applies the unitary to the target wires. In the case where the target wires are in prepared in an eigenstate of the unitary, the eigenvalue corresponding to this eigenstate is kicked back to the control wires.

Then, the control wires pick up a phase of $e^{2\pi i\theta}$. This is called phase kickback.

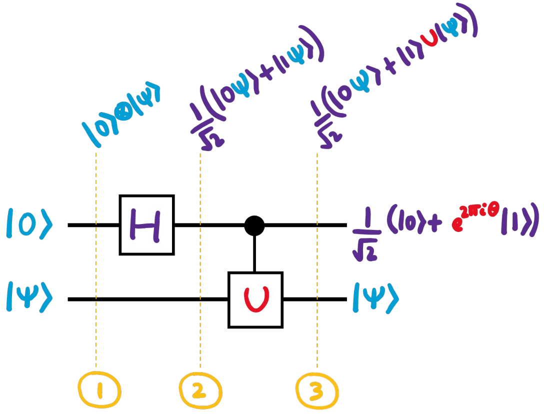

$|0\rangle\otimes|\psi\rangle$
$\frac{1}{\sqrt{2}}(|0\psi\rangle+|1\psi\rangle)$
$\frac{1}{\sqrt{2}}(|0\psi\rangle+|1\rangle U|\psi\rangle)$
$|0\rangle$
$|\psi\rangle$
H
U
$\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i\theta}|1\rangle)$
$|\psi\rangle$
1
2
3

Phase KickBack

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle.$$

We have transformed the eigenvalue which was initially a global phase into a relative phase on the control wire!

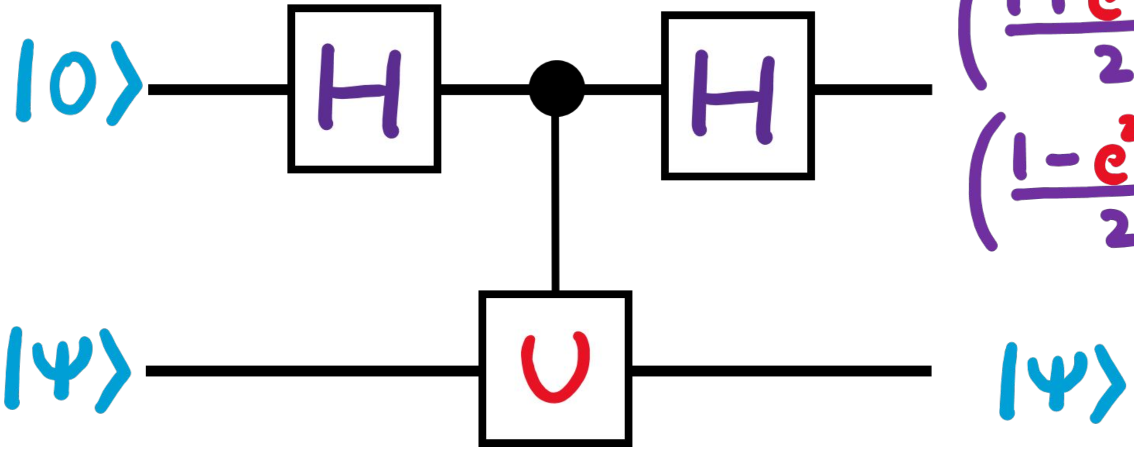
This still doesn't solve our problem of finding the eigenvalue of U, but we are one step closer. If we change the basis by applying the Hadamard gate after the controlled unitary, we can obtain some more information.


$|0\rangle\otimes|\psi\rangle$
$\frac{1}{\sqrt{2}}(|0\psi\rangle+|1\psi\rangle)$
$\frac{1}{\sqrt{2}}(|0\psi\rangle+|1\rangle U|\psi\rangle)$
$|0\rangle$
$|\psi\rangle$
H
U
$\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i\theta}|1\rangle)$
$|\psi\rangle$
1
2
3

Phase kickback or eigenvalue kickback is neat trick used by many quantum algorithms.

Phase Kickback

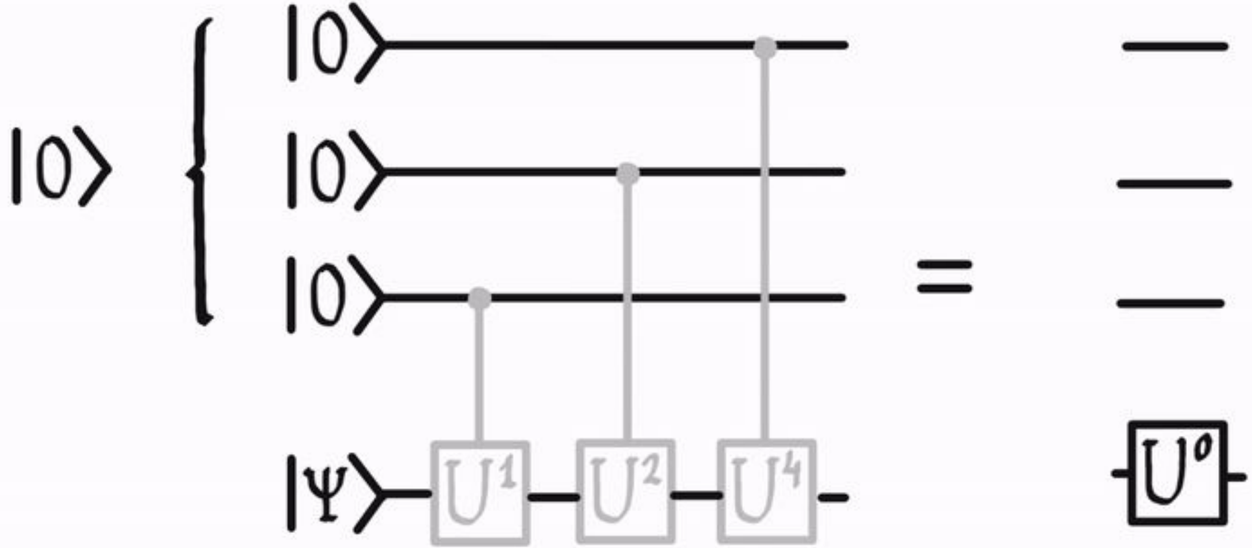
$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$$



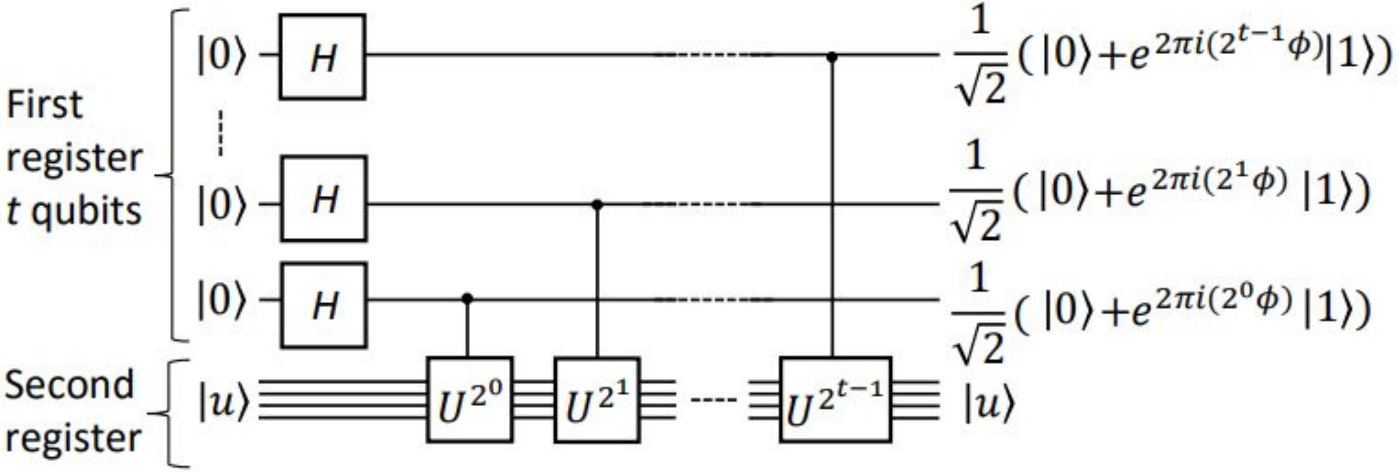
$$\left(\frac{1+e^{2\pi i\theta}}{2}\right)|0\rangle + \left(\frac{1-e^{2\pi i\theta}}{2}\right)|1\rangle$$

$$P(0) = \left|\frac{1+e^{2\pi i\theta}}{2}\right|^2$$

$$P(1) = \left|\frac{1-e^{2\pi i\theta}}{2}\right|^2$$

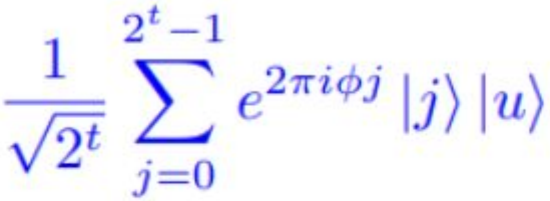


|0⟩
|0⟩
|0⟩
|0⟩
|Ψ⟩
U¹
U²
U⁴
=
U⁰


First register t qubits
Second register
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|u\rangle$
H
H
H
U^{2^0}
U^{2^1}
$U^{2^{t-1}}$
$\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i(2^{t-1}\phi)}|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i(2^1\phi)}|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle+e^{2\pi i(2^0\phi)}|1\rangle)$
$|u\rangle$

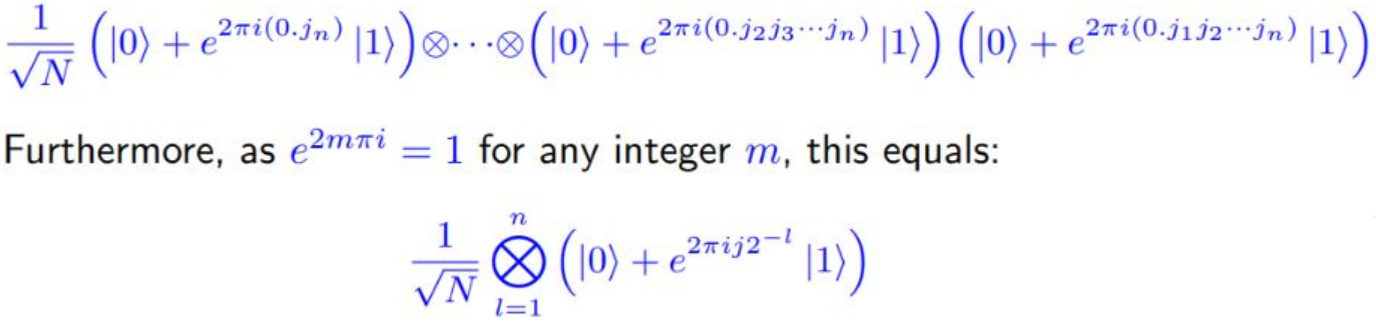
Using the previous analysis, the final state can be expressed:

$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1} e^{2\pi i\phi j}\,|j\rangle\,|u\rangle$$



$$\frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1} e^{2\pi i\phi j}\,|j\rangle\,|u\rangle$$

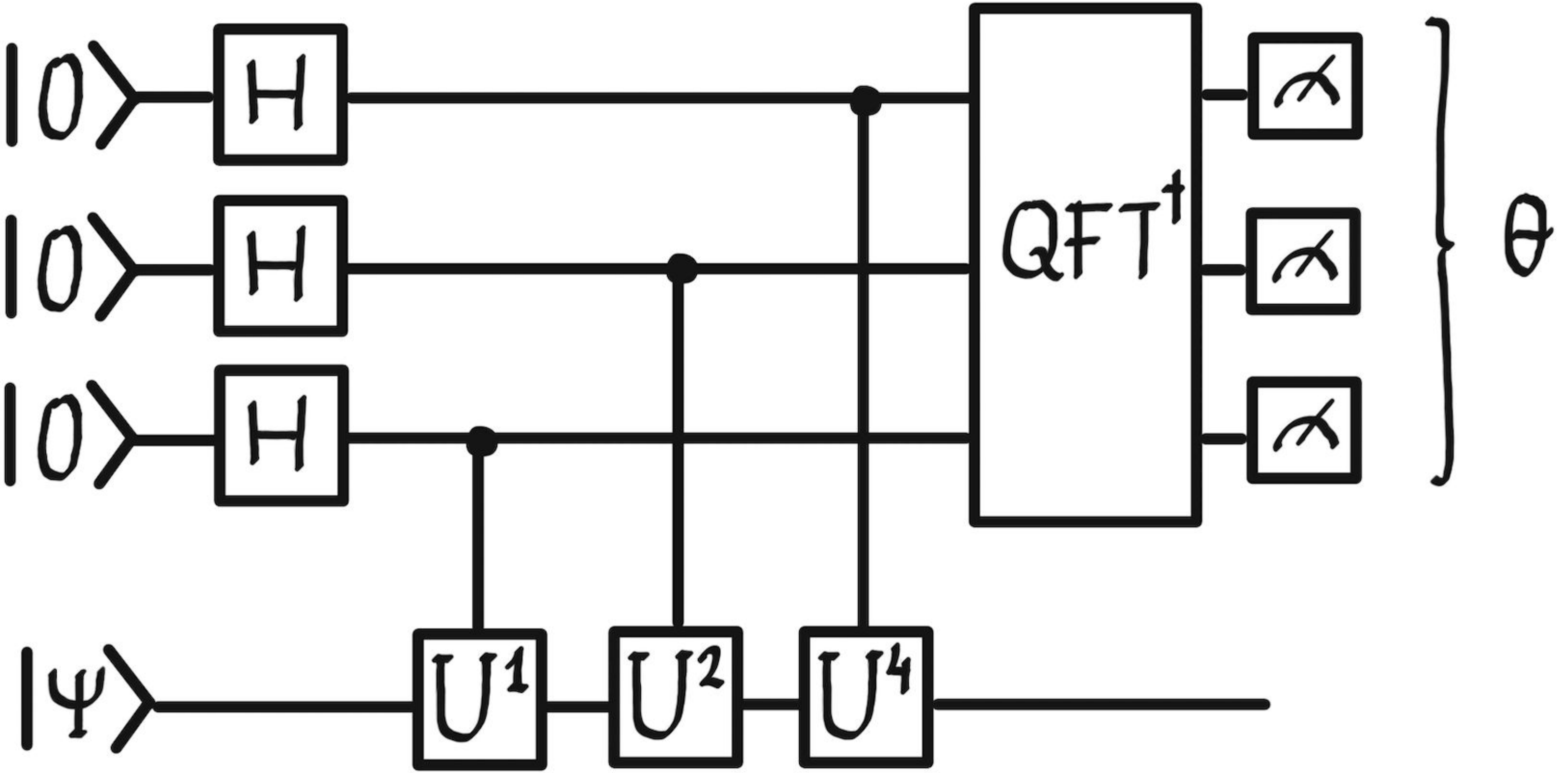
This looks like QFT



$$\frac{1}{\sqrt{N}}\left(|0\rangle+e^{2\pi i(0.j_n)}\,|1\rangle\right)\otimes\cdots\otimes\left(|0\rangle+e^{2\pi i(0.j_2j_3\cdots j_n)}\,|1\rangle\right)\left(|0\rangle+e^{2\pi i(0.j_1j_2\cdots j_n)}\,|1\rangle\right)$$

Furthermore, as $e^{2m\pi i}=1$ for any integer m, this equals:

$$\frac{1}{\sqrt{N}}\bigotimes_{l=1}^{n}\left(|0\rangle+e^{2\pi i j 2^{-l}}\,|1\rangle\right)$$


|0⟩
|0⟩
|0⟩
|Ψ⟩
H
H
H
U¹
U²
U⁴
QFT†
θ

Part 1: Representing the phase

A first step is to find a quantum circuit that performs the transformation

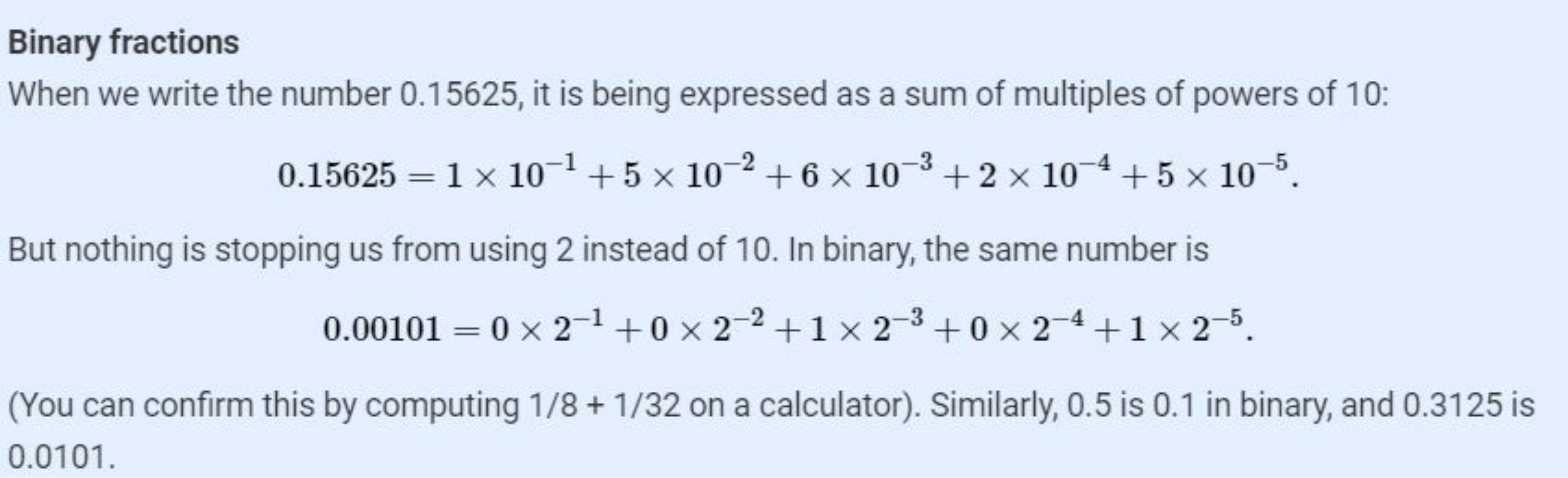
$$|\psi\rangle|0\rangle \rightarrow |\psi\rangle|\phi\rangle$$

We could then obtain ϕ directly by measuring the second register. We call this the estimation register.

Part 1: Representing the phase

Complex exponential has period 2π, technically the phase is not unique.

Instead, we define $\phi=2\pi\theta$ so that θ is a number between 0 and 1; this forces ϕ to be between 0 and 2π



Binary fractions

When we write the number 0.15625, it is being expressed as a sum of multiples of powers of 10:

$$0.15625 = 1 \times 10^{-1} + 5 \times 10^{-2} + 6 \times 10^{-3} + 2 \times 10^{-4} + 5 \times 10^{-5}.$$

But nothing is stopping us from using 2 instead of 10. In binary, the same number is

$$0.00101 = 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5}.$$

(You can confirm this by computing 1/8 + 1/32 on a calculator). Similarly, 0.5 is 0.1 in binary, and 0.3125 is 0.0101.

Part 1: Representing the phase

A binary representation is useful because we can encode it using qubits, e.g., $|110010\rangle$ for $\theta=0.110010$.

The phase is retrieved by measuring the qubits.

The precision of the estimate is determined by the number of qubits.

We've used examples of fractions that can be conveniently expressed exactly with just a few binary points, but this won't always be possible.

The binary expansion of 0.8 is 0.11001100… which does not terminate.

From now on, we'll use n for the number of estimation qubits.

Part 2: Quantum Fourier Transform

The second clever part of the algorithm is to follow advice given to many physicists: “When in doubt, take the quantum Fourier transform (QFT)”.

$$\mathrm{QFT}|\theta\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0} e^{2\pi i\theta k}|k\rangle.$$

Part 2: Quantum Fourier Transform

Note that this results in a uniform superposition, where each basis state has an additional phase.

If we can prepare that state, then applying the inverse QFT would give |θ⟩ in the estimation register.

This looks more promising, especially if we notice the appearance of the eigenvalues e2πiθ, although with an extra factor of k

$$\mathrm{QFT}|\theta\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0} e^{2\pi i \theta k} |k\rangle.$$

Part 2: Quantum Fourier Transform

We can obtain this factor by applying the unitary k times to the state |ψ⟩:

$$U^k|\psi\rangle = e^{2\pi i\theta k}|\psi\rangle$$

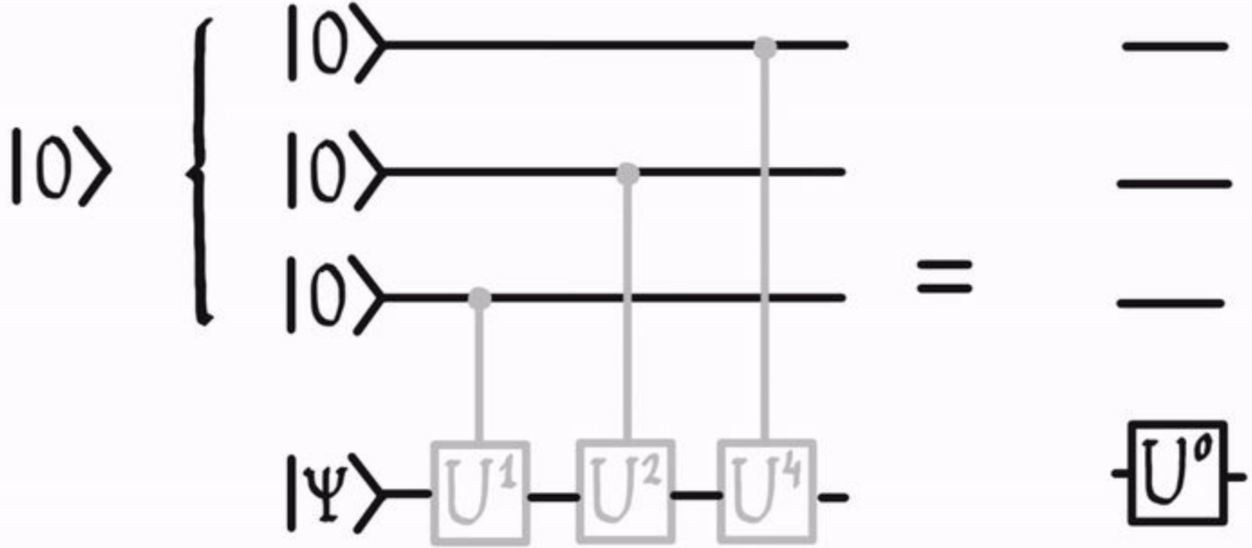
$$|\psi\rangle|k\rangle \to U^k|\psi\rangle|k\rangle$$

$$\frac{1}{\sqrt{2^n}}\sum_{k=0}|\psi\rangle|k\rangle \to \frac{1}{\sqrt{2^n}}\sum_{k=0}U^k|\psi\rangle|k\rangle = |\psi\rangle\frac{1}{\sqrt{2^n}}\sum_{k=0}e^{2\pi i\theta k}|k\rangle$$

Part 3: Controlled sequence

The power of U is the same as the binary representation of the corresponding basis state:

$$|\psi\rangle|00\rangle + |\psi\rangle|01\rangle + |\psi\rangle|10\rangle + |\psi\rangle|11\rangle \rightarrow |\psi\rangle|00\rangle + U|\psi\rangle|01\rangle + U^2|\psi\rangle|10\rangle + U^3|\psi\rangle|11\rangle.$$


|0⟩
|0⟩
|0⟩
|0⟩
|Ψ⟩
U^1
U^2
U^4
=
U^0

Quantum Phase Estimation

1. Start with the state |ψ⟩|0⟩. Apply a Hadamard gate to all estimation qubits to implement the transformation

$$|\psi\rangle|0\rangle \to |\psi\rangle \frac{1}{\sqrt{2^n}} \sum_{k=0} |k\rangle$$

2. Apply a ControlledSequence operation, i.e., U^(2^m) controlled on the m-th estimation qubit. This gives

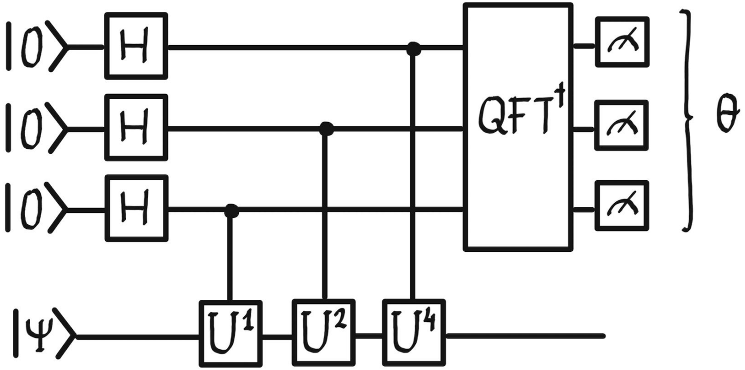
$$|\psi\rangle \frac{1}{\sqrt{2^n}} \sum_{k=0} |k\rangle \to |\psi\rangle \frac{1}{\sqrt{2^n}} \sum_{k=0} e^{2\pi i\theta k}|k\rangle.$$

Quantum Phase Estimation

3. Apply the inverse quantum Fourier transform to the estimation qubits

$$|\psi\rangle \frac{1}{\sqrt{2^n}} \sum_{k=0} e^{2\pi i\theta k}|k\rangle \rightarrow |\psi\rangle|\theta\rangle$$

4. Measure the estimation qubits to recover θ



Quantum Phase Estimation

- QPE enables the phase of an eigenvalue to be estimated to an arbitrary number of bits of precision.
- It can be shown that the estimate is good even when the phase cannot be exactly expanded as a binary fraction.
- QPE is at the heart of quantum chemistry and many quantum computing algorithms.
- As QPE uses the oracle U and the prepared state |ui it should be thought of as a subroutine that can be called, rather than an entire algorithm in and of itself.

References for QPE

https://pennylane.ai/qml/demos/tutorial_qpe/

https://pennylane.ai/codebook/09-quantum-phase-estimation/01-catch-the-phase

https://medium.com/quantum-untangled/kitaevs-phase-estimation-qpe-algorithms-b1cc6a1c9cab

https://jonathan-hui.medium.com/qc-phase-estimation-in-shors-algorithm-acef265ebe50

https://pennylane.ai/qml/demos/tutorial_qpe/

https://pennylane.ai/codebook/09-quantum-phase-estimation/02-its-not-just-a-phase

https://pennylane.ai/codebook/09-quantum-phase-estimation/03-lets-be-rational

Let's make Some Shor

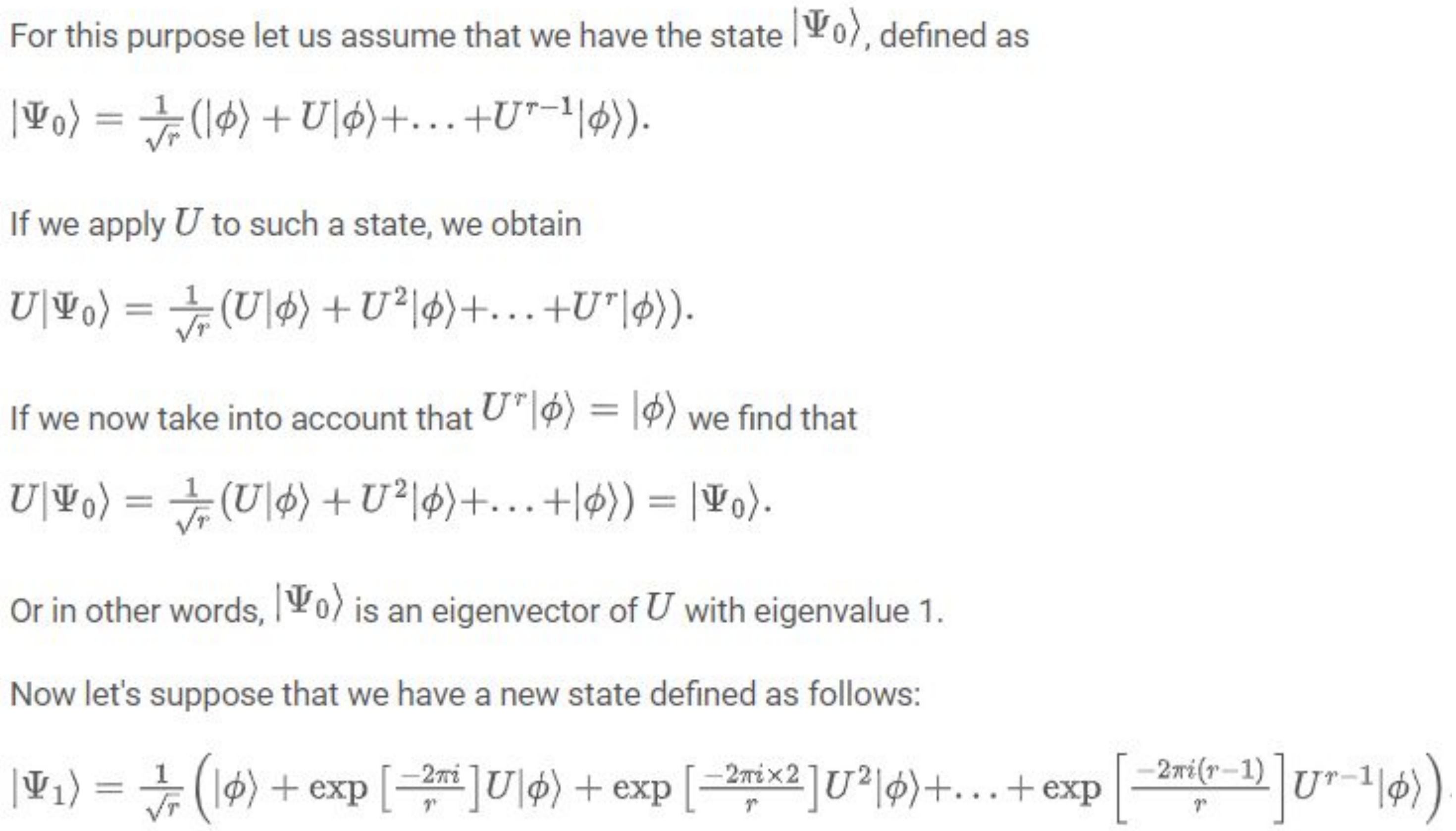
In 1994 Peter Shor invented a quantum algorithm which can factor numbers in polynomial time. This remains one of the (or probably the) most important and impressive potential application of quantum computing.

Period Finding

This algorithm determines the period of a given state $|\phi\rangle$ with respect to an operator U, that is, it finds the minimum positive integer r such that

$$U^r|\phi\rangle = |\phi\rangle.$$

Period Finding



For this purpose let us assume that we have the state $|\Psi_0\rangle$, defined as

$$|\Psi_0\rangle = \frac{1}{\sqrt{r}}(|\phi\rangle + U|\phi\rangle + \ldots + U^{r-1}|\phi\rangle).$$

If we apply U to such a state, we obtain

$$U|\Psi_0\rangle = \frac{1}{\sqrt{r}}(U|\phi\rangle + U^2|\phi\rangle + \ldots + U^r|\phi\rangle).$$

If we now take into account that $U^r|\phi\rangle = |\phi\rangle$ we find that

$$U|\Psi_0\rangle = \frac{1}{\sqrt{r}}(U|\phi\rangle + U^2|\phi\rangle + \ldots + |\phi\rangle) = |\Psi_0\rangle.$$

Or in other words, $|\Psi_0\rangle$ is an eigenvector of U with eigenvalue 1.

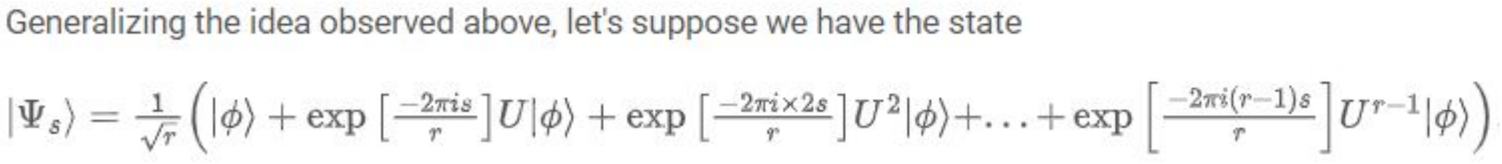
Now let's suppose that we have a new state defined as follows:

$$|\Psi_1\rangle = \frac{1}{\sqrt{r}}\left(|\phi\rangle + \exp\left[\frac{-2\pi i}{r}\right]U|\phi\rangle + \exp\left[\frac{-2\pi i\times 2}{r}\right]U^2|\phi\rangle + \ldots + \exp\left[\frac{-2\pi i(r-1)}{r}\right]U^{r-1}|\phi\rangle\right).$$

$$|\Psi_1\rangle = \frac{1}{\sqrt{r}}\left(|\phi\rangle + \exp\left[\frac{-2\pi i}{r}\right]U|\phi\rangle + \exp\left[\frac{-2\pi i\times 2}{r}\right]U^2|\phi\rangle + \ldots + \exp\left[\frac{-2\pi i(r-1)}{r}\right]U^{r-1}|\phi\rangle\right).$$

We can see that in this case, we are defining it in a very similar way to the $|\Psi_0\rangle$ state but with an extra phase in each of its terms. If we apply U in the same manner as before, we will find that $|\Psi_1\rangle$ is also an eigenvector of U, but associated with a different eigenvalue.

Period Finding



Generalizing the idea observed above, let's suppose we have the state

$$|\Psi_s\rangle = \frac{1}{\sqrt{r}}\left(|\phi\rangle + \exp\left[\frac{-2\pi i s}{r}\right]U|\phi\rangle + \exp\left[\frac{-2\pi i\times 2s}{r}\right]U^2|\phi\rangle + \ldots + \exp\left[\frac{-2\pi i(r-1)s}{r}\right]U^{r-1}|\phi\rangle\right)$$

Where s is an integer between 0 to r-1

$$|\Psi\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\Psi_s\rangle = |\phi\rangle.$$

This surprising result is a consequence of the following equality:

$$\sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i k}{r}\right] = 0.$$

$$
|\psi\rangle = \frac{1}{\sqrt{r}} \cdot \begin{array}{l} |\psi_0\rangle + \\ + |\psi_1\rangle + \\ \quad\vdots \\ + |\psi_{r-1}\rangle \end{array} = \frac{1}{\sqrt{r}} \cdot \begin{array}{l}
1/\sqrt{r}\,\left(|\phi\rangle + 1\cdot U|\phi\rangle + \dots + 1\cdot U^{r-1}|\phi\rangle\right) \\
+ 1/\sqrt{r}\,\left(|\phi\rangle + e^{-\frac{2\pi i}{r}} U|\phi\rangle + \dots + e^{-\frac{2\pi i (r-1)}{r}} U^{r-1}|\phi\rangle\right) \\
\quad\vdots \\
+ 1/\sqrt{r}\,\left(|\phi\rangle + e^{-\frac{2\pi i (r-1)}{r}} U|\phi\rangle + \dots + e^{-\frac{2\pi i (r-1)^2}{r}} U^{r-1}|\phi\rangle\right)
\end{array}
$$

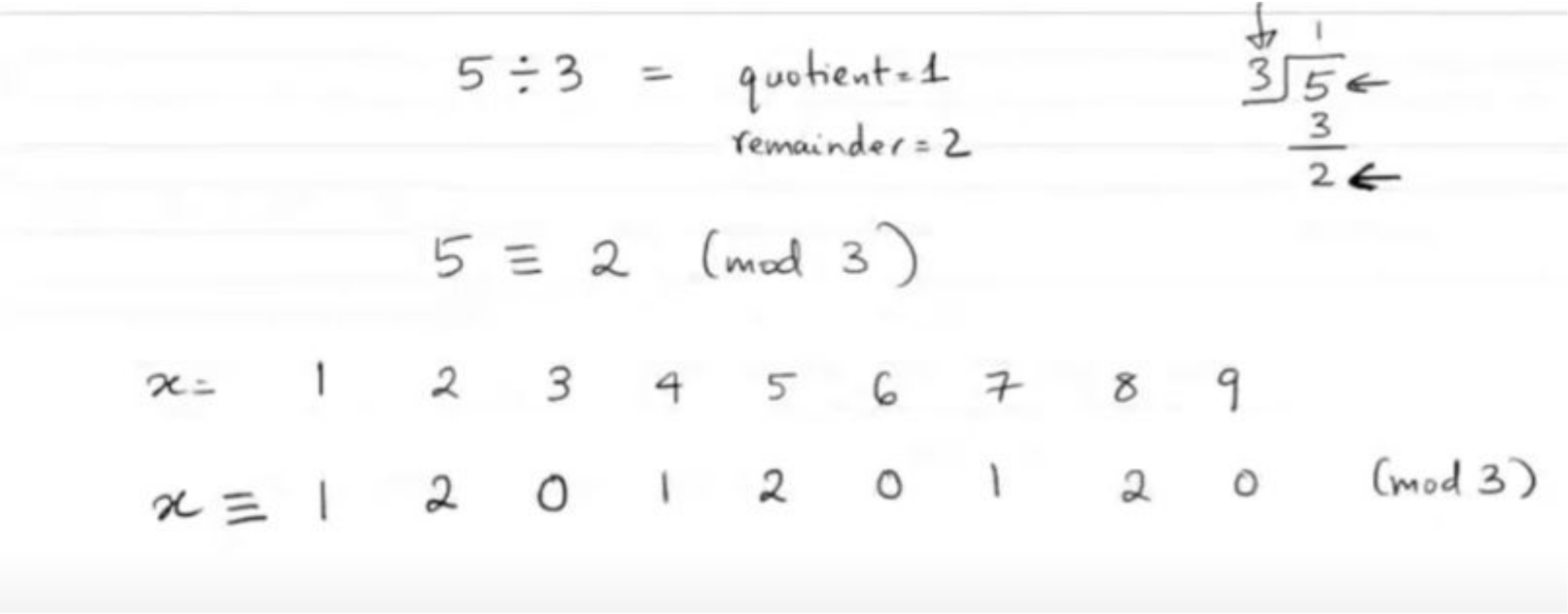
$$
\underbrace{\begin{array}{ccccccc} \downarrow & & \downarrow & & \downarrow & & \downarrow \\ r & & 0 & \cdots & 0 & \cdots & 0 \end{array}}_{=\ 1/\sqrt{r}\,\cdot\, 1/\sqrt{r}\,\cdot\, r\,|\phi\rangle\ =\ \boxed{|\phi\rangle}}
$$

$$|\Psi\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\Psi_s\rangle = |\phi\rangle.$$

This surprising result is a consequence of the following equality:

$$\sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i k}{r}\right] = 0.$$

Modular Arithmetic



$5 \div 3 =$ quotient $= 1$

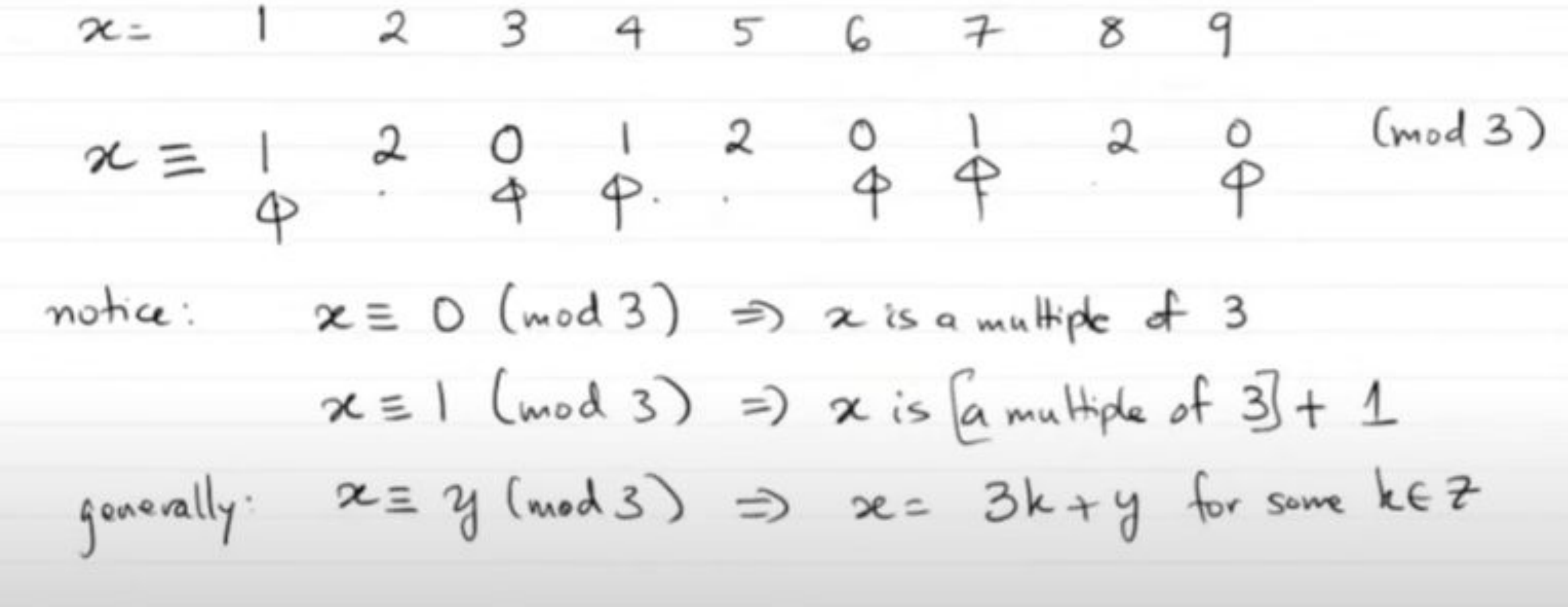
remainder $= 2$

$$\begin{array}{r} \downarrow \quad 1 \\ 3\overline{)\,5} \leftarrow \\ 3 \\ \hline 2 \leftarrow \end{array}$$

$$5 \equiv 2 \pmod 3$$

$x =$	1	2	3	4	5	6	7	8	9	
$x \equiv$	1	2	0	1	2	0	1	2	0	(mod 3)

Modular Arithmetic

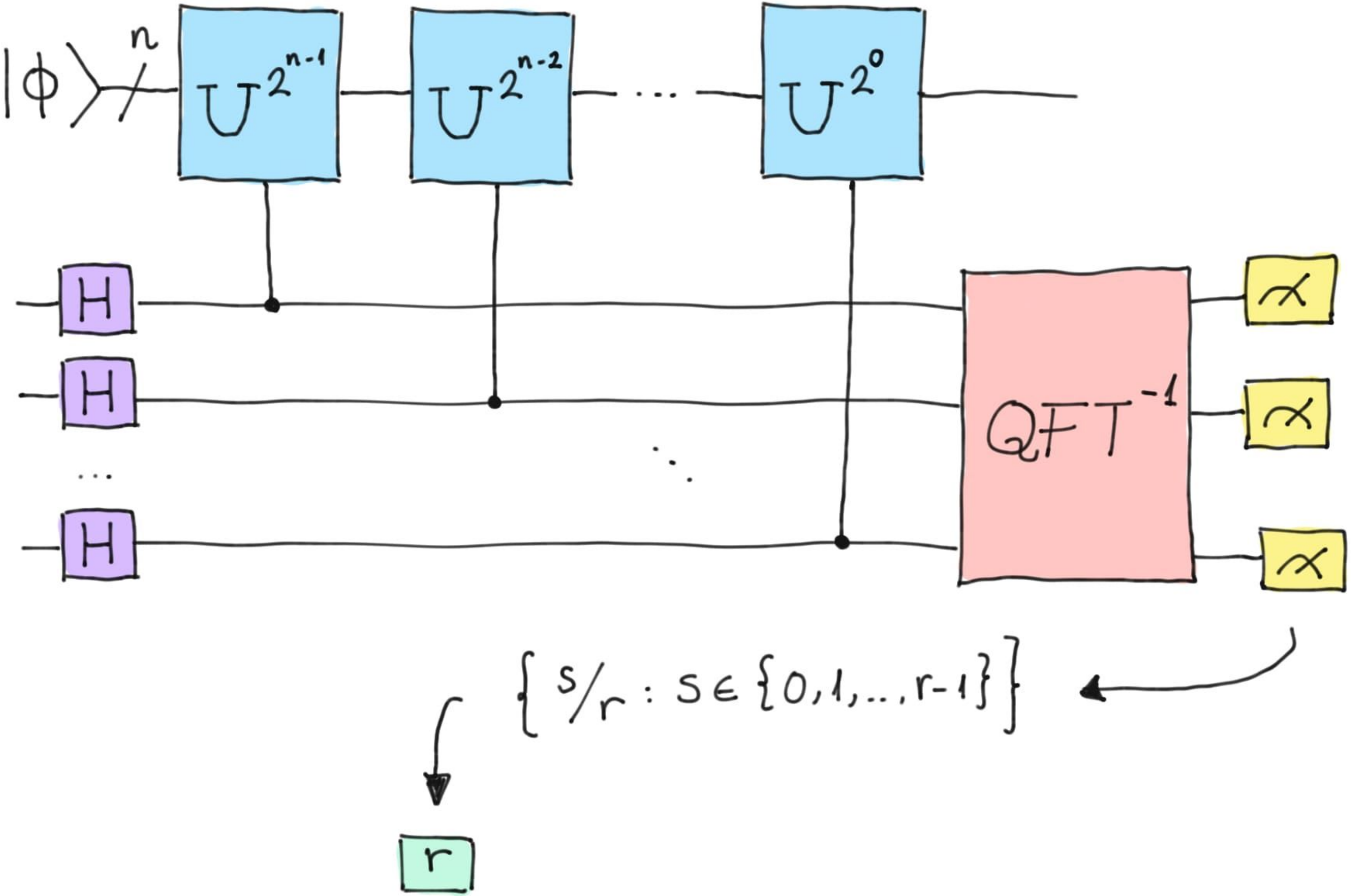


$x =$	1	2	3	4	5	6	7	8	9	
$x \equiv$	1	2	0	1	2	0	1	2	0	(mod 3)
	ϕ	.	ϕ	ϕ.	.	ϕ	ϕ	.	ϕ	

notice: $x \equiv 0 \pmod 3 \Rightarrow x$ is a multiple of 3

$x \equiv 1 \pmod 3 \Rightarrow x$ is [a multiple of 3] + 1

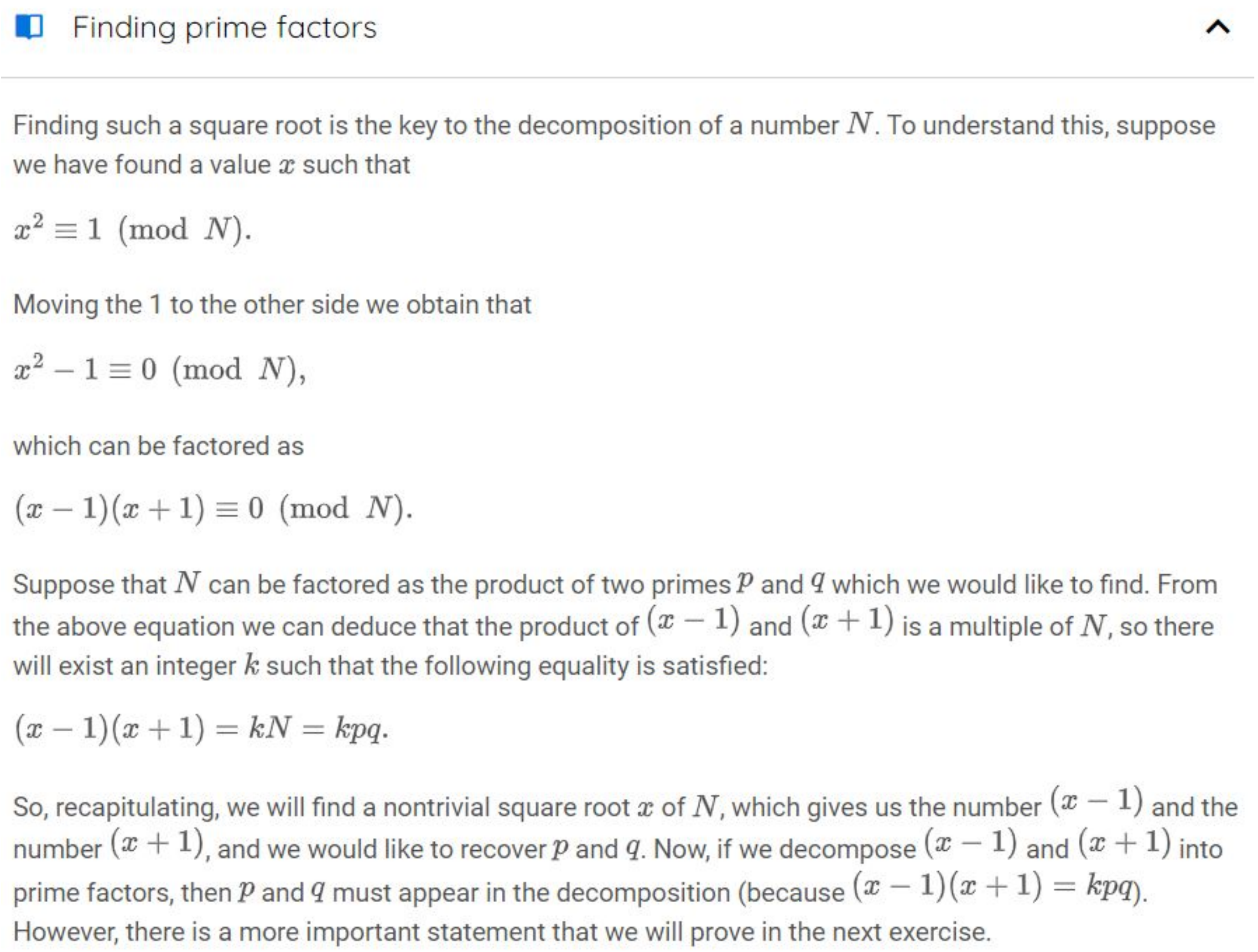
generally: $x \equiv y \pmod 3 \Rightarrow x = 3k + y$ for some $k \in \mathbb{Z}$


$|\phi\rangle$
n
$U^{2^{n-1}}$
$U^{2^{n-2}}$
...
U^{2^0}
H
H
...
H
QFT^{-1}
$\{s/r : s \in \{0,1,\dots,r-1\}\}$
r

$U|y\rangle = |a \cdot y \mod N\rangle.$ $\quad \exp\left[\frac{2\pi i s}{r}\right]$ $\quad \frac{s}{r}$ = phase

$$U|\psi_s\rangle = \underbrace{e^{2\pi i \phi}}_{\text{eigenvalue}} |\psi_s\rangle$$

$$U|u_s\rangle = \underbrace{\exp\left[\frac{2\pi i s}{r}\right]}_{\text{eigenvalue}} |u_s\rangle \qquad \text{with} \qquad \underbrace{|u_s\rangle}_{\text{eigenvector}} = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i s k}{r}\right] |x^k \mod N\rangle$$



Finding prime factors

Finding such a square root is the key to the decomposition of a number N. To understand this, suppose we have found a value x such that

$$x^2 \equiv 1 \pmod{N}.$$

Moving the 1 to the other side we obtain that

$$x^2 - 1 \equiv 0 \pmod{N},$$

which can be factored as

$$(x-1)(x+1) \equiv 0 \pmod{N}.$$

Suppose that N can be factored as the product of two primes p and q which we would like to find. From the above equation we can deduce that the product of $(x-1)$ and $(x+1)$ is a multiple of N, so there will exist an integer k such that the following equality is satisfied:

$$(x-1)(x+1) = kN = kpq.$$

So, recapitulating, we will find a nontrivial square root x of N, which gives us the number $(x-1)$ and the number $(x+1)$, and we would like to recover p and q. Now, if we decompose $(x-1)$ and $(x+1)$ into prime factors, then p and q must appear in the decomposition (because $(x-1)(x+1) = kpq$). However, there is a more important statement that we will prove in the next exercise.

$\mathrm{GCD}(150, 250) = 50.$

This is because $150 = 5 \cdot 5 \cdot 3 \cdot 2$ and $250 = 5 \cdot 5 \cdot 5 \cdot 5 \cdot 2$. The common factors are 5,5 and 2, which we can multiply together to obtain 50. In addition, this algorithm can be efficiently implemented in a classical way through what is known as **Euclid's algorithm**.

We know $N = pq$ and $(x+1) = sp$, with s being an integer whose decomposition will not depend on q. Knowing this, calculating the greatest common divisor we will get that:

$p = \mathrm{GCD}((x-1), N),$

and similarly, $q = \mathrm{GCD}((x+1), N).$

Shor's Algorithm

Step: 1 Input N, the integer you want to factorize.

Step: 2 Randomly choose an integer k such that 1<a<N-1.

Step: 3 Compute gcd(N,k)

If gcd(a,N)≠1, then N has a factor (namely, gcd(a,N)), and you're done. Otherwise, proceed to the next step.

Step: 4 We need to find **smallest positive integer r** such that if **$f(x) = a^x$ mod N, then f(k) = f(k+r)**

Step: 4.1 Define a new variable q=1

Step: 4.2 Find (q*k) mod N

If remainder ≠1, then set the value of q to the value of remainder we got, repeat this until remainder is 1.

Otherwise, proceed to the next step.

Step: 4.3 The number of transformations you did in Step 4.2 is your value of r.

Shor's Algorithm

Step: 5 If r is odd, go back to Step 2 and choose a different value of k.

Step: 6 Define p = remainder in (r/2)th transformation.

If p + 1 = N, then go back to Step 2 and choose a different value of k. Else, proceed to Step 7.

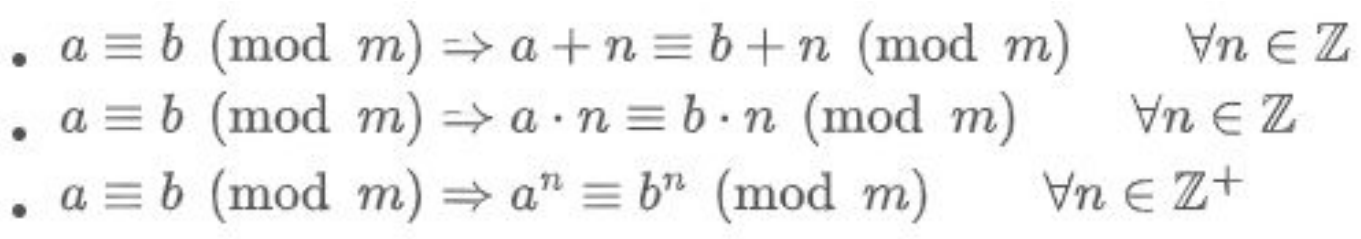
Step: 7 This is the final step. The factors of N are

f_1 = GCD (p+1, N)

f_2 = GCD (p-1, N)

We have covered numerical examples in class from the attached link in Reference.

Modular Arithmetic



- $a \equiv b \pmod{m} \Rightarrow a + n \equiv b + n \pmod{m} \qquad \forall n \in \mathbb{Z}$
- $a \equiv b \pmod{m} \Rightarrow a \cdot n \equiv b \cdot n \pmod{m} \qquad \forall n \in \mathbb{Z}$
- $a \equiv b \pmod{m} \Rightarrow a^n \equiv b^n \pmod{m} \qquad \forall n \in \mathbb{Z}^+$

https://jonathan-hui.medium.com/qc-period-finding-in-shors-algorithm-7eb0c22e8202