Quantum Fourier-transform

Did you already encounter the classical Fourier transform?

→ Many applications in Mathematics, physics
computer science

(Discrete) Fourier transform

→ implements a basis transform via a unitary $N \times N$ matrix F

→ All entries of F should have the same absolute value: $|F_{ij}| = \text{const.}$

→ F unitary $\Leftrightarrow$ all columns (& rows) are orthonormal

Interlude: roots of unity

Consider the sequence $(\alpha^n)_{n \in \mathbb{N}}$ for $\alpha \in \mathbb{C}$
if $\alpha^N = 1$ for some $N \in \mathbb{N}$ the α is called an N-th root of unity.

Examples: • $\alpha = 1 \Rightarrow \alpha^1 = 1$

• $\alpha = -1 \Rightarrow \alpha^2 = 1$

What about $\alpha = i$?

In general set $\omega_N = e^{2\pi i/N}$

then $(\omega_N)^j = e^{2\pi i \frac{j}{N}} = \cos\left(\frac{2\pi j}{N}\right) + i \sin\left(\frac{2\pi j}{N}\right)$

$= 1$ if $\frac{j}{N}$ is an integer.

Discrete Fourier transform

Fix $N = 2^n$ $\qquad \omega_N = e^{\frac{2\pi i}{N}}$

$$(F_N)_{i,j} = \frac{(\omega_N)^{i \cdot j}}{\sqrt{N}}$$

$$F_N = \frac{1}{\sqrt{N}} \begin{array}{cc} \begin{pmatrix} & & \\ & (\omega_N)^{i\cdot j} & \\ & & \end{pmatrix} & \Big\downarrow\ i \in \{0, \dots, N-1\} \\ \xrightarrow{} & \\ j \in \{0, \dots, N-1\} & \end{array}$$

Example: $N=2 \Rightarrow \omega = -1$

$$F_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Determine: F_4 for $\omega_4 = e^{\frac{2\pi i}{4}} = e^{\frac{\pi i}{2}} = i$

$$F_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

Fact: F_N is unitary

$\Rightarrow$ F_N can be implemented on a quantum computer.

How? later (+ exercises)

We call the quantum circuit that implements F_N on n qubits the quantum Fourier transform QFT_N

$$|\psi\rangle = \sum_{i \in \{0,1\}^n} \alpha_i |i\rangle \xmapsto{QFT_N} \sum_{j \in \{0,1\}^n} \overbrace{\tilde{\alpha}_j}^{= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{j \cdot k} \alpha_k = \left(F_N \cdot \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_N \end{pmatrix}\right)_j} |j\rangle$$

Observation: QFT produces a state vector with amplitudes $\hat{\alpha}_j$ corresponding to the discrete Fourier transform

but: measurement will only reveal random $\hat{\alpha}_j$

In particular: $|k\rangle \longmapsto F_N |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{j \cdot k} |j\rangle$

Note: $j = j_1, \dots, j_n \Leftrightarrow j = \sum_{l=1}^{N} j_l 2^{n-l}$

$\Rightarrow \frac{j}{2^n} = \sum_{l=1}^{N} j_l 2^{-l}$

$\Rightarrow F_N |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i \left(\sum_{l=1}^{n} j_l 2^{-l}\right) \cdot k} |j_1 \dots j_n\rangle$

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

Find QFT_4 of

$$|\varphi_1\rangle = \frac{1}{2}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle\right) = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \qquad |\hat{\varphi}_1\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|\varphi_2\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |\hat{\varphi}_2\rangle = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$|\varphi_3\rangle = |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad |\hat{\varphi}_3\rangle = \begin{pmatrix} 1 \\ i \\ -1 \\ -i \end{pmatrix}$$

Observations:

- spread out vectors such as $|\psi_1\rangle$ get mapped to sparse and focused vectors and vice versa
- The shift between $|\psi_2\rangle$ and $|\psi_3\rangle$ resulted in relative phase in the QFT.

In general:

$$QFT_N \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_N \end{pmatrix} \quad \text{then} \quad QFT_N \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_0 \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \omega\, \beta_1 \\ \vdots \\ \omega^{N-1} \beta_{N-1} \end{pmatrix}$$

Homework: check this property for $N=4$

Application: Phase estimation

Problem: Given a unitary U with eigenvector $|\psi\rangle$ s.t. $U|\psi\rangle = \lambda|\psi\rangle$

Determine λ

Observation I: $\lambda = e^{i2\pi\theta}$, $\theta \in [0,1)$

assume: $\theta = \underbrace{0.\theta_1 \ldots\ldots \theta_n}_{\text{binary representation}}$

Phase estimation algorithm

1) Initialize in $|0\rangle^{\otimes n}|\psi\rangle$

2) Apply $H^{\otimes n}$ on $|0\rangle^{\otimes n}$

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

3) Apply a controlled unitary:

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

Result: $\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \left(e^{2\pi i\theta}\right)^j |j\rangle|\psi\rangle$

4) Apply QFT_N^{-1} to first register

Observation:

$$\theta = 0.\theta_1, \dots, \theta_n$$

is equivalent to $\dfrac{2^{n-1}\theta_1 + 2^{n-2}\theta_2 + \dots + \theta_n}{2^n}$

$$\Rightarrow \left(e^{2\pi i \theta}\right)^s = \left(e^{2\pi i \frac{2^n\theta}{2^n}}\right)^s$$

Therefore:

$$QFT_N |\theta_1, \dots, \theta_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} \left(e^{2\pi i \, 0.\theta_1\theta_2 \dots \theta_n}\right)^s |s\rangle$$

$\Rightarrow$ Apply $QFT_N^{\#}$ in the last step yields

$$|\theta_1 \dots \theta_n\rangle |\psi\rangle$$

Hence, phase estimation implements

$$|0^n\rangle|\psi\rangle \longmapsto |\theta_1, \dots \theta_n\rangle |\psi\rangle$$

measure to obtain $\theta_1 \dots \theta_n$

QFT circuit

Classical fast Fourier transform

set $N = 2^n$

Can we perform Fourier transform more efficient than $\mathcal{O}(n^2)$ ≙ matrix-vector multiplication $F_N v$?

Key insight:

$$\hat{v}_j = (F_N v)_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{j \cdot k} v_k$$

$$= \frac{1}{\sqrt{N}} \left(\sum_{k\ \text{even}} \omega_N^{jk} v_k + \omega_N \sum_{k\ \text{odd}} \omega_N^{j(k-1)} v_k \right)$$

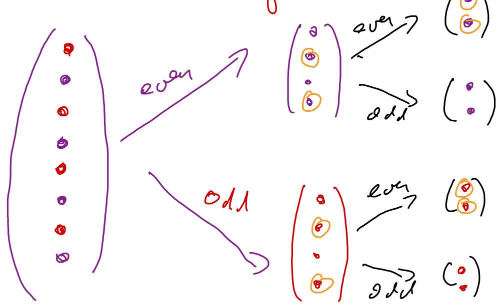
$$= \frac{1}{\sqrt{N}} \left(\sum_{k\ \text{even}} \omega_{\frac{N}{2}}^{j\frac{k}{2}} v_k + \omega_N \sum_{k\ \text{odd}} \omega_{\frac{N}{2}}^{j\frac{(k-1)}{2}} v_k \right)$$

$$= \frac{1}{\sqrt{2}} \Bigg(\underbrace{\frac{1}{\sqrt{\frac{N}{2}}} \sum_{k'=0}^{\frac{N}{2}-1} \omega_{\frac{N}{2}}^{jk'} v_{2k'}}_{\left(F_{\frac{N}{2}} v_{even}\right)_j} + \omega_N \underbrace{\sum_{k'=0}^{\frac{N}{2}-1} \omega_{\frac{N}{2}}^{j \cdot k'} v_{2k'+1}}_{\left(F_{\frac{N}{2}} v_{odd}\right)_j} \Bigg)$$

$$\Rightarrow \hat{v}_j = \left(\hat{v}_{even,j} + \omega_N \hat{v}_{odd,j} \right) / \sqrt{2}$$

Using this relation recursively, we can compute the Fourier transform in $\mathcal{O}(N \log N)$ relevant in practice!

idea $v =$



If we know $\hat{U}_{odd}$ & $\hat{U}_{even}$ perform 2 additions & 1 multiplication to obtain $\hat{U}_j$ $\Rightarrow$ $O(N)$ for N entries

$\Rightarrow$ if $T(N)$ is the cost of computing $F_N U$ then

$$T(N) = \underbrace{2\,T(N/2)}_{\substack{\text{cost for}\\ \text{computing } \hat{U}_{odd}\\ \& \ \hat{U}_{even}}} + O(N)$$

Repeat recursively:

$$T\left(\tfrac{N}{2}\right) = 2\,T\left(\tfrac{N}{4}\right) + O(N)$$

$$\Rightarrow \quad T(N) = 2^3\,T\left(\tfrac{N}{4}\right) + 2\,O(N)$$

Iterate $n = \log N$ times

$$T(N) = N\,T(2) + \log(N)\,O(N)$$

$$= O(N \log(N))$$

<u>Quantum</u>: $O(n^2)$!

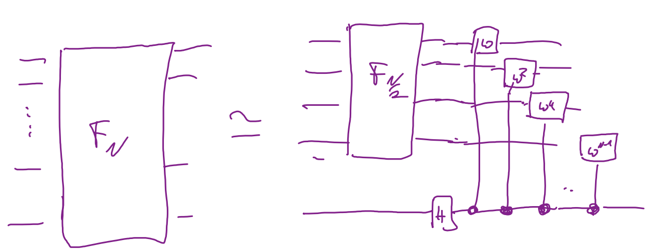
Same approach as FFT:

compute $F_{N/2}$ on even/odd $\rightarrow$ can be done in parallel

then apply additional phase

odd instances

$n-1$ qubits



least significant bit of

$|i\rangle = (i_0 \cdots i_{n-1})$

decides even/odd

Can improve by omitting gates close to the identity $\rightsquigarrow O(n \log n)$

Application: HHL

Problem: given $A \in \mathbb{C}^n \times \mathbb{C}^n$, $b \in \mathbb{C}$

Find $x \in \mathbb{C}^n$ s.t. $Ax = b$

Quantum: output a state $|\tilde{x}\rangle$ s.t.

$\| |x\rangle - |\tilde{x}\rangle \| \leq \varepsilon$ $\quad A|x\rangle = |b\rangle$

Interlude: Solving linear equations

Simplest algorithm: L-U decomposition

$$A = L \cdot U$$

Runtime $\mathcal{O}(3)$ + Savings based on add. structure

Sparsity: A has only few non-zero entries

Column sparsity s: every col. of A has at most s non-zero elements

Fact: If A has column sparsity s, we can compute Ax in $\mathcal{O}(n \cdot s)$ as opposed to $\mathcal{O}(n^2)$

Ex: Conjugate gradient algorithm

- → A pos. semi-definite
- → each iteration requires matrix-vector multipl. $\hat{=}\ \mathcal{O}(n \cdot s)$
- → Overall runtime: $\mathcal{O}(\kappa(A)\, n \cdot s)$

$\hat{=}\ \mathcal{O}(\kappa(A))$ iterations

Def: Condition number

$$\kappa(A) = |\lambda_{max}| \cdot |\lambda_{min}^{-1}|$$

HHL (most basic version):

$$\mathcal{O}(\kappa(A)^2 \cdot s \cdot \log(n)) \leftarrow \text{exp. savings}$$

HHL assumptions

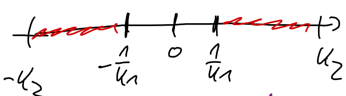
1) A is Hermitian   else consider $\begin{bmatrix} 0 & A^{\dagger} \\ A & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$

   solution $x_1 = A^{-1} b$
   $x_2 = 0$ ✓

2) Can prepare $\frac{|b\rangle}{\| |b\rangle \|}$ as quantum state

3) Can implement e^{iAt} $\hat{=}$ unitary for A Hermitian

4) A is well conditioned



Intuition: $A = U \overset{\text{diagonal}}{D} U^{\dagger} \Rightarrow A^{-1} = U^{*} D^{-1} U^{\dagger}$

goal: implement $\frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}$

if $|a_i\rangle$ eigenvectors: $A|a_i\rangle = \lambda_i |a_i\rangle$

and $|b\rangle = \sum_i \beta_i |a_i\rangle$

target state: $A^{-1}|b\rangle = \sum_i \frac{\beta_i}{\lambda_i} |a_i\rangle$

$e^{itA} = \sum_{k=0}^{\infty} \frac{1}{k!} (iAt)^k$ has same eigenvectors as A

$= U e^{itD} U^{\dagger} \Rightarrow e^{itA}|a_i\rangle = e^{it\lambda_i}|a_i\rangle$

Recap: Phase estimation: given U, $|\xi\rangle$ $U|\xi\rangle = e^{i2\pi\lambda}|\xi\rangle$ ($\lambda = 0.\lambda_1 \dots \lambda_n$)

$|\xi\rangle|0\rangle \longmapsto |\xi\rangle|\lambda\rangle$ with $\lambda = 2^n \lambda_1 \cdots \lambda_n$

Fact: define $Q(\lambda) = \begin{bmatrix} \frac{\kappa}{\lambda} & \sqrt{1-\left(\frac{\kappa}{\lambda}\right)^2} \\ -\sqrt{1-\left(\frac{\kappa}{\lambda}\right)^2} & \frac{\kappa}{\lambda} \end{bmatrix}$

Q is unitary for $|\lambda| > |\kappa|$

HHL: initial state

1) $|b\rangle |0\rangle^{\otimes n} |0\rangle$

2) Apply phase estimation

$$\sum_i \beta_i |a_i\rangle |0\rangle^{\otimes n} |0\rangle \mapsto \sum_i \beta_i |a_i\rangle |\lambda_i\rangle |0\rangle$$

3.) Apply controlled $Q(\lambda_j)$ to third register controlled by second

$$\Rightarrow \sum_j \beta_j |a_j\rangle |\lambda_j\rangle \left(\frac{\kappa}{\lambda_j} |0\rangle + \sqrt{1-\left(\frac{\kappa^2}{\lambda_j}\right)} |1\rangle \right)$$

$$= \sum_j \beta_j \frac{\kappa}{\lambda_j} |a_j\rangle |\lambda_j\rangle |0\rangle$$

$$+ \sum_j \beta_j \sqrt{1-\left(\frac{\kappa}{\lambda_j}\right)^2} |a_j\rangle |\lambda_j\rangle |1\rangle$$

4) Measure third register if result is 0 remaining state given as

$$\sum_j \kappa \frac{\beta_j}{\lambda_j} |a_j\rangle |\lambda_j\rangle |0\rangle$$

outcome 1 $\Rightarrow$ try again.

Success propapility

$$p_0 = \sum_i \kappa^2 \frac{|\beta_i|^2}{\lambda_i^2} \geq \kappa(A)^2$$

$\Rightarrow$ run $\mathcal{O}(\kappa(A)^2)$ times to see $|0\rangle$

Complexity

$\mathcal{O}((\text{preparation of } b + \underbrace{\text{phase est}}_{\mathcal{O}(\log(n) \cdot s)}) \cdot \kappa^2(A))$

$\Rightarrow \mathcal{O}(s\, \kappa(A)^2 \log(n))$

State of the art: $\mathcal{O}(\sqrt{s}^{\,1+\mathcal{O}(1)} \kappa(A) \log(n))$

need $\mathcal{O}(n)$ runs for classical description