# Introduction to quantum numerical linear algebra

Dong An

Joint Center for Quantum Information and Computer Science,
University of Maryland

*dongan@umd.edu*

September 12, 2023

# Overview

# Introduction

# Quantum numerical linear algebra

- Numerical linear algebra: using matrix operations to design algorithms
  - Operations: Matrix/vector addition, multiplication
  - Tasks: solving linear systems of equations, matrix factorization, eigenvalue/singular value decomposition/transformation
- Quantum numerical linear algebra

# Quantum numerical linear algebra: desired speedup

$$\text{poly}(N) \quad \text{vs} \quad \text{poly}\log(N)$$

- For $N$-dimensional system (suppose $N = 2^n$), classical algorithms typically take cost at least linear in $N$
  - Storing an $N$-dimensional vector takes $\sim N$ cost
  - A single application of matrix-vector multiplication takes at least $\sim N$ computational cost
- For quantum algorithms, we expect computational cost to be $\mathcal{O}(\text{poly}\log(N)) = \mathcal{O}(\text{poly}(n))$
  - An $n$-qubit quantum state can be viewed as a $2^n$-dimensional unit vector
  - Matrix operations are implemented by quantum operations on these $n$ qubits

# Quantum numerical linear algebra: restrictions

- Vectors (quantum states) are normalized under 2-norm
  - May lose some information
- Only a subset of operations are efficiently implementable: unitary matrices
  - For general matrix operations, we will embed its rescaled version into a sub-block of unitary operations
  - Will introduce extra computational cost
- No cloning theorem
  - Iterative methods are not generally efficient for quantum

# Quantum vs Classical

| | Classical | Quantum |
|---|---|---|
| Space | $2^n$ | $n$ |
| Unitary | | ✓ |
| General matrix | ✓ | |
| Copying | ✓ | |
| Entry-wise information | ✓ | |

- Quantum numerical linear algebra: linear algebra algorithms with restrictions but possible speedup
- Speedup for certain tasks:
    - Factorization
    - Unstructured search
    - Discrete Fourier transform
    - Applied math: linear system, differential equation, optimization, machine learning, ...

Quantum algorithm zoo: https://quantumalgorithmzoo.org
Lecture notes by Lin Lin: [arXiv:2201.08309]

# Today's talk

- Basic linear algebra operations
  - Input models for vectors and matrices
  - Matrix-vector multiplication
  - Matrix/vector addition: linear combination of unitaries (LCU)
  - Matrix multiplication
- Linear systems of equations
  - General algorithms: HHL, LCU, adiabatic quantum computing
  - Preconditioning
- Eigenvalue problems
- Matrix functions
  - Functions of Hermitian matrices: quantum signal processing (QSP), qubitization
  - Functions of general matrices: quantum singular value transformation (QSVT)
  - LCU

# Basic linear algebra operations

# Input model: vectors

- Single-qubit state $\cong \mathbb{C}^2/\|\cdot\|_2$

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

  - Measurement: we get 0 with probability $|\alpha|^2$, and 1 with probability $|\beta|^2$
- General $n$ qubit space: tensor product of $n$ multiple single qubit

$$|i_1 i_2 \cdots i_n\rangle = |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle \in \mathbb{C}^{2^n}/\|\cdot\|_2$$

- Common notations:
  - We use $|j\rangle, 0 \leq j \leq 2^n - 1$ to represent the orthonormal basis of the Hilbert space
  - An element (a quantum state) in the $n$-qubit space: $|v\rangle = \sum_{j=0}^{2^n-1} \alpha_j |j\rangle = (\alpha_0, \cdots, \alpha_{2^n-1})^T$
  - Measurement: we get $j$ with probability $|\alpha_j|^2$ (but destroy the superposition)

# Input model: vectors

Input model: state preparation oracle

$$O_u : |0\rangle \to |u\rangle = \sum_{j=0}^{2^n-1} u_j |j\rangle$$

Constructing $O_u$ is generally hard, but easy in special cases[1]

[1] Grover-Rudolph [arXiv:quant-ph/0208112], Zhang-Li-Yuan [arXiv:2201.11495]

# Input model: matrices

### Definition (Block-encoding)

Let $A$ be a $2^n$-by-$2^n$ matrix. A block-encoding of $A$ is a $2^{n+a}$-by-$2^{n+a}$ unitary $U_A$ such that

$$A \approx \alpha \left( \langle 0|^{\otimes a} \otimes I \right) U_A \left( |0\rangle^{\otimes a} \otimes I \right),$$

or equivalently

$$U_A \approx \begin{pmatrix} \frac{1}{\alpha} A & * \\ * & * \end{pmatrix}.$$

- $\alpha$ is called the block-encoding factor and should satisfy $\alpha \geq \|A\|$

# Input model: matrices

$$U_A \approx \begin{pmatrix} \frac{1}{\alpha}A & * \\ * & * \end{pmatrix}.$$

- For a arbitrarily given matrix $A$, constructing $U_A$ is in general hard
- Special cases: unitary, sparse matrices, structured matrices, $\cdots$[2]

[2] Gilyen Etal [arXiv:1806.01838], Camps Etal [arXiv:2203.10236]

# Matrix-vector multiplication

## Input:

Block-encoding of A:

$$A \approx \alpha \left( \langle 0| \otimes I \right) U_A \left( |0\rangle \otimes I \right)$$

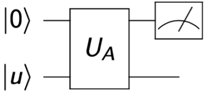
$$U_A \approx \begin{array}{cc} \begin{array}{cc} \langle 0| & \langle 1| \end{array} & \\ \begin{pmatrix} \frac{1}{\alpha}A & * \\ * & * \end{pmatrix} & \begin{array}{c} |0\rangle \\ |1\rangle \end{array} \end{array}$$

or $$U_A \approx |0\rangle \langle 0| \otimes \frac{A}{\alpha} + |0\rangle \langle 1| \otimes * + |1\rangle \langle 0| \otimes * + |1\rangle \langle 1| \otimes *$$

Quantum state:

$$|u\rangle = \sum_{j=0}^{2^n-1} u_j |j\rangle$$

## 'Algorithm': applying block-encoding




or $$U_A |0\rangle |u\rangle \approx \frac{1}{\alpha} |0\rangle A |u\rangle + c |1\rangle |*\rangle$$

or $$\begin{pmatrix} \frac{1}{\alpha}A & * \\ * & * \end{pmatrix} \begin{pmatrix} u \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha}Au \\ * \end{pmatrix}$$

Need to measure the first ancilla qubit

Success probability: $(\| A |u\rangle \| / \alpha)^2$

Number of repeats (after amplitude amplification): $\mathcal{O}\left( \alpha / \| A |u\rangle \| \right)$

# Matrix addition: Linear combination of unitaries (LCU)

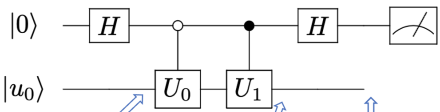
Task of LCU[3]: given a set of unitary operators $U_j$ and coefficients $c_j$, compute

$$\sum_j c_j U_j$$

[3] Childs-Wiebe [arXiv:1202.5822], Childs-Kothari-Somma [arXiv:1511.02306]

# Matrix addition: LCU

A toy example: computing $\frac{1}{2}(U_0 + U_1)\,|u_0\rangle$



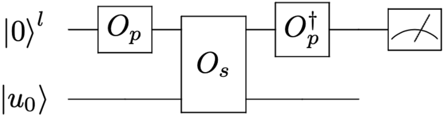

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|u_0\rangle$$

$$\frac{1}{\sqrt{2}}(|0\rangle\, U_0\, |u_0\rangle + |1\rangle\, U_1\, |u_0\rangle)$$

$$\frac{1}{2}\,|0\rangle\,(U_0 + U_1)\,|u_0\rangle + \frac{1}{2}\,|1\rangle\,(U_0 - U_1)\,|u_0\rangle)$$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \begin{aligned} |0\rangle &\to \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ |1\rangle &\to \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \end{aligned}$$

General: computing $\sum_j c_j U_j$




Prepare Oracle $O_p : |0\rangle \to \frac{1}{\sqrt{\|c\|_1}} \sum_j \sqrt{c_j}\,|j\rangle$

Select Oracle $O_s = \sum_j |j\rangle\langle j| \otimes U_j$

# Matrix addition: LCU

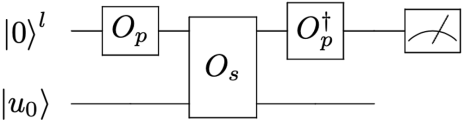
$$|0\rangle |u_0\rangle \xrightarrow{O_p} \frac{1}{\sqrt{\|c\|_1}} \sum_j \sqrt{c_j} |j\rangle |u_0\rangle$$

$$\xrightarrow{O_s} \frac{1}{\sqrt{\|c\|_1}} \sum_j \sqrt{c_j} |j\rangle U_j |u_0\rangle$$

$$\xrightarrow{O_p^\dagger} \frac{1}{\|c\|_1} |0\rangle \sum_j c_j U_j |u_0\rangle + |\perp\rangle$$

- Repeats: $\mathcal{O}(\|c\|_1 / \| \sum c_j U_j |u_0\rangle \|)$
- Overall complexity depends on the cost of constructing $O_p$ and $O_s$

General: computing $\sum_j c_j U_j$



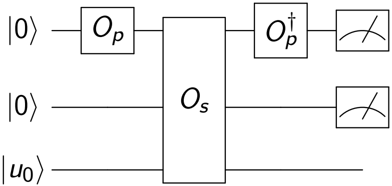

Prepare Oracle $O_p : |0\rangle \to \frac{1}{\sqrt{\|c\|_1}} \sum_j \sqrt{c_j} |j\rangle$

Select Oracle $O_s = \sum_j |j\rangle \langle j| \otimes U_j$

# Matrix addition

- Goal: compute (block encode) $\sum_j c_j A_j$ for general matrices $\|A_j\| \leq 1$
- Algorithm: LCU where unitaries are block-encodings

$$O_s = \sum_j |j\rangle\langle j| \otimes U_{A_j}, \qquad U_{A_j} = \begin{pmatrix} A_j & * \\ * & * \end{pmatrix}$$



# Vector addition

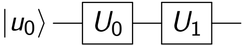
- Goal: compute $\sum_j c_j |u_j\rangle$ for quantum states $|u_j\rangle$
- Algorithm: Implement $\sum_j c_j U_j$ on $|0\rangle$ where $U_j$'s are state preparation oracles

$$U_j : |0\rangle \to |u_j\rangle\,, \qquad \sum_j c_j U_j |0\rangle = \sum_j c_j |u_j\rangle$$

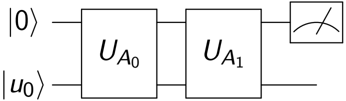
# Matrix multiplication

Let us start with two matrices

- Two unitaries $U_1 U_0$




- Two matrices: $A_0$ and $A_1$, block-encodings $U_{A_0}$ and $U_{A_1}$. Can we try this?




??

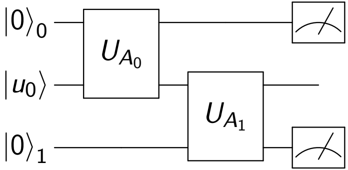
- Does not work:

$$U_{A_1} U_{A_0} |0\rangle |u_0\rangle = U_{A_1} (|0\rangle A_0 |u_0\rangle + |1\rangle |*\rangle)$$

$$\begin{pmatrix} A_1 & * \\ * & * \end{pmatrix} \begin{pmatrix} A_0 & * \\ * & * \end{pmatrix} \begin{pmatrix} u_0 \\ 0 \end{pmatrix} = \begin{pmatrix} A_1 & * \\ * & * \end{pmatrix} \begin{pmatrix} A_0 u_0 \\ * \end{pmatrix}$$

# Matrix multiplication

Method 1: duplicate ancilla qubits

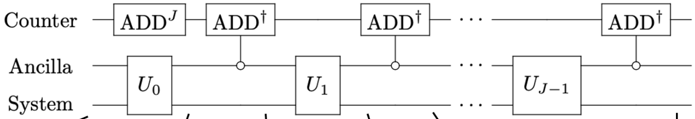



$$
\begin{aligned}
|0\rangle_1 |0\rangle_0 |u_0\rangle &\xrightarrow{U_{A_0}} |0\rangle_1 (|0\rangle_0 A_0 |u_0\rangle + |1\rangle_0 |*\rangle) \\
&= |0\rangle_0 |0\rangle_1 A_0 |u_0\rangle + |1\rangle_0 |0\rangle_1 |*\rangle \\
&\xrightarrow{U_{A_1}} |0\rangle_0 (|0\rangle_1 A_1 A_0 |u_0\rangle + |1\rangle_1 |*\rangle) + |1\rangle_0 |0\rangle_1 |*\rangle + |1\rangle_0 |1\rangle_1 |*\rangle
\end{aligned}
$$

Multiplication of $J$ matrices: using $\mathcal{O}(J)$ extra ancilla qubits

# Matrix multiplication

Method 2: compression gadget (Low-Wiebe [arXiv:1805.00675], Fang-Lin-Tong [arXiv:2208.06941])




$|0\rangle\,|0\rangle\,|u_0\rangle$

$|J\rangle\,|0\rangle\,|u_0\rangle$

$|J\rangle\,(|0\rangle\,A_0\,|u_0\rangle + |1\rangle\,|*\rangle)$

$|J-1\rangle\,|0\rangle\,A_0\,|u_0\rangle + |J\rangle\,|1\rangle\,|*\rangle$

$|J-1\rangle\,(|0\rangle\,A_1A_0\,|u_0\rangle + |1\rangle\,|*\rangle)$
$+\,|J\rangle\,(|0\rangle\,|*\rangle + |1\rangle\,|*\rangle)$

$|J-2\rangle\,|0\rangle\,A_1A_0\,|u_0\rangle + |J-1\rangle\,|1\rangle\,|*\rangle$
$+\,|J-1\rangle\,|0\rangle\,|*\rangle + |J\rangle\,|1\rangle\,|*\rangle$

$|0\rangle\,|0\rangle\,A_{J-1}\cdots A_1A_0\,|u_0\rangle + \sum_{j>0}|j\rangle\,|*\rangle\,|*\rangle$

$$\mathrm{ADD}: |0\rangle \to |1\rangle \to |2\rangle \to \cdots \to |J\rangle \to |0\rangle$$

# Summary: basic linear algebra operations

- Input models: quantum state, block-encoding
- Matrix-vector multiplication: applying block-encoding
- Matrix/vector addition: LCU
- Matrix multiplication

# Linear systems of equations

# Quantum linear system problem (QLSP)

- Classical: given $A$: $N \times N$ Hermitian matrix, $b$: $N$-dimensional vector, compute

$$x = A^{-1} b$$

  - Non-Hermitian: consider $\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$
- Quantum: find an $\epsilon$-approximation of the quantum state

$$|x\rangle = \frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}$$

  - Assume $\|A\| = 1$ and we have some black-box access to $A$ and $|b\rangle$ (e.g., block-encoding and state preparation oracle)
- Important parameters: dimension $N$, tolerated error level $\epsilon$, condition number $\kappa = \|A\|\|A^{-1}\|$

# Harrow-Hassidim-Lloyd (HHL)

- [4]The first quantum algorithm for solving QLSP
- Key equation: let $(\lambda_j, |v_j\rangle)$ be the eigenvalues and eigenvectors of $A$, and $|b\rangle = \sum_{j=0}^{N-1} \beta_j |v_j\rangle$, then

$$A^{-1}|b\rangle = \left(\sum_{j=0}^{N-1} \lambda_j^{-1} |v_j\rangle \langle v_j|\right) \left(\sum_{j=0}^{N-1} \beta_j |v_j\rangle\right) = \sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j} |v_j\rangle$$

- Need to do:
  - store the information (binary encoding) of $\lambda_j$'s in an ancilla register coherently
  - multiply the factor $\lambda_j^{-1}$ to each eigenvector $|v_j\rangle$

[4]Harrow-Hassidim-Lloyd [arXiv:0811.3171]

# HHL

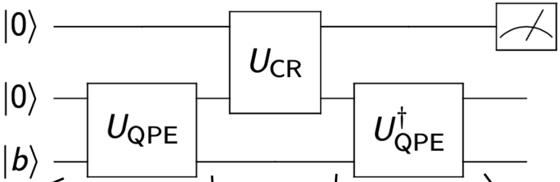
- Need to do:
  - store the information (binary encoding) of $\lambda_j$'s in an ancilla register coherently
  - multiply the factor $\lambda_j^{-1}$ to each eigenvector $|v_j\rangle$
- Useful subroutines:
  - Quantum phase estimation (QPE): Let $U$ be a unitary and $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$ for a real number $\theta \in [0,1]$. The (ideal) QPE algorithm is $U_{\text{QPE}}$ such that

$$U_{\text{QPE}}|\psi\rangle|0\rangle = |\psi\rangle|\theta\rangle .$$

    Here $|\theta\rangle = |\theta_{m-1}\rangle \cdots |\theta_1\rangle |\theta_0\rangle$ where $\theta = (.\theta_0\theta_1\cdots\theta_{m-1})$ is its binary representation
  - Controlled rotation: A unitary $U_{\text{CR}}$ such that

$$U_{\text{CR}}|\theta\rangle|0\rangle = |\theta\rangle\left(f(\theta)|0\rangle + \sqrt{1-|f(\theta)|^2}\,|1\rangle\right)$$

# HHL




$$\sum_j \beta_j |0\rangle |0\rangle |v_j\rangle$$

QPE for $e^{2\pi i A}$

$$\sum_j \beta_j |0\rangle |\lambda_j\rangle |v_j\rangle$$

$$\sum_j C \frac{\beta_j}{\lambda_j} |0\rangle |0\rangle |v_j\rangle + \sum_j * |1\rangle |0\rangle |v_j\rangle$$

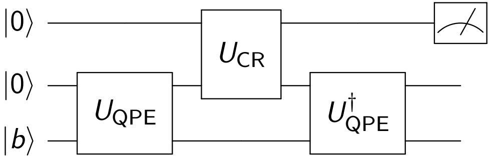
$$C \sim 1/\kappa$$

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

$$= \sum_j C \frac{\beta_j}{\lambda_j} |0\rangle |\lambda_j\rangle |v_j\rangle + \sum_j * |1\rangle |\lambda_j\rangle |v_j\rangle$$

# HHL

- Complexity analysis: two sources
  - The cost of a single run: mainly due to QPE
$$\mathcal{O}(\kappa/\epsilon)$$
  - Number of repeats:
$$\mathcal{O}(1/(C\|A^{-1}|b\rangle\|)) = \mathcal{O}(\kappa)$$
- Overall complexity:
$$\mathcal{O}(\kappa^2/\epsilon)$$
- Need two efficient subroutines:
  - Hamiltonian simulation: implement $e^{2\pi i A}$
  - QPE

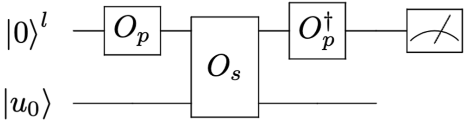



Output (here $C \sim 1/\kappa$):

$$\sum_j C|0\rangle|0\rangle(A^{-1}|b\rangle) + |\perp\rangle$$

# LCU

General: computing $\sum_j c_j U_j$




Prepare Oracle $O_p : |0\rangle \to \frac{1}{\sqrt{\|c\|_1}} \sum_j \sqrt{c_j} |j\rangle$

Select Oracle $O_s = \sum_j |j\rangle \langle j| \otimes U_j$

- Review of LCU:
  - Output: $\frac{1}{\|c\|_1} |0\rangle \sum_j c_j U_j |u_0\rangle + |\perp\rangle$
  - Cost of each run: depend on $O_s$
  - Repeats: $\mathcal{O}(\|c\|_1 / \| \sum c_j U_j |u_0\rangle \|)$
- Idea for QLSP: decompose $A^{-1}$ as linear combination of unitaries

[4] Childs-Kothari-Somma [arXiv:1511.02306]

# LCU: Fourier approach

- Key identity:

$$\frac{1}{x} = \frac{i}{\sqrt{2\pi}} \int_0^\infty dy \int_{-\infty}^\infty dz z e^{-z^2/2} e^{-ixyz}.$$

- For a Hermitian matrix $A$,

$$\begin{aligned} A^{-1} &= \frac{i}{\sqrt{2\pi}} \int_0^\infty dy \int_{-\infty}^\infty dz z e^{-z^2/2} e^{-iyzA} \\ &\approx \frac{i}{\sqrt{2\pi}} \int_0^Y dy \int_{-Z}^Z dz z e^{-z^2/2} e^{-iyzA} \\ &\approx \sum c_{j,j'} e^{-iy_j z_{j'} A} \end{aligned}$$

- $\epsilon$-approximation if $Y = \mathcal{O}(\kappa\sqrt{\log(\kappa/\epsilon)})$, $Z = \mathcal{O}(\sqrt{\log(\kappa/\epsilon)})$
- Cost of Hamiltonian simulation for $e^{-iHT}$: $\mathcal{O}(T \operatorname{poly}\log(1/\epsilon))$
- Overall complexity

$$\kappa \operatorname{poly}\log(\kappa/\epsilon) \times \kappa\sqrt{\log(\kappa/\epsilon)} = \mathcal{O}(\kappa^2 \operatorname{poly}\log(\kappa/\epsilon))$$

# LCU: Chebyshev approach

- Idea: expand $1/x$ using Chebyshev polynomials
  - Chebyshev polynomials:

$$T_n(\cos(\theta)) = \cos(n\theta)$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad T_0(x) = 1,\, T_1(x) = x$$

  - Bounded by 1 on $[-1,1]$, minimize Runge's phenomenon, close to the best polynomial approximation
- Approach:

$$\begin{aligned}
\frac{1}{x} &\approx \frac{1-(1-x^2)^d}{x} && (d \sim \kappa^2 \log(\kappa/\epsilon)) \\
&= 4\sum_{j=0}^{d-1}(-1)^j \left(2^{-2d}\sum_{i=j+1}^{d}\binom{2d}{d+i}\right) T_{2j+1}(x) \\
&\approx 4\sum_{j=0}^{J}(-1)^j \left(2^{-2d}\sum_{i=j+1}^{d}\binom{2d}{d+i}\right) T_{2j+1}(x) && (J \sim \sqrt{d\log(d/\epsilon)})
\end{aligned}$$

# LCU: Chebyshev approach

$$A^{-1} \approx \sum_{j=0}^{\mathcal{O}(\kappa \text{ poly} \log(\kappa/\epsilon))} c_j T_{2j+1}(A)$$

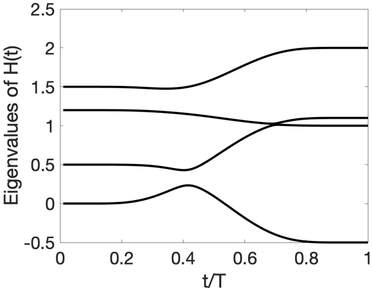
- $\|T_{2j+1}(A)\| \leq 1$ but not unitary, so we need to construct its block-encoding
- The same overall complexity:

$$\kappa \text{ poly} \log(\kappa/\epsilon) \times \kappa\sqrt{\log(\kappa/\epsilon)} = \mathcal{O}(\kappa^2 \text{poly} \log(\kappa/\epsilon))$$

# Adiabatic Quantum Computing (AQC)

$$\imath\partial_t |\psi(t)\rangle = H(t/T) |\psi(t)\rangle \,, \quad t \in [0, T]$$
$$H(0) |\psi(0)\rangle = \lambda_0 |\psi(0)\rangle$$

- Starting from the (easily prepared) eigenvector of $H(0)$, the wavefunction at the final time will approximate the corresponding eigenvector of $H(1)$ if
  - the Hamiltonian is slow enough (equivalently $T$ is large enough)
  - gap condition is satisfied
- Application: a quantum computing model to solve eigenvalue problem[a]




[a] Albash-Lidar [arXiv:1611.04471]

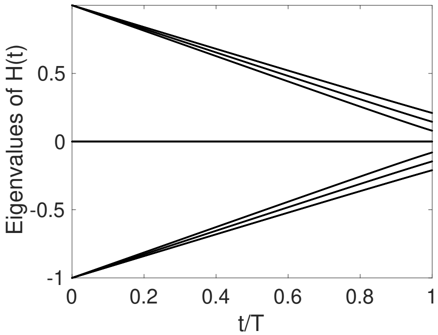
# AQC for QLSP

(Vanilla) AQC for QLSP algorithm:

$$H_0 = \begin{pmatrix} 0 & Q_b \\ Q_b & 0 \end{pmatrix}, \quad H_1 = \begin{pmatrix} 0 & AQ_b \\ Q_bA & 0 \end{pmatrix},$$

$$Q_b = I - |b\rangle \langle b|$$

$$H(s) = (1-s)H_0 + sH_1,$$




- Eigenpath corresponding to eigenvalue 0 is of interest, which connects $(b^\top, 0^\top)^\top$ and $(x^\top, 0^\top)^\top$ [5]

[5] Subasi-Somma-Orsucci [arXiv:1805.10549]

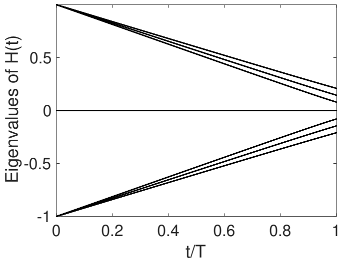
# Quantum Adiabatic Theorem

## Theorem (Jansen-Ruskai-Seiler (arXiv:quant-ph/0603175))

*Assume gap $\Delta(s)$, then the distance between the dynamics and the eigenvector can be bounded by*

$$\eta(s) = C\left\{\frac{\|H'(0)\|_2}{T\Delta^2(0)} + \frac{\|H'(s)\|_2}{T\Delta^2(s)} + \frac{1}{T}\int_0^s \left(\frac{\|H''(\tau)\|_2}{\Delta^2(\tau)} + \frac{\|H'(\tau)\|_2^2}{\Delta^3(\tau)}\right) d\tau\right\}.$$

- To bound the error by $\epsilon$: $T = \mathcal{O}(\Delta_*^{-3}\epsilon^{-1})$
- Cubic dependence on the gap
- In QLSP, $\Delta_* \sim 1/\kappa \Longrightarrow T = \mathcal{O}(\kappa^3\epsilon^{-1})$

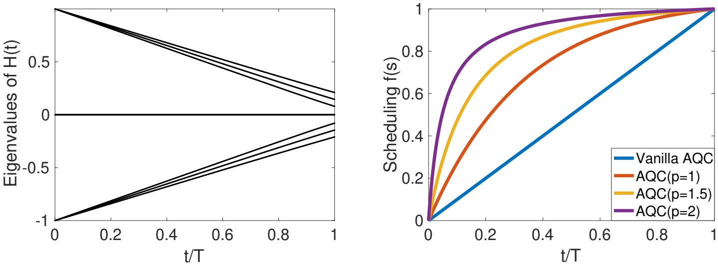


# Time-optimal AQC

$$\imath\partial_t |\psi(t)\rangle = H(t/T)|\psi(t)\rangle, \quad t \in [0, T]$$

- Idea: generally interpolate $H(s) = (1 - f(s))H_0 + f(s)H_1$, choose proper $f(s)$ to slow down the Hamiltonian when the gap is small
- AQC(p): $\dot{f}(s) = c\Delta^p(f(s))$

$$\implies T = \mathcal{O}(\kappa/\epsilon)$$ [6]




[6] An-Lin [arXiv:1909.05500]

# AQC(exp)

- Quantum adiabatic theorem can be improved to error $\sim \mathcal{O}(T^{-k})$ if we only care about the final state error[7]
- Requiring boundary cancellation condition, *i.e.*, the support of $H'(s)$ is in $(0,1)$

$$\eta(s) = C\left\{ \frac{\|H'(0)\|_2}{T\Delta^2(0)} + \frac{\|H'(s)\|_2}{T\Delta^2(s)} + \frac{1}{T}\int_0^s \left( \frac{\|H''(\tau)\|_2}{\Delta^2(\tau)} + \frac{\|H'(\tau)\|_2^2}{\Delta^3(\tau)} \right) d\tau \right\}.$$

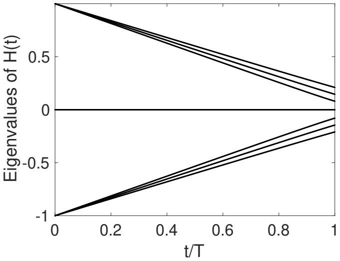
- Error $=$ Boundary$_1 + \frac{1}{T}\int_0^1 =$ Boundary$_1 +$ Boundary$_2 + \frac{1}{T^2}\int_0^1 =$ Boundary$_1 +$ Boundary$_2 +$ Boundary$_3 + \frac{1}{T^3}\int_0^1 = \cdots$

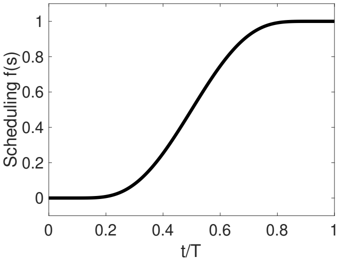
[7] Nenciu(1993)

# AQC(exp)

- AQC(exp): $f(s) = c^{-1}\int_0^s \exp[-u^{-1}(1-u)^{-1}]du$, happens to be slow as well at the smallest gap

$$\implies T = \mathcal{O}\left(\kappa \operatorname{poly}\log(\kappa/\epsilon)\right) \quad [8]$$




[8] An-Lin [arXiv:1909.05500]

# Preconditioning

- QLSP algorithms with cost $\mathcal{O}(\kappa \operatorname{poly}\log(\kappa/\epsilon))$, can still be expensive if the system is ill-conditioned
- Classical solution: preconditioning

$$Ax = b \iff MAx = Mb$$

- Effective if
  - $\kappa(MA) \ll \kappa(A)$
  - the matrix-vector multiplication $My$ is easily accessible, and in particular its cost is independent of $\kappa(M)$
- Classical: diagonal matrix, incomplete factorization, sparse approximate inverse (SPAI), etc.
- Quantum:
  - SPAI (Clader-Jacobs-Sprouse [arXiv:1301.2340])
  - Circulant matrix (Shao-Xiang [arXiv:1807.04563])
  - Diagonal matrix (Tong-An-Wiebe-Lin [arXiv:2008.13295])

# Preconditioning

$$(A + B)\,|x\rangle \sim |b\rangle$$

- Assume $A$ is easily invertible with very large $\|A\|$ and moderate $\|B\|, \|A^{-1}\|, \|(A+B)^{-1}\|$
- $\kappa(A+B) \sim \mathcal{O}(\|A\|)$
- An example: Poisson's equation $-\Delta u(r) + V(r)u(r) = b(r)$
- Preconditioner: $A^{-1}$
  - $\kappa(I + A^{-1}B) = \mathcal{O}(1)$
- Algorithm: $A^{-1} \to A^{-1}B \to I + A^{-1}B \to (I + A^{-1}B)^{-1} \to (I + A^{-1}B)^{-1}A^{-1} = (A+B)^{-1}$
  - Need matrix addition and multiplication, and fast-inversion of a diagonal matrix

# Summary: QLSP

- HHL algorithm:
  - QPE and Hamiltonian simulation
  - $\mathcal{O}(\kappa^2/\epsilon) \xrightarrow{\textit{improvable}} \mathcal{O}(\kappa/\epsilon^3)$
- LCU:
  - Polynomial approximation of $1/x$
  - $\mathcal{O}(\kappa^2 \operatorname{poly}\log(\kappa/\epsilon)) \xrightarrow{\textit{improvable}} \mathcal{O}(\kappa \operatorname{poly}\log(\kappa/\epsilon))$
- AQC:
  - QLSP as an eigenvalue/eigenvector problem
  - $\mathcal{O}(\kappa \operatorname{poly}\log(\kappa/\epsilon)) \xrightarrow{\textit{improvable}} \mathcal{O}(\kappa \log(1/\epsilon))$
- Lower bound[9]: $\Omega(\kappa \log(1/\epsilon))$
- Preconditioning

[9] Harrow-Kothari (in preparation)

# Eigenvalue problems

# Eigenvalue problems

- We discussed AQC approach, and "discussed" QPE
- Ground state/energy problem
- General optimization: variational quantum eigensolvers, a hybrid quantum-classical approach

# Matrix functions

# Matrix functions

- For Hermitian matrices: eigenvalue transformation

$$A = V\mathrm{diag}(\lambda_j)V^\dagger \quad \xrightarrow{\text{or}} \quad f(A) = V\mathrm{diag}(f(\lambda_j))V^\dagger$$

- For general matrices: singular value transformation

$$\begin{aligned} A = W\mathrm{diag}(\sigma_j)V^\dagger \quad &\rightarrow \quad f(A) = W\mathrm{diag}(f(\sigma_j))V^\dagger \\ &\xrightarrow{\text{or}} \quad f(A) = V\mathrm{diag}(f(\sigma_j))V^\dagger \\ &\xrightarrow{\text{or}} \quad f(A) = W\mathrm{diag}(f(\sigma_j))W^\dagger \end{aligned}$$

# Main result

Suppose that $U_A$ is the block-encoding of a Hermitian matrix $A$ with $\|A\| \leq 1$, and $p(x)$ is a real-coefficient polynomial such that

1. degree of $p(x)$ is $d$,
2. $|p(x)| \leq 1$ for all $x \in [-1, 1]$.

Then, $p(A)$ can be block-encoded with complexity

$$\mathcal{O}(d)$$

[9] Gilyen Etal [arXiv:1806.01838]

# Applications

- Solving linear systems of equations: $f(x) = \frac{1}{\kappa x}$
- Hamiltonian simulation: $f(x) = e^{-iAt}$
- Filtering[10]
- Amplitude amplification:

$$U|0\rangle|\psi\rangle = \frac{1}{q}|0\rangle A|\psi\rangle + |\perp\rangle \quad \rightarrow \quad \widetilde{U}|0\rangle|\psi\rangle = \frac{1}{2}|0\rangle A|\psi\rangle + |\perp\rangle$$

$$f(x) = qx/2, \quad x \in [-1/q, 1/q]$$

$$f(x) \approx p(x) \quad \text{where} \quad \deg(p(x)) \sim q\log(1/\epsilon)$$

- ......

[10] Lin-Tong [arXiv:1910.14596]

# Toy example: Chebyshev polynomials

Consider a 2-by-2 matrix

$$O = \begin{pmatrix} \lambda & -\sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & \lambda \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Then

$$O^k = \begin{pmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{pmatrix} = \begin{pmatrix} T_n(\lambda) & * \\ * & * \end{pmatrix}$$

where $T_k = \cos(k\theta)$ is the Chebyshev polynomial.

- For Hermitian matrix case $T_k(A) = VT_k(\Lambda)V^\dagger$: for each eigenvalue, find its corresponding 2-dimensional subspace and perform this $O^k$

# Qubitization

Suppose $A = \sum \lambda_j |v_j\rangle \langle v_j|$ and $U_A$ is its *Hermitian* block-encoding ($U_A = U_A^\dagger$).

$$U_A |0\rangle |v_j\rangle = |0\rangle A |v_j\rangle + * = \lambda_j |0\rangle |v_j\rangle + \sqrt{1-\lambda_j^2} |\perp_j\rangle$$

where $\Pi |\perp_j\rangle = 0$, $\Pi = |0\rangle \langle 0| \otimes I$.
Apply $U_A$ again yields

$$U_A^2 |0\rangle |v_j\rangle = \lambda_j(\lambda_j |0\rangle |v_j\rangle + \sqrt{1-\lambda_j^2} |\perp_j\rangle) + U_A \sqrt{1-\lambda_j^2} |\perp_j\rangle$$
$$U_A |\perp_j\rangle = \sqrt{1-\lambda^2} |0\rangle |v_j\rangle - \lambda_j |\perp_j\rangle .$$

Invariant space: $\mathcal{H}_j = \operatorname{span}\{|0\rangle |v_j\rangle, |\perp_j\rangle\}$. We may write

$$[U_A]_{\mathcal{H}_j} = \begin{pmatrix} \lambda_j & \sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & -\lambda_j \end{pmatrix}, \quad [\Pi]_{\mathcal{H}_j} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

[10] Low-Chuang [arXiv:1610.06546]

# Qubitization

$$[U_A]_{\mathcal{H}_j} = \begin{pmatrix} \lambda_j & \sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & -\lambda_j \end{pmatrix}, \quad [\Pi]_{\mathcal{H}_j} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Let

$$Z_\Pi = 2\Pi - 1, \quad [Z_\Pi]_{\mathcal{H}_j} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Then

$$O = U_A Z_\Pi, \quad [O]_{\mathcal{H}_j} = \begin{pmatrix} \lambda_j & -\sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & \lambda_j \end{pmatrix},$$

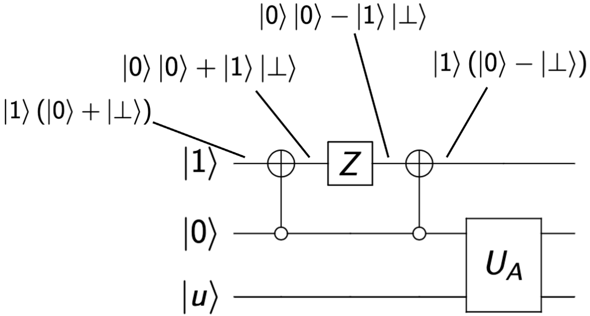
and thus

$$[O^k]_{\mathcal{H}_j} = \begin{pmatrix} T_k(\lambda_j) & * \\ * & * \end{pmatrix}, \quad O^k = \begin{pmatrix} T_k(A) & * \\ * & * \end{pmatrix}$$

# Qubitization

$$U_A = \begin{pmatrix} A & * \\ * & * \end{pmatrix}, \quad U_A = U_A^\dagger, \quad \Pi = |0\rangle\langle 0| \otimes I, \quad Z_\Pi = 2\Pi - 1$$

$$(U_A Z_\Pi)^k = \begin{pmatrix} T_k(A) & * \\ * & * \end{pmatrix}$$




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

# Qubitization

So far we have assumed *Hermitian* block-encoding for Hermitian matrices (i.e., $U_A = U_A^\dagger$), now we relax this assumption

$$U_A \left|0\right\rangle \left|v_j\right\rangle = \lambda_j \left|0\right\rangle \left|v_j\right\rangle + \sqrt{1-\lambda_j^2} \left|\perp_j'\right\rangle$$

where $\Pi \left|\perp_j'\right\rangle = 0$, $\Pi = \left|0\right\rangle \left\langle 0\right| \otimes I$.
Notice that since $A$ is Hermitian,

$$U_A^\dagger = \begin{pmatrix} A & * \\ * & * \end{pmatrix}, \quad U_A^\dagger \left|0\right\rangle \left|v_j\right\rangle = \lambda_j \left|0\right\rangle \left|v_j\right\rangle + \sqrt{1-\lambda_j^2} \left|\perp_j\right\rangle$$

where $\Pi \left|\perp_j\right\rangle = 0$. Apply $U_A$, then we have

$$\left|0\right\rangle \left|v_j\right\rangle = \lambda_j(\lambda_j \left|0\right\rangle \left|v_j\right\rangle + \sqrt{1-\lambda_j^2} \left|\perp_j'\right\rangle) + \sqrt{1-\lambda_j^2} U_A \left|\perp_j\right\rangle$$

$$U_A \left|\perp_j\right\rangle = \sqrt{1-\lambda_j^2} \left|0\right\rangle \left|v_j\right\rangle - \lambda_j \left|\perp_j'\right\rangle$$

# Qubitization

$$U_A |0\rangle |v_j\rangle = \lambda_j |0\rangle |v_j\rangle + \sqrt{1-\lambda_j^2} |\perp_j'\rangle$$
$$U_A |\perp_j\rangle = \sqrt{1-\lambda_j^2} |0\rangle |v_j\rangle - \lambda_j |\perp_j'\rangle$$

So $U_A$ maps $\mathcal{H}_j = \text{span}\{|0\rangle |v_j\rangle, |\perp_j\rangle\}$ to $\mathcal{H}_j' = \text{span}\left\{|0\rangle |v_j\rangle, |\perp_j'\rangle\right\}$, we can also verify that $U_A^\dagger$ maps $\mathcal{H}_j'$ to $\mathcal{H}_j$,

$$[U_A]_{\mathcal{H}_j \to \mathcal{H}_j'} = \begin{pmatrix} \lambda_j & \sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & -\lambda_j \end{pmatrix}, \quad [U_A^\dagger]_{\mathcal{H}_j' \to \mathcal{H}_j} = \begin{pmatrix} \lambda_j & \sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & -\lambda_j \end{pmatrix}$$

For the projector $\Pi = |0\rangle\langle 0| \otimes I$, $Z_\Pi = 2\Pi - 1$,

$$[Z_\Pi]_{\mathcal{H}_j} = [Z_\Pi]_{\mathcal{H}_j'} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

# Qubitization

Let $\mathcal{H}_j = \text{span}\{|0\rangle|v_j\rangle, |\perp_j\rangle\}$, $\mathcal{H}'_j = \text{span}\left\{|0\rangle|v_j\rangle, |\perp'_j\rangle\right\}$,

$$[U_A]_{\mathcal{H}_j \to \mathcal{H}'_j} = [U_A^\dagger]_{\mathcal{H}'_j \to \mathcal{H}_j} = \begin{pmatrix} \lambda_j & \sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & -\lambda_j \end{pmatrix}, \quad [Z_\Pi]_{\mathcal{H}_j} = [Z_\Pi]_{\mathcal{H}'_j} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Then

$$[U_A^\dagger Z_\Pi U_A Z_\Pi]_{\mathcal{H}_j} = \begin{pmatrix} \lambda_j & \sqrt{1-\lambda_j^2} \\ \sqrt{1-\lambda_j^2} & -\lambda_j \end{pmatrix}^2, \quad [(U_A^\dagger Z_\Pi U_A Z_\Pi)^k]_{\mathcal{H}_j} = \begin{pmatrix} T_{2k}(\lambda_j) & * \\ * & * \end{pmatrix}.$$

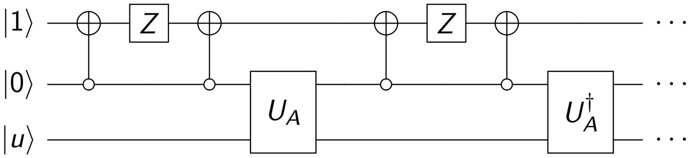
Therefore $(U_A^\dagger Z_\Pi U_A Z_\Pi)^k$ block encodes $T_{2k}(A)$. For odd polynomials, notice that $\mathcal{H}_j$ and $\mathcal{H}'_j$ share common $|0\rangle|v_j\rangle$,

$$[U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k]_{\mathcal{H}_j \to \mathcal{H}'_j} = \begin{pmatrix} T_{2k+1}(\lambda_j) & * \\ * & * \end{pmatrix},$$

$$U_A Z_\Pi (U_A^\dagger Z_\Pi U_A Z_\Pi)^k = \begin{pmatrix} T_{2k+1}(A) & * \\ * & * \end{pmatrix}$$

# Qubitization

$$(U_A^\dagger Z_\Pi U_A Z_\Pi)^k$$




Now we can implement any polynomial by LCU, but may introduce extra overhead and control logic

# Quantum signal processing (QSP)

Let us start with the 2-by-2 matrix again

$$U = \begin{pmatrix} \lambda & \sqrt{1-\lambda^2} \\ \sqrt{1-\lambda^2} & -\lambda \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

we have shown that

$$(UZ)^k = \begin{pmatrix} T_k(\lambda) & * \\ * & * \end{pmatrix}.$$

Notice that

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = -i \begin{pmatrix} e^{i\frac{\pi}{2}} & 0 \\ 0 & e^{-i\frac{\pi}{2}} \end{pmatrix} = -ie^{i\frac{\pi}{2}Z}$$

What if we consider a more general

$$e^{i\phi_d Z} U e^{i\phi_{d-1} Z} \dots U e^{i\phi_2 Z} U e^{i\phi_1 Z} U e^{i\phi_0 Z}$$

where $(\phi_0, \phi_1, \cdots, \phi_d) \in \mathbb{R}^{d+1}$.

# Quantum signal processing (QSP)

## Theorem (QSP)

*Let*

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

*Then there exist phase factors* $(\phi_0, \phi_1, \cdots, \phi_d) \in \mathbb{R}^{d+1}$ *such that*

$$e^{i\phi_d Z} U e^{i\phi_{d-1} Z} \ldots U e^{i\phi_2 Z} U e^{i\phi_1 Z} U e^{i\phi_0 Z} = \begin{pmatrix} p(\lambda) & -q(\lambda)\sqrt{1-\lambda^2} \\ q^*(\lambda)\sqrt{1-\lambda^2} & p^*(\lambda) \end{pmatrix}$$

*if and only if* $p(\lambda), q(\lambda)$ *are complex-coefficient polynomials such that*

1. $deg(p) \leq d$, $deg(q) \leq d-1$,
2. $p$ *has parity* $d \mod 2$ *and* $q$ *has parity* $d-1 \mod 2$,
3. $|p(\lambda)|^2 + (1-\lambda^2)|q(\lambda)|^2 = 1$ *for all* $\lambda \in [-1,1]$.

[10] Low-Chuang [arXiv:1606.02685]

# QSP

## Theorem (QSP for real polynomials)

*Let*

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

*Then there exist phase factors* $(\phi_0, \phi_1, \cdots, \phi_d) \in \mathbb{R}^{d+1}$ *such that*

$$e^{i\phi_d Z} U e^{i\phi_{d-1} Z} \cdots U e^{i\phi_2 Z} U e^{i\phi_1 Z} U e^{i\phi_0 Z} = \begin{pmatrix} P(\lambda) & * \\ * & * \end{pmatrix}$$

*if* $Re(P(\lambda)) = p(\lambda)$ *and*

1. $deg(p) \leq d$,
2. $p$ *has parity* $d \mod 2$,
3. $|p(\lambda)| \leq 1$ *for all* $\lambda \in [-1, 1]$.

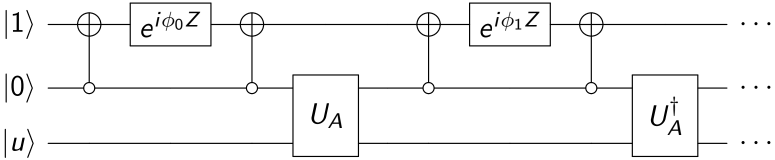
- The parity assumption can be further removed by $p = p_{\text{even}} + p_{\text{odd}}$ and LCU, or Motlagh-Wiebe [arXiv:2308.01501]

# QSP, qubitization, and QSVT

Through qubitization, for

1. any Hermitian matrix $A$ with $\|A\| \leq 1$ and its block-encoding $U_A$,
2. any $d$-degree real polynomial $p(\lambda)$ with $|p(\lambda)| \leq 1$ for all $\lambda \in [-1, 1]$,

we can block encode $p(A)$ with $\mathcal{O}(d)$ cost.




If $A$ is not Hermitian, we are performing singular value transformation[11].

[11] Gilyen Etal [arXiv:1806.01838]

# Phase factors

Finding phase factors was a hard task at the time when QSP was proposed, but has been practically solved so far.

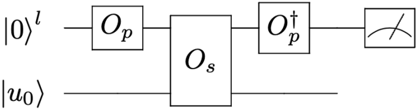
- Direct methods:
  - Remez exchange algorithm
  - roots of polynomials (Gilyen Etal [arXiv:1806.01838])
  - Capitalization (Chao Etal [arXiv:2003.02831])
  - Prony's method (Ying [arXiv:2202.02671])
- Iterative methods:
  - optimization based algorithm (Dong Etal [arXiv:2002.11649])
  - fixed point iteration (Dong Etal [arXiv:2209.10162])

# QSP/QSVT vs LCU

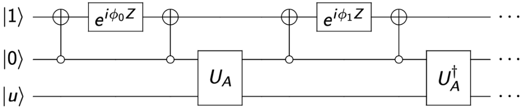
- Both QSP/QSVT and LCU can implement matrix functions
- For Hermitian matrices, LCU has computational overhead due to 1-norm of the coefficients and require extra control logic
- For general matrices
  - QSVT: singular value transformation
  - LCU: eigenvalue transformation

$$f(A) = \frac{1}{2\pi i} \int_\Gamma f(z)(z - A)^{-1} dz.$$

**LCU:**




**QSP:**



# Summary: matrix functions

- Qubitization for block-enccoding Chebyshev polynomials
- Quantum signal processing
- Quantum singular value transformation

# Summary

- Basic linear algebra operations
  - Input models for vectors and matrices: quantum state and block-encoding
  - Matrix-vector multiplication: applying block-encoding
  - Matrix/vector addition: linear combination of unitaries (LCU)
  - Matrix multiplication: compression gadget
- Linear systems of equations
  - General algorithms: HHL, LCU, AQC
  - Preconditioning
- Eigenvalue problems
- Matrix functions: Qubitization, QSP, QSVT