CS 59300 – Algorithms for Data Science

Classical and Quantum approaches

Lecture 18 (11/11)

Quantum linear algebra toolkits (III)

https://ruizhezhang.com/course_fall_2025.html

The slides are partly based on Dong An and Lin Lin's talk

Quantum linear algebra toolbox

- Basic linear algebra operations
- Linear systems of equations
- Matrix functions
  - Functions of Hermitian matrices: quantum signal processing (QSP), qubitization
  - Functions of general matrices: quantum singular value transformation (QSVT), linear combinations of Hamiltonian simulations (LCHS)

Matrix functions

- For Hermitian matrices: eigenvalue transformation

$$A = V\mathrm{diag}(\{\lambda_i\})V^\dagger \xrightarrow{f(x)} f(A) = V\mathrm{diag}(\{f(\lambda_i)\})V^\dagger$$

- For non-Hermitian matrices: singular value transformation

$$A = W\mathrm{diag}(\{\sigma_i\})V^\dagger \xrightarrow{f(x)} f(A) = W\mathrm{diag}(\{f(\sigma_i)\})V^\dagger$$

$$\xrightarrow{f(x)} f(A) = V\mathrm{diag}(\{f(\sigma_i)\})V^\dagger$$

$$\xrightarrow{f(x)} f(A) = W\mathrm{diag}(\{f(\sigma_i)\})W^\dagger$$

Matrix functions

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. $\deg(p) = d$
2. $|p(x)| \leq 1$ for all $x \in [-1,1]$

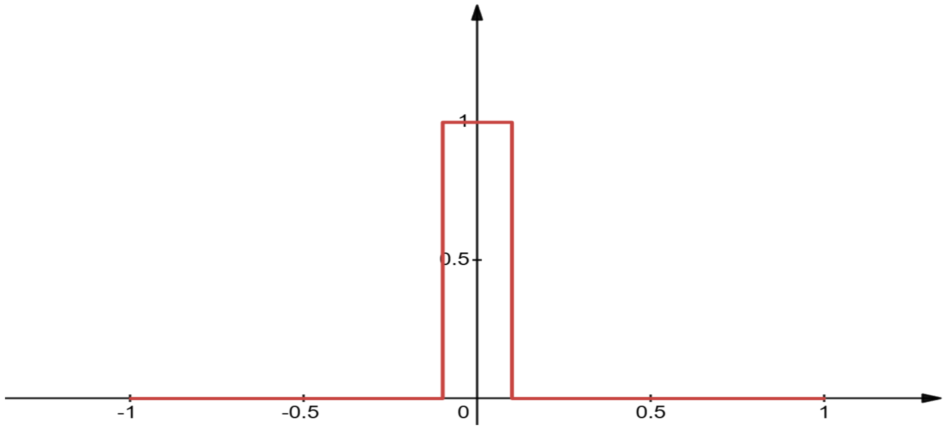
Then, $p(A)$ can be block-encoded with complexity $\mathcal{O}(d)$

Applications:

- QLSP: $f(x) = \frac{1}{\kappa x}$
- Hamiltonian simulation: $e^{-\mathbf{i}At} \implies f_1(x) = \cos(xt)$ and $f_2(x) = -\sin(xt)$, then by LCU

More applications

- Eigen-filtering:

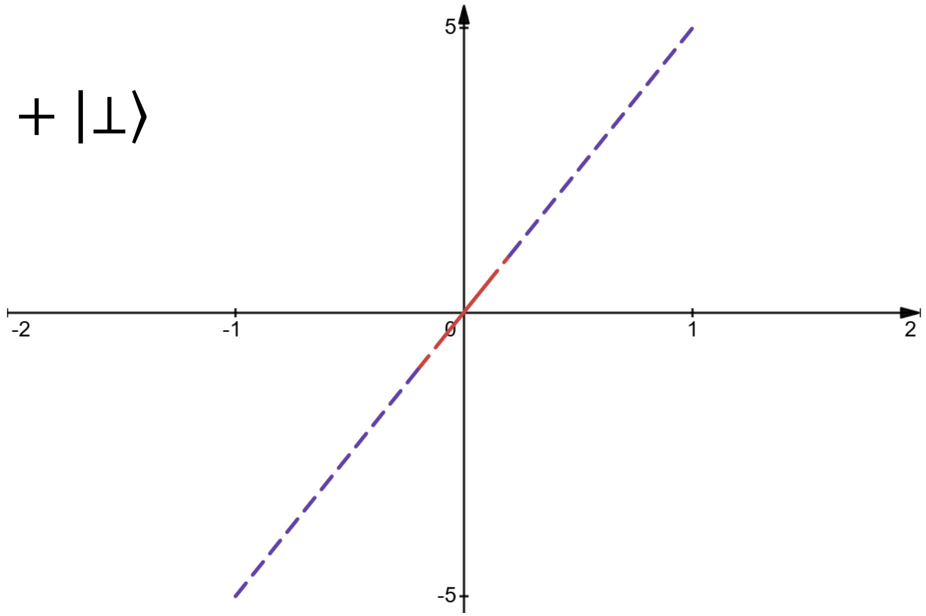


- Amplitude amplification:

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

$$f(x) = ax/2 \;\text{ for all } x \in [-1/a, 1/a]$$

$$f(x) \approx p(x) \;\text{ with }\; \deg(p) = \mathcal{O}(a \log(1/\epsilon))$$



Matrix functions

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. $\deg(p) = d$
2. $|p(x)| \leq 1$ for all $x \in [-1,1]$

Then, $p(A)$ can be block-encoded with complexity $\mathcal{O}(d)$

Chebyshev polynomial

For $A = \cos(\theta) \in \mathbb{C}^{1\times 1}$, how to block-encode $T_k(A) = T_k(\cos(\theta)) = \cos(k\theta)$?

- Consider the 2×2 matrix:

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

- Then, it is easy to check that

$$O^k = \begin{bmatrix} \cos(k\theta) & -\sin(k\theta) \\ \sin(k\theta) & \cos(k\theta) \end{bmatrix} = \begin{bmatrix} T_k(\lambda) & * \\ * & * \end{bmatrix},$$

  which is a $(1,1,0)$-block-encoding for $T_k(A)$

What about a general Hermitian matrix A?

- Qubitization (Low-Chuang '16): apply O^k to each 2-dimensional invariant subspace

Qubitization with Hermitian block-encoding

For $A = \sum \lambda_i |v_i\rangle\langle v_i|$, suppose U_A is its Hermitian block-encoding ($U_A = U_A^\dagger$)

- Apply U_A to an eigenstate $|0\rangle|v_i\rangle$:

$$U_A|0\rangle|v_i\rangle = |0\rangle A|v_i\rangle + |\perp\rangle = \lambda_i|0\rangle|v_i\rangle + \sqrt{1-\lambda_i^2}|\perp_i\rangle$$

- Apply U_A again:

$$|0\rangle|v_i\rangle = U_A^2|0\rangle|v_i\rangle = \lambda_i\left(\lambda_i|0\rangle|v_i\rangle + \sqrt{1-\lambda_i^2}|\perp_i\rangle\right) + \sqrt{1-\lambda_i^2}U_A|\perp_i\rangle$$

$$\Longrightarrow \quad U_A|\perp_i\rangle = \sqrt{1-\lambda_i^2}|0\rangle|v_i\rangle - \lambda_i|\perp_i\rangle$$

- $\mathcal{H}_i \coloneqq \mathrm{span}\{|0\rangle|v_i\rangle, |\perp_i\rangle\}$ is an invariant subspace

$$[U_A]_{\mathcal{H}_i} = \begin{bmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{bmatrix}$$

Qubitization with Hermitian block-encoding

$$[U_A]_{\mathcal{H}_i} = \begin{bmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{bmatrix}$$

- For the block-encoding projector $\Pi = |0\rangle\langle 0| \otimes I$, define

$$Z_\Pi := 2\Pi - I, \qquad [Z_\Pi]_{\mathcal{H}_j} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- For $O := U_A Z_\Pi$,

$$[O]_{\mathcal{H}_i} = \begin{bmatrix} \lambda_i & -\sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & +\lambda_i \end{bmatrix}$$

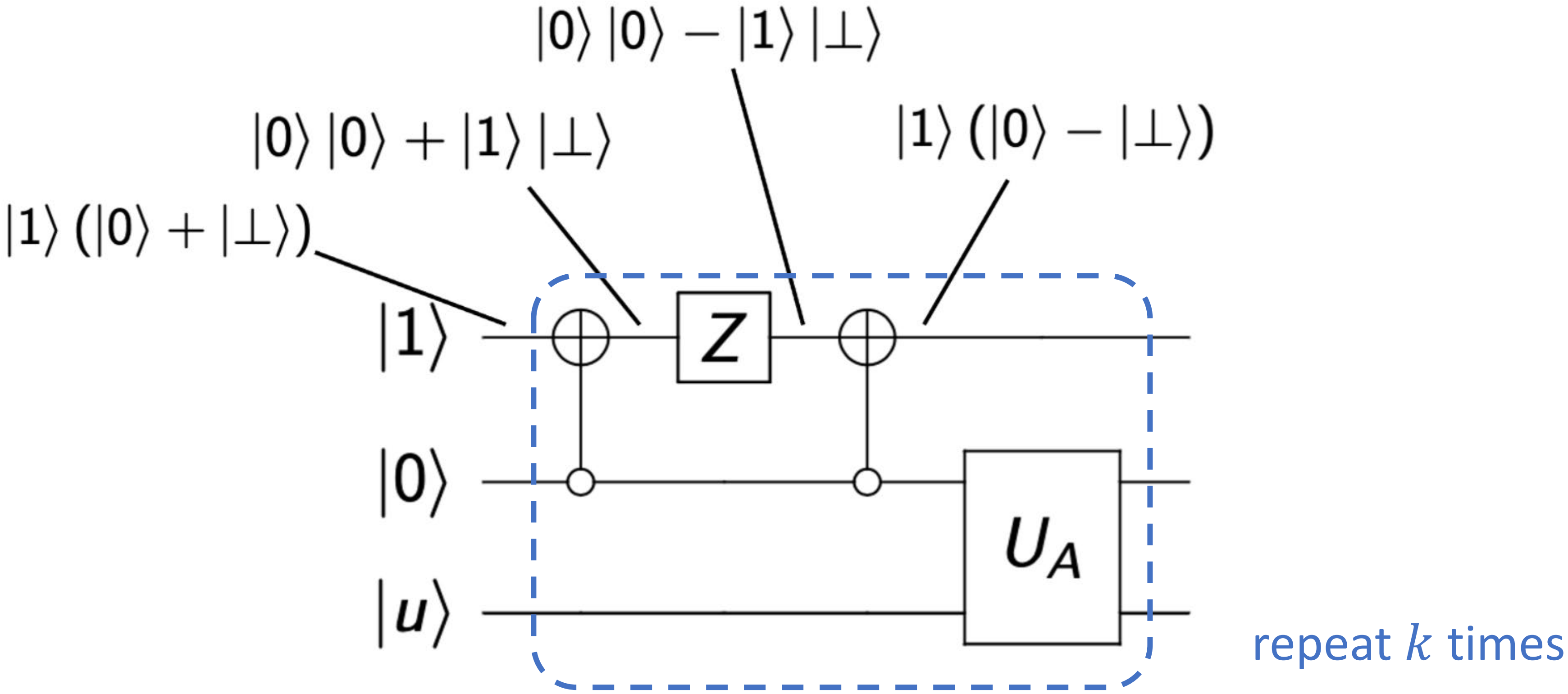
- Thus,

$$[O^k]_{\mathcal{H}_i} = \begin{bmatrix} T_k(\lambda_i) & * \\ * & * \end{bmatrix}, \qquad O^k = \begin{bmatrix} T_k(A) & * \\ * & * \end{bmatrix}$$

$$\mathbb{C}^{N+1} = \mathcal{H}_0 \oplus \cdots \oplus \mathcal{H}_{N-1}$$

Qubitization with Hermitian block-encoding

- $U_A = U_A^\dagger,\ \Pi = |0\rangle\langle 0| \otimes I,\ Z_\Pi = 2\Pi - I$
- $U_{T_k(A)} = (U_A Z_\Pi)^k$



Qubitization

For $A = \sum \lambda_i |v_i\rangle\langle v_i|$, suppose U_A is its block-encoding but non-Hermitian

- Apply U_A to an eigenstate $|0\rangle|v_i\rangle$:

$$U_A|0\rangle|v_i\rangle = |0\rangle A|v_i\rangle + |\perp\rangle = \lambda_i|0\rangle|v_i\rangle + \sqrt{1-\lambda_i^2}|\perp_i'\rangle$$

- Apply $U_A^\dagger$ to $|0\rangle|v_i\rangle$:

$$U_A^\dagger|0\rangle|v_i\rangle = \lambda_i|0\rangle|v_i\rangle + \sqrt{1-\lambda_i^2}|\perp_i\rangle \qquad U_A^\dagger = \begin{bmatrix} A & * \\ * & * \end{bmatrix}$$

- Apply U_A again:

$$|0\rangle|v_i\rangle = \lambda_i\left(\lambda_i|0\rangle|v_i\rangle + \sqrt{1-\lambda_i^2}|\perp_i'\rangle\right) + \sqrt{1-\lambda_i^2}\,U_A|\perp_i\rangle$$

$$\Longrightarrow \; U_A|\perp_i\rangle = \sqrt{1-\lambda_i^2}|0\rangle|v_i\rangle - \lambda_i|\perp_i'\rangle$$

Qubitization

$$U_A|0\rangle|v_i\rangle = \lambda_i|0\rangle|v_i\rangle + \sqrt{1-\lambda_i^2}|\perp_i'\rangle$$

$$U_A|\perp_i\rangle = \sqrt{1-\lambda_i^2}|0\rangle|v_i\rangle - \lambda_i|\perp_i'\rangle$$

- U_A maps $\mathcal{H}_i \coloneqq \mathrm{span}\{|0\rangle|v_i\rangle, |\perp_i\rangle\}$ to $\mathcal{H}_i' \coloneqq \mathrm{span}\{|0\rangle|v_i\rangle, |\perp_i'\rangle\}$
- Similarly, $U_A^\dagger$ maps $\mathcal{H}_i'$ to $\mathcal{H}_i$

$$[U_A]_{\mathcal{H}_i \to \mathcal{H}_i'} = \begin{bmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{bmatrix}, \qquad [U_A^\dagger]_{\mathcal{H}_i' \to \mathcal{H}_i} = \begin{bmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{bmatrix}$$

Qubitization

$$[U_A]_{\mathcal{H}_i \to \mathcal{H}_i'} = [U_A^\dagger]_{\mathcal{H}_i' \to \mathcal{H}_i} = \begin{bmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{bmatrix}$$

$$[U_A^\dagger Z_\Pi U_A Z_\Pi]_{\mathcal{H}_i} = \begin{bmatrix} \lambda_i & -\sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & +\lambda_i \end{bmatrix}^2 = \begin{bmatrix} T_2(\lambda_i) & * \\ * & * \end{bmatrix}, \qquad \left[\left(U_A^\dagger Z_\Pi U_A Z_\Pi\right)^k\right]_{\mathcal{H}_i} = \begin{bmatrix} T_{2k}(\lambda_i) & * \\ * & * \end{bmatrix}$$

- $\left(U_A^\dagger Z_\Pi U_A Z_\Pi\right)^k$ is a block-encoding for $T_{2k}(A)$
- For odd polynomials, $U_A Z_\Pi \left(U_A^\dagger Z_\Pi U_A Z_\Pi\right)^k$ is a block-encoding for $T_{2k+1}(A)$

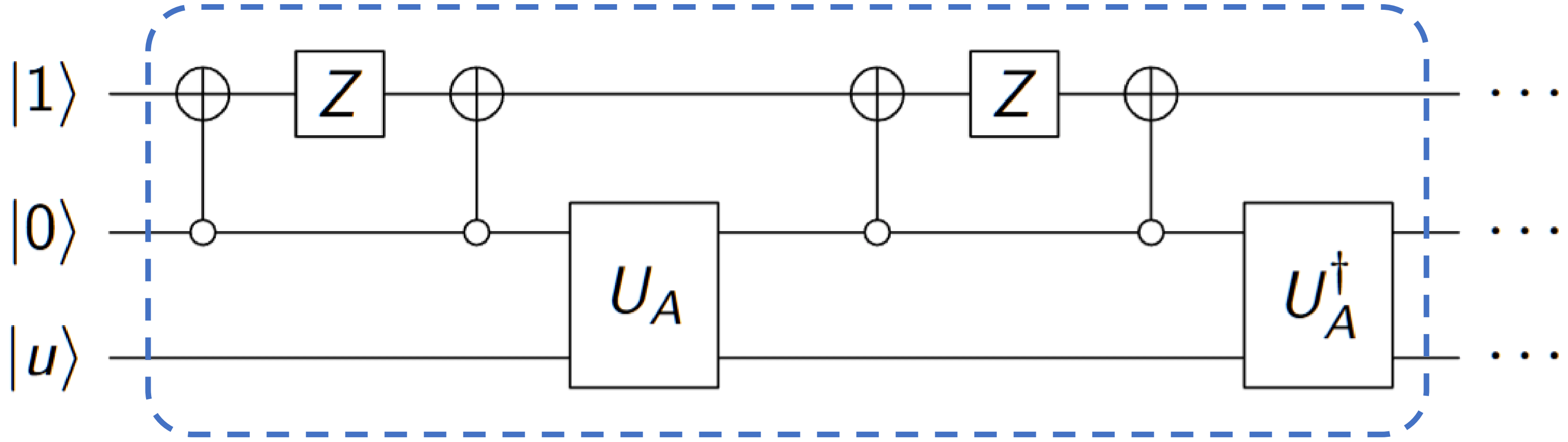
$\mathcal{H}_i$ and $\mathcal{H}_i'$ share a common basis state $|0\rangle|v_i\rangle$

Qubitization

- $U_{T_{2k}(A)} = \left(U_A^\dagger Z_\Pi U_A Z_\Pi\right)^k$
- $U_{T_{2k+1}(A)} = U_A Z_\Pi \left(U_A^\dagger Z_\Pi U_A Z_\Pi\right)^k$

Block-encoding for Chebyshev polynomials

Other polynomials?



Quantum signal processing (QSP)

- Consider a 2×2 qubitization:

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

- We have proved that $[(UZ)^k]_{1,1} = T_k(\lambda)$

- Notice that $Z = -\mathbf{i} \begin{bmatrix} e^{\frac{\pi}{2}\mathbf{i}} & 0 \\ 0 & e^{-\frac{\pi}{2}\mathbf{i}} \end{bmatrix} = -\mathbf{i} e^{\mathbf{i}\frac{\pi}{2}Z}$

- What if

$$e^{\mathbf{i}\phi_d Z} U e^{\mathbf{i}\phi_{d-1} Z} U e^{\mathbf{i}\phi_{d-2} Z} \dots U e^{\mathbf{i}\phi_0 Z}$$

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

Theorem (QSP, Low-Chuang '16).

Let

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

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

$$e^{\mathbf{i}\phi_d Z} U e^{\mathbf{i}\phi_{d-1} Z} U e^{\mathbf{i}\phi_{d-2} Z} \cdots U e^{\mathbf{i}\phi_0 Z} = \begin{bmatrix} p(\lambda) & -q(\lambda)\sqrt{1-\lambda^2} \\ q^*(\lambda)\sqrt{1-\lambda^2} & p^*(\lambda) \end{bmatrix}$$

if and only if $p, q \in \mathbb{C}[\lambda]$ such that

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

Theorem (QSP for real polynomials).

Let

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

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

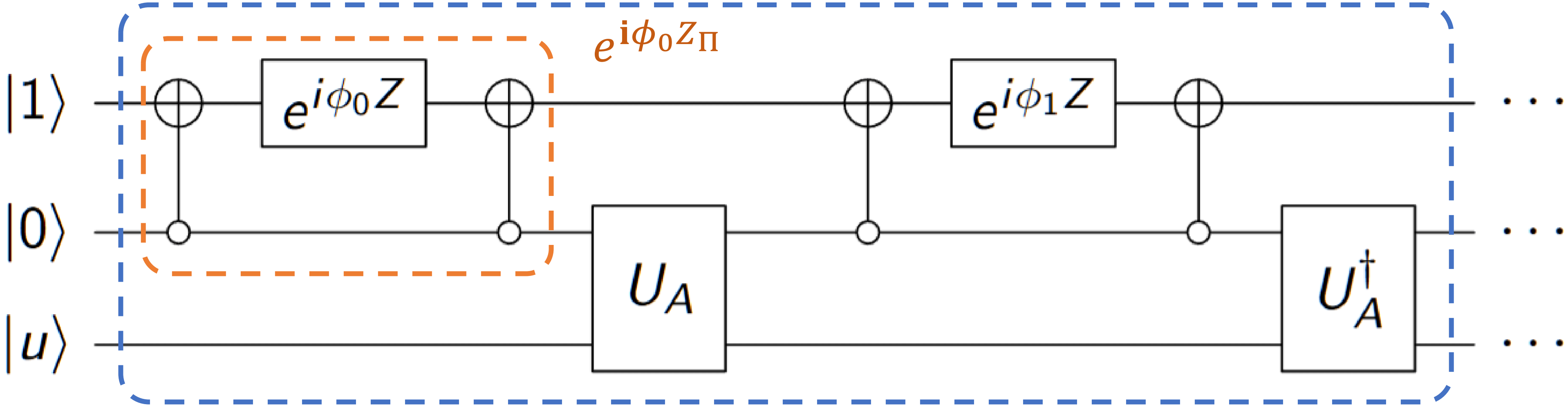
$$e^{\mathbf{i}\phi_d Z} U e^{\mathbf{i}\phi_{d-1} Z} U e^{\mathbf{i}\phi_{d-2} Z} \dots U e^{\mathbf{i}\phi_0 Z} = \begin{bmatrix} p(\lambda) + \mathbf{i} * & * \\ * & * \end{bmatrix}$$

if and only if $p \in \mathbb{R}[\lambda]$ such that

1. $\deg(p) \leq d$
2. p has parity $d \bmod 2$ $\quad$ ($p = p_{\text{odd}} + p_{\text{even}}$ and apply LCU)
3. $|p(\lambda)| \leq 1$ for all $\lambda \in [-1,1]$

Qubitization for general polynomials

For any Hermitian matrix A with $\|A\| \leq 1$, given its block-encoding U_A, and for any degree-d real polynomial $p(\lambda)$ with $|p(\lambda)| \leq 1$ for all $\lambda \in [-1,1]$, we can block-encode $p(A)$ with $\tilde{O}(d)$ cost:



Q1: how to find the phase factors $(\phi_0, \phi_1, \dots, \phi_d)$ for a given polynomial $p(\lambda)$?

Q2: how to approximate functions (e.g., $1/x$, e^x, x^c, $\mathrm{sgn}(x)$) by QSP polynomials?

Quantum linear algebra toolbox

- Basic linear algebra operations
- Linear systems of equations
- Matrix functions
  - Functions of Hermitian matrices: quantum signal processing (QSP), qubitization
  - Functions of general matrices: quantum singular value transformation (QSVT)
  - Linear combinations of Hamiltonian simulations (LCHS)

Quantum singular value transformation (QSVT)

Singular value transformation

- Let A be a general matrix with singular value decomposition $A = W\Sigma V^\dagger$
- Let f be an even or odd function
- Then the singular value transformation of A is

$$f^{\mathrm{SV}}(A) := \begin{cases} W f(\Sigma) V^\dagger, & \text{if } f \text{ is odd} \\ V f(\Sigma) V^\dagger, & \text{if } f \text{ is even} \end{cases} \qquad \neq f(A)\ !$$

$$A^{-1} = V\Sigma^{-1}W^\dagger$$
$$f^{\mathrm{SV}}(A) = W\Sigma^{-1}V^\dagger$$
$$f^{\mathrm{SV}}(A^\dagger) = V\Sigma^{-1}W^\dagger = A^{-1}$$

The goal of QSVT is to block encode $f^{\mathrm{SV}}(A)$

Quantum singular value transformation (QSVT)

Theorem (Gilyen et al. '2019).

For any (square) matrix A, let U_A be its block-encoding

Let $\Phi = (\phi_0, \dots, \phi_d)$ be the phase factor of a QSP polynomial $p \in \mathbb{R}[\lambda]$ of degree d

$$U_\Phi := \begin{cases} e^{\mathbf{i}\phi_d Z_\Pi} U_A e^{\mathbf{i}\phi_{d-1} Z_\Pi} \prod_{j=0}^{\frac{n-3}{2}} U_A^\dagger e^{\mathbf{i}\phi_{2j+1} Z_\Pi} U_A e^{\mathbf{i}\phi_{2j} Z_\Pi}, & \text{odd } d \\ e^{\mathbf{i}\phi_d Z_\Pi} \prod_{j=0}^{\frac{n-2}{2}} U_A^\dagger e^{\mathbf{i}\phi_{2j+1} Z_\Pi} U_A e^{\mathbf{i}\phi_{2j} Z_\Pi}, & \text{even } d \end{cases}$$

U_Φ is a block-encoding for $p^{\mathrm{SV}}(A)$ if and only if p is a QSP polynomial

Prove QSVT

Suppose $A = W\Sigma V^\dagger$. Then the SVD for U_A is

$$U_A = \begin{bmatrix} A & U_{01} \\ U_{10} & U_{11} \end{bmatrix} = \begin{bmatrix} W & \\ & W' \end{bmatrix} \underbrace{\begin{bmatrix} \Sigma & D_{01} \\ D_{10} & D_{11} \end{bmatrix}}_{D} \begin{bmatrix} V & \\ & V' \end{bmatrix}^\dagger$$

Consider the odd degree case:

$$U_\Phi = e^{\mathbf{i}\phi_d Z_\Pi} U_A e^{\mathbf{i}\phi_{d-1} Z_\Pi} \prod_{j=0}^{\frac{n-3}{2}} U_A^\dagger e^{\mathbf{i}\phi_{2j+1} Z_\Pi} U_A e^{\mathbf{i}\phi_{2j} Z_\Pi} \qquad e^{\mathbf{i}\phi Z_\Pi} = \begin{bmatrix} e^{\mathbf{i}\phi} I_n & \\ & e^{-\mathbf{i}\phi} I_{N-n} \end{bmatrix}$$

$$= \begin{bmatrix} W & \\ & W' \end{bmatrix} \underbrace{\left(e^{\mathbf{i}\phi_d Z_\Pi} D e^{\mathbf{i}\phi_{d-1} Z_\Pi} \prod_{j=0}^{\frac{n-3}{2}} D^\dagger e^{\mathbf{i}\phi_{2j+1} Z_\Pi} D e^{\mathbf{i}\phi_{2j} Z_\Pi} \right)}_{D_\Phi} \begin{bmatrix} V & \\ & V' \end{bmatrix}^\dagger$$

Interlude: Cosine-Sine (CS) decomposition

For any block unitary matrix $U = \begin{bmatrix} W & \\ & W' \end{bmatrix} \begin{bmatrix} D_{00} & D_{01} \\ D_{10} & D_{11} \end{bmatrix} \begin{bmatrix} V & \\ & V' \end{bmatrix}^{\dagger}$,

$$D := \begin{bmatrix} D_{00} & D_{01} \\ D_{10} & D_{11} \end{bmatrix} = \left[\begin{array}{ccc|ccc} 0 & & & I & & \\ & C & & & S & \\ & & I & & & 0 \\ \hline I & & & 0 & & \\ & S & & & -C & \\ & & 0 & & & -I \end{array}\right] = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \oplus \begin{bmatrix} C & S \\ S & -C \end{bmatrix} \oplus \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

where C and S are diagonal matrices with diagonal entries in $(0, 1)$

- 0 blocks could be rectangular and I blocks may not be the same size
- Unitary $\Longrightarrow C^2 + S^2 = I$

Prove QSVT

$$D_\Phi = e^{\mathbf{i}\phi_d Z_\Pi} D e^{\mathbf{i}\phi_{d-1} Z_\Pi} \prod_{j=0}^{\frac{n-3}{2}} D^\dagger e^{\mathbf{i}\phi_{2j+1} Z_\Pi} D e^{\mathbf{i}\phi_{2j} Z_\Pi}$$

$$D = \begin{bmatrix} \Sigma & D_{01} \\ D_{10} & D_{11} \end{bmatrix} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \oplus \begin{bmatrix} C & S \\ S & -C \end{bmatrix} \oplus \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

$$e^{\mathbf{i}\phi Z_\Pi} = \begin{bmatrix} e^{\mathbf{i}\phi} I & \\ & e^{-\mathbf{i}\phi} I \end{bmatrix} = \begin{bmatrix} e^{\mathbf{i}\phi} I & \\ & e^{-\mathbf{i}\phi} I \end{bmatrix} \oplus \begin{bmatrix} e^{\mathbf{i}\phi} I & \\ & e^{-\mathbf{i}\phi} I \end{bmatrix} \oplus \begin{bmatrix} e^{\mathbf{i}\phi} I & \\ & e^{-\mathbf{i}\phi} I \end{bmatrix}$$

$$D_\Phi = \left(\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}\right)_\Phi \oplus \left(\begin{bmatrix} C & S \\ S & -C \end{bmatrix}\right)_\Phi \oplus \left(\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}\right)_\Phi$$

Prove QSVT

$$\left(\begin{bmatrix}0 & I\\ I & 0\end{bmatrix}\right)_\Phi = \begin{bmatrix}e^{\mathbf{i}\phi_d}I & \\ & e^{-\mathbf{i}\phi_d}I\end{bmatrix}\begin{bmatrix}0 & I\\ I & 0\end{bmatrix}\begin{bmatrix}e^{\mathbf{i}\phi_{d-1}}I & \\ & e^{-\mathbf{i}\phi_{d-1}}I\end{bmatrix}\prod_{j=0}^{\frac{n-3}{2}}\begin{bmatrix}0 & I\\ I & 0\end{bmatrix}e^{\mathbf{i}\phi_{2j+1}Z_\Pi}\begin{bmatrix}0 & I\\ I & 0\end{bmatrix}e^{\mathbf{i}\phi_{2j}Z_\Pi}$$

$$= \begin{bmatrix} & e^{\mathbf{i}(\phi_d-\phi_{d-1})}I\\ e^{-\mathbf{i}(\phi_d-\phi_{d-1})}I & \end{bmatrix}\prod_{j=0}^{\frac{n-3}{2}}\begin{bmatrix}0 & I\\ I & 0\end{bmatrix}\begin{bmatrix}e^{\mathbf{i}\phi_{2j+1}}I & \\ & e^{-\mathbf{i}\phi_{2j+1}}I\end{bmatrix}\begin{bmatrix}0 & I\\ I & 0\end{bmatrix}\begin{bmatrix}e^{\mathbf{i}\phi_{2j}}I & \\ & e^{-\mathbf{i}\phi_{2j}}I\end{bmatrix}$$

$$= \begin{bmatrix} & e^{\mathbf{i}(\phi_d-\phi_{d-1})}I\\ e^{-\mathbf{i}(\phi_d-\phi_{d-1})}I & \end{bmatrix}\prod_{j=0}^{\frac{n-3}{2}}\begin{bmatrix}e^{\mathbf{i}(\phi_{2j}-\phi_{2j+1})}I & \\ & e^{-\mathbf{i}(\phi_{2j}-\phi_{2j+1})}I\end{bmatrix}$$

$$= \begin{bmatrix}0 & e^{-\mathbf{i}\sum_{j=0}^{d}(-1)^j\phi_j}I\\ e^{\mathbf{i}\sum_{j=0}^{d}(-1)^j\phi_j}I & 0\end{bmatrix} = \begin{bmatrix}p(0) & *\\ * & *\end{bmatrix}$$

Prove QSVT

$$\left(\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}\right)_{\Phi} = \begin{bmatrix} e^{\mathbf{i}\phi_d} I & \\ & e^{-\mathbf{i}\phi_d} I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} e^{\mathbf{i}\phi_{d-1}} I & \\ & e^{-\mathbf{i}\phi_{d-1}} I \end{bmatrix} \prod_{j=0}^{\frac{n-3}{2}} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} e^{\mathbf{i}\phi_{2j+1} Z_{\Pi}} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} e^{\mathbf{i}\phi_{2j} Z_{\Pi}}$$

$$= \begin{bmatrix} e^{\mathbf{i}(\phi_d+\phi_{d-1})} I & \\ & -e^{-\mathbf{i}(\phi_d+\phi_{d-1})} I \end{bmatrix} \prod_{j=0}^{\frac{n-3}{2}} \begin{bmatrix} e^{\mathbf{i}\left(\phi_{2j}+\phi_{2j+1}\right)} I & \\ & e^{-\mathbf{i}\left(\phi_{2j}+\phi_{2j+1}\right)} I \end{bmatrix}$$

$$= \begin{bmatrix} e^{\mathbf{i}\sum_{j=0}^{d}\phi_j} I & 0 \\ 0 & -e^{-\mathbf{i}\sum_{j=0}^{d}\phi_j} I \end{bmatrix} = \begin{bmatrix} p(I) & * \\ * & * \end{bmatrix}$$

Prove QSVT

$$\begin{bmatrix} C & S \\ S & -C \end{bmatrix} = \bigoplus_{i=1}^{m} \begin{bmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{bmatrix}$$

$$C = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix}$$

$$\left(\begin{bmatrix} C & S \\ S & -C \end{bmatrix}\right)_\Phi = \bigoplus_{i=1}^{m} \left(\begin{bmatrix} \lambda_i & \sqrt{1-\lambda_i^2} \\ \sqrt{1-\lambda_i^2} & -\lambda_i \end{bmatrix}\right)_\Phi = \bigoplus_{i=1}^{m} \begin{bmatrix} p(\lambda_i) & * \\ * & * \end{bmatrix} = \begin{bmatrix} p(C) & * \\ * & * \end{bmatrix}$$

Prove QSVT

$$D_\Phi = \left(\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}\right)_\Phi \oplus \left(\begin{bmatrix} C & S \\ S & -C \end{bmatrix}\right)_\Phi \oplus \left(\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}\right)_\Phi$$

$$= \begin{bmatrix} p(0) & * \\ * & * \end{bmatrix} \oplus \begin{bmatrix} p(C) & * \\ * & * \end{bmatrix} \oplus \begin{bmatrix} p(I) & * \\ * & * \end{bmatrix}$$

$$= \left[\begin{array}{ccc|c} p(0) & & & \\ & p(C) & & * \\ & & p(I) & \\ \hline & * & & * \end{array}\right] = \begin{bmatrix} p(\Sigma) & * \\ * & * \end{bmatrix}$$

Thus,

$$U_\Phi = \begin{bmatrix} W & \\ & W' \end{bmatrix} \begin{bmatrix} p(\Sigma) & * \\ * & * \end{bmatrix} \begin{bmatrix} V & \\ & V' \end{bmatrix}^\dagger = \begin{bmatrix} p^{\mathrm{SV}}(A) & * \\ * & * \end{bmatrix}$$

∎

Two remaining questions

Q1: how to find the phase factors $(\phi_0, \phi_1, \dots, \phi_d)$ for a given polynomial $p(\lambda)$?

Q2: how to approximate functions (e.g., $1/x$, e^x, x^c, $\mathrm{sgn}(x)$) by QSP polynomials?

Target function $f : \mathcal{D} \to \mathbb{R}$ → Low-degree polynomial approximation → Phase factors finding

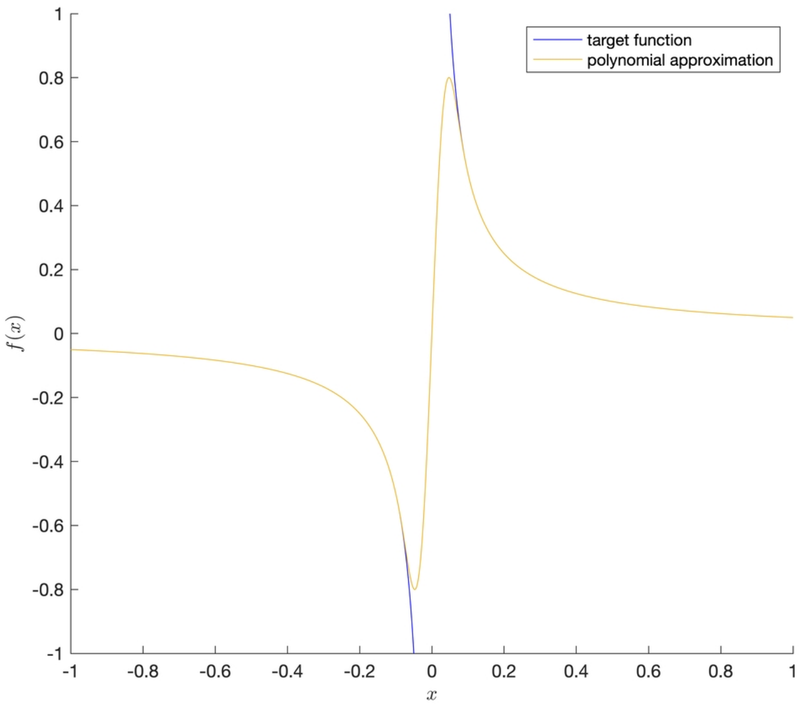
Example: QLSP

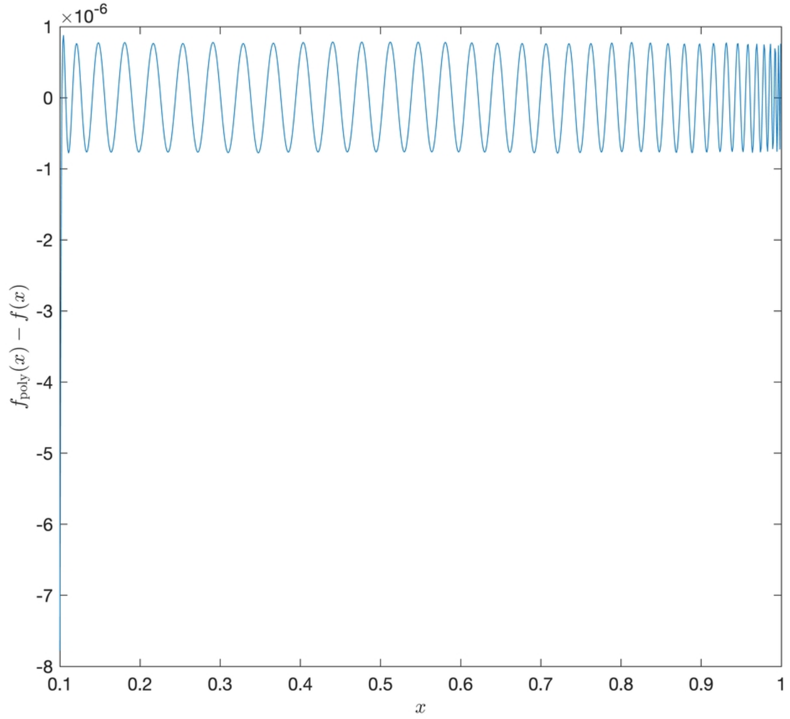
QSPPACK

Quantum Signal Processing PACKage

a package for quantum numerical linear algebra

- $f(x) = \frac{1}{x}$ for $x \in \mathcal{D} = [-1, -\kappa^{-1}] \cup [\kappa^{-1}, 1]$
- Preprocessing: $f(x) = \frac{1}{2\kappa x}$ so that $\|f\|_\infty = \frac{1}{2}$
- Chebyshev approximation





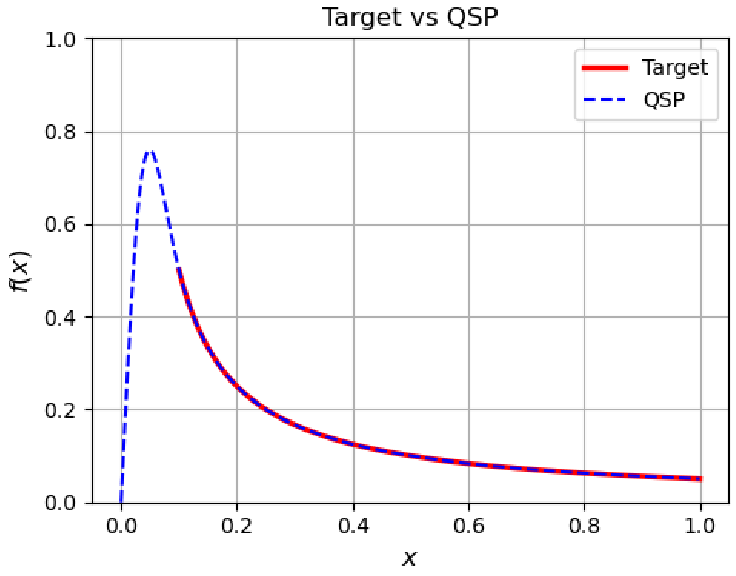
Example: QLSP

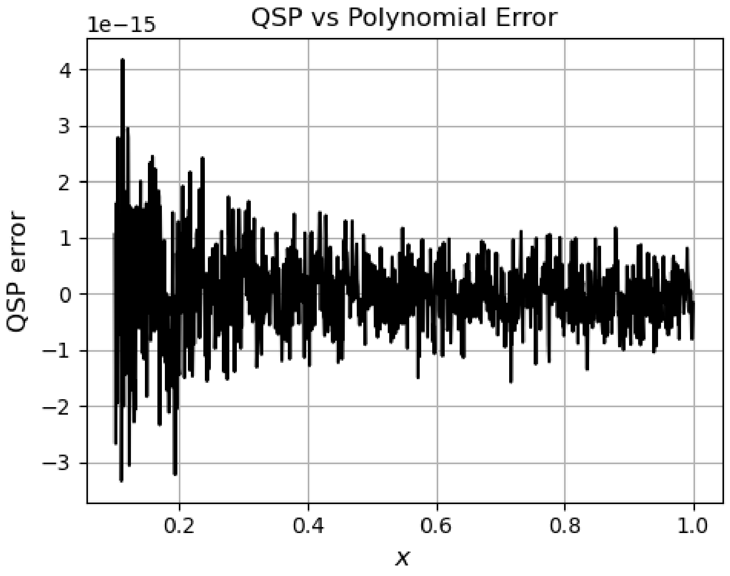
QSPPACK

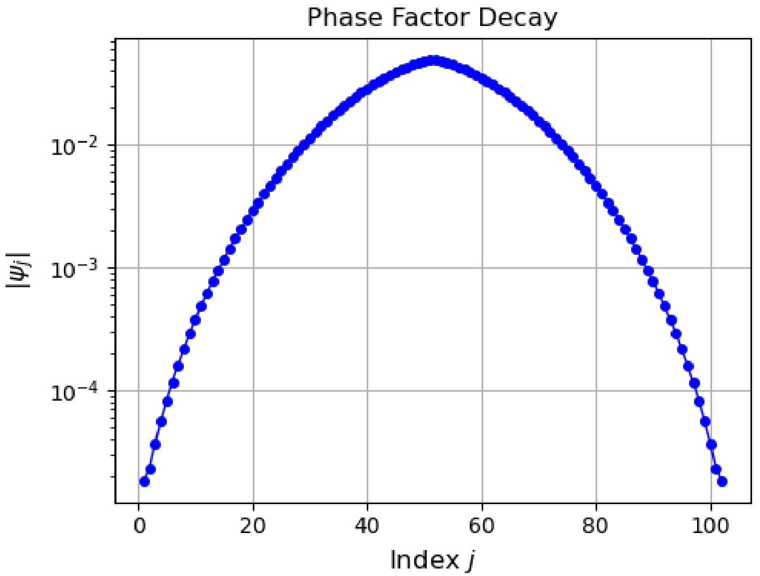
Quantum Signal Processing PACKage

a package for quantum numerical linear algebra

- $f(x) = \frac{1}{x}$ for $x \in \mathcal{D} = [-1, -\kappa^{-1}] \cup [\kappa^{-1}, 1]$
- Preprocessing: $f(x) = \frac{1}{2\kappa x}$ so that $\|f\|_\infty = \frac{1}{2}$
- Chebyshev approximation
- Phase factors finding





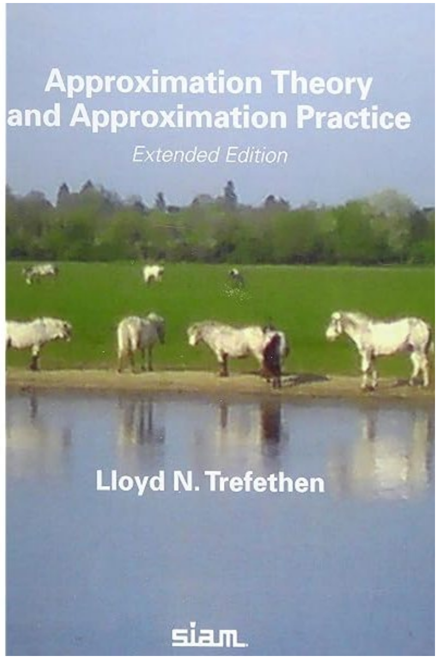


Polynomial approximation

Fact. For any nice (i.e. Lipschitz) function $f: [-1,1] \to \mathbb{C}$, f has a unique Chebyshev series decomposition:

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x), \qquad \text{where} \quad \sum_{k=0}^{\infty} |a_k| < \infty$$

- Equivalent to the Laurent series for $g: \mathbb{T} \to \mathbb{C}$
- Also equivalent to the Fourier series for $h: [-\pi, \pi] \to \mathbb{C}$
- More about Chebyshev approximation will be discussed in my course *CS 58500: Theoretical Computer Science Toolkit* next semester



Polynomial approximation

Theorem (Tang-Tian ’24).

Let f be an analytic function in $[-1, 1]$, analytically continuable to the interior of the Bernstein ellipse

$$\mathcal{E}_\rho = \{(z + z^{-1})/2 : |z| = \rho\}$$

where $\rho = 1 + \alpha$ and $|f| \leq M$ on $\mathcal{E}_\rho$

For $\delta \in \left(0,\ \frac{1}{C}\min\{1, \alpha^2\}\right)$ where C is a sufficiently large constant, $\varepsilon \in (0, 1)$, and $b > 1$, there is a polynomial q of degree $\mathcal{O}\left(\frac{b}{\delta}\log\frac{b}{\delta\epsilon}\right)$ such that

- $\|f - q\|_{[-1,1]} \leq M\epsilon$
- $\|q\|_{[-1-\delta,1+\delta]} \leq M$
- $\|q\|_{[-b,-1-\delta]\cup[1+\delta,b]} \leq M\epsilon$

Example 1: QLSP

$f(x) = \frac{1}{\kappa x}$ for $x \in [\kappa^{-1}, 1]$

1. Linearly transform the approximation region to $[-1,1]$:

$$x = \frac{1-\kappa^{-1}}{2} y + \frac{1+\kappa^{-1}}{2}, \qquad g(y) = \frac{2}{(\kappa-1)y+\kappa+1}$$

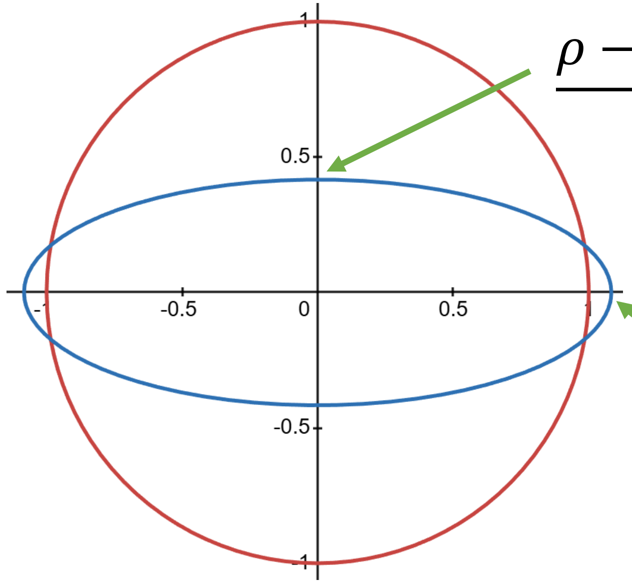
2. Determine the size of the Bernstein ellipse $\mathcal{E}_\rho$

→ $g(y)$ has a pole at $y = -\frac{\kappa+1}{\kappa-1}$

→ $1 + \frac{\alpha^2}{2} \leq \frac{\kappa+1}{\kappa-1} \quad \Rightarrow \quad \alpha \leq \sqrt{\frac{4}{\kappa-1}}$

→ Boundedness: $|g(z)| \leq g(-1-\alpha^2/2) \leq M$

$\Longrightarrow \quad \alpha \leq \sqrt{\frac{4(M-1)}{M(\kappa-1)}}$



$$\frac{\rho - \rho^{-1}}{2} \leq \alpha$$

$$\frac{\rho + \rho^{-1}}{2} \leq 1 + \frac{\alpha^2}{2}$$

Example 1: QLSP

$f(x) = \frac{1}{\kappa x}$ for $x \in [\kappa^{-1}, 1]$

1. Linearly transform the approximation region to $[-1,1]$:

$$x = \frac{1-\kappa^{-1}}{2} y + \frac{1+\kappa^{-1}}{2}, \qquad g(y) = \frac{2}{(\kappa-1)y + \kappa + 1}$$

2. Determine the size of the Bernstein ellipse $\mathcal{E}_\rho$: $\alpha \leq \sqrt{\frac{4(M-1)}{M(\kappa-1)}}$

3. Determine the bounded region: $x \in [-\kappa^{-1}, \kappa^{-1}] \quad \Leftrightarrow \quad y \in \left[-\frac{\kappa+3}{\kappa-1}, -1\right]$

→ For $\delta = \frac{1}{C}\alpha^2$ with a sufficiently large C,

$$-1 - \delta \geq -\frac{\kappa+3}{\kappa-1} \quad \Longrightarrow \quad \alpha \leq \sqrt{\frac{4C}{\kappa-1}}$$

Example 1: QLSP

$f(x) = \frac{1}{\kappa x}$ for $x \in [\kappa^{-1}, 1]$

1. Linearly transform the approximation region to $[-1,1]$:

$$x = \frac{1-\kappa^{-1}}{2} y + \frac{1+\kappa^{-1}}{2}, \qquad g(y) = \frac{2}{(\kappa-1)y + \kappa + 1}$$

2. Determine the size of the Bernstein ellipse $\mathcal{E}_\rho$: $\alpha \leq \sqrt{\frac{4(M-1)}{M(\kappa-1)}}$

3. Determine the bounded region: $x \in [-\kappa^{-1}, \kappa^{-1}] \quad \Leftrightarrow \quad y \in \left[-\frac{\kappa+3}{\kappa-1}, -1\right]$

4. Determine the vanishing region: $x \in [-1, -\kappa^{-1}] \quad \Leftrightarrow \quad y \in \left[-\frac{3\kappa+1}{\kappa-1}, -\frac{\kappa+3}{\kappa-1}\right]$

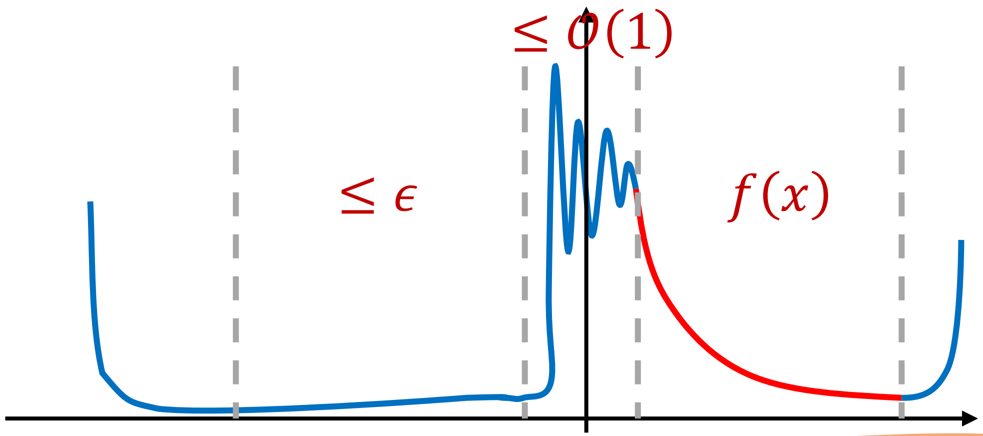
$\rightarrow \quad -b < -\frac{3\kappa+1}{\kappa-1} \quad \Longrightarrow \quad b = 4$

If we take $M = 2$,

then $\alpha = \sqrt{\frac{2}{\kappa-1}}$,

$\delta = \mathcal{O}(\kappa^{-1})$

Example 1: QLSP



$f(x) = \frac{1}{\kappa x}$ for $x \in [\kappa^{-1}, 1]$

1. Linearly transform the approximation region to $[-1,1]$:

$$x = \frac{1-\kappa^{-1}}{2} y + \frac{1+\kappa^{-1}}{2}, \qquad g(y) = \frac{2}{(\kappa-1)y + \kappa + 1}$$

By the approximation theorem, there is a polynomial p of degree $\mathcal{O}\left(\frac{b}{\delta}\log\frac{b}{\delta\epsilon}\right) = \mathcal{O}(\kappa \log(\kappa/\epsilon))$ s.t.

- $\|g(y) - p(y)\|_{[-1,1]} \leq \epsilon$
- $\|p(y)\|_{[-1-\delta, 1+\delta]} \leq 2$
- $\|p(y)\|_{[-4,-1-\delta]\cup[1+\delta,4]} \leq \epsilon$

$\Rightarrow$

Let $q(x) \coloneqq p\left(\frac{2}{1-\kappa^{-1}}\left(x - \frac{1+\kappa^{-1}}{2}\right)\right)$

- $\|f(x) - q(x)\|_{[\kappa^{-1},1]} \leq \epsilon$
- $\|q(x)\|_{[-\kappa^{-1},\kappa^{-1}]} \leq 2$
- $\|q(x)\|_{[-1,-\kappa^{-1}]} \leq \epsilon$

Example 2: Gibbs sampling

$f(x) = e^{\beta x}$ for $x \in [-1,0]$

1. Linearly transform the approximation region to $[-1,1]$:

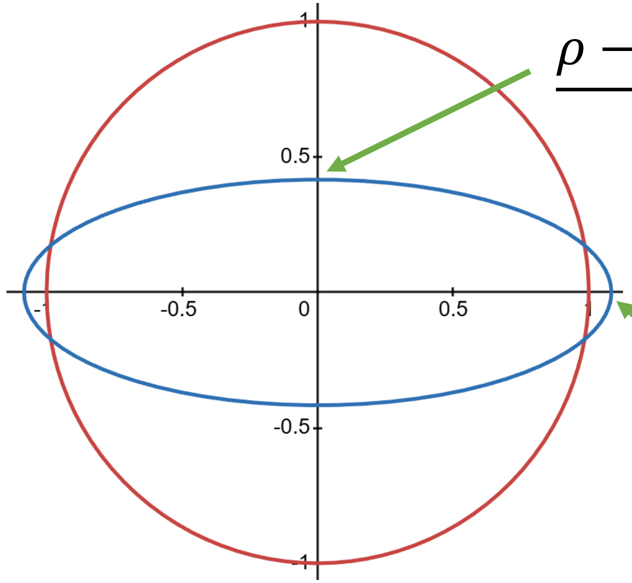
$$x = (y-1)/2\,, \qquad g(y) = \exp(\beta(y-1)/2)$$

2. Determine the size of the Bernstein ellipse $\mathcal{E}_\rho$:

→ $g(y)$ is analytic everywhere

→ $|g(z)| \leq g\left(1+\frac{\alpha^2}{2}\right) = e^{\frac{\beta\alpha^2}{4}} \leq M$

→ $\alpha \leq 2\sqrt{\beta^{-1}\log M}$



$$\frac{\rho-\rho^{-1}}{2} \leq \alpha$$

$$\frac{\rho+\rho^{-1}}{2} \leq 1+\frac{\alpha^2}{2}$$

Example 2: Gibbs sampling

$f(x) = e^{\beta x}$ for $x \in [-1,0]$

1. Linearly transform the approximation region to $[-1,1]$:
$$x = (y-1)/2\,, \qquad g(y) = \exp(\beta(y-1)/2)$$
2. Determine the size of the Bernstein ellipse $\mathcal{E}_\rho$: $\alpha \leq 2\sqrt{\beta^{-1}\log M}$
3. Determine the bounded region: $x \in [0,1] \iff y \in [1,3]$

→ $b = 3$

→ $M = 2$

→ $\delta = \Theta(\alpha^2) = \Theta\left(\frac{1}{\beta}\right)$

Example 2: Gibbs sampling

$f(x) = e^{\beta x}$ for $x \in [-1,0]$

1. Linearly transform the approximation region to $[-1,1]$:

$$x = (y-1)/2\,, \qquad g(y) = \exp(\beta(y-1)/2)$$

By the approximation theorem, there is a polynomial p of degree $\mathcal{O}\left(\frac{b}{\delta}\log\frac{b}{\delta\epsilon}\right) = \mathcal{O}(\beta\log(\beta/\epsilon))$ s.t.

- $\|g(y) - p(y)\|_{[-1,1]} \leq \epsilon$
- $\|p(y)\|_{[1,1+\delta]} \leq 2$
- $\|p(y)\|_{[1+\delta,3]} \leq \epsilon$

$\Longrightarrow$

Let $q(x) \coloneqq p(2x+1)$

- $\|f(x) - q(x)\|_{[-1,0]} \leq \epsilon$
- $\|q(x)\|_{[0,1]} \leq 2$

The block-encoding for $e^{\beta A}$ has been used in quantum LP/SDP solvers (van Apeldoorn-Gilyén ’19; Bouland et al. ’23)

Finding phase factors

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

- Remez exchange algorithm
- Weiss algorithm/root-finding (Gilyen et al. ’18; Haah ’19) Numerically unstable
- Capitalization (Chao et al. ’20)
- Prony’s method (Ying ’22)
- Half-Cholesky (Ni-Ying ’24)
- Inverse non-linear fast Fourier transform (Ni et al. ’25)
- Optimization based algorithm (Dong et al. ’20; Wang et al ’22; Dong et al.’23)
- Fixed point iteration (Dong et al. ’22)

Direct methods

Iterative methods

Examples of phase factors

For $\Phi = (\phi_0, \phi_1, \dots, \phi_d) \in \mathbb{R}^{d+1}$, define the QSP operator in W-convention:

$$U_d(x, \Phi) := e^{\mathbf{i}\phi_0 Z} \prod_{j=1}^{d} \left(W(x) e^{\mathbf{i}\phi_j Z} \right), \qquad W(x) = e^{\mathbf{i} \arccos(x) X} = \begin{bmatrix} x & \mathbf{i}\sqrt{1-x^2} \\ \mathbf{i}\sqrt{1-x^2} & x \end{bmatrix}$$

- $W(x) = e^{-\mathbf{i}\frac{\pi}{4}Z} O(x) e^{\mathbf{i}\frac{\pi}{4}Z}$, where $O(x) = \begin{bmatrix} x & -\sqrt{1-x^2} \\ \sqrt{1-x^2} & x \end{bmatrix}$ as in our previous definition

Example 1: Chebyshev polynomial

- $\Phi = (0, \dots, 0)$
- $U_d(x, \Phi) = \begin{bmatrix} \cos(d\theta) & \mathbf{i}\sin(d\theta) \\ \mathbf{i}\sin(d\theta) & \cos(d\theta) \end{bmatrix}$ where $x = \cos(\theta)$
- $T_d(x) = \mathrm{Re}\left(U_d(x, \Phi)_{1,1} \right)$

Examples of phase factors

For $\Phi = (\phi_0, \phi_1, \dots, \phi_d) \in \mathbb{R}^{d+1}$, define the QSP operator in W-convention:

$$U_d(x, \Phi) \coloneqq e^{\mathbf{i}\phi_0 Z} \prod_{j=1}^{d} \left(W(x) e^{\mathbf{i}\phi_j Z} \right), \qquad W(x) = e^{\mathbf{i}\arccos(x) X} = \begin{bmatrix} x & \mathbf{i}\sqrt{1-x^2} \\ \mathbf{i}\sqrt{1-x^2} & x \end{bmatrix}$$

- $W(x) = e^{-\mathbf{i}\frac{\pi}{4}Z} O(x) e^{\mathbf{i}\frac{\pi}{4}Z}$, where $O(x) = \begin{bmatrix} x & -\sqrt{1-x^2} \\ \sqrt{1-x^2} & x \end{bmatrix}$ as in our previous definition

Example 2: all-zero function

- $\Phi = \left(\frac{\pi}{4}, 0, \dots, 0, \frac{\pi}{4}\right)$
- $U_d(x, \Phi) = e^{\mathbf{i}\frac{\pi}{4}Z} e^{\mathbf{i}d\theta X} e^{\mathbf{i}\frac{\pi}{4}Z} = \begin{bmatrix} \mathbf{i}\cos(d\theta) & \mathbf{i}\sin(d\theta) \\ \mathbf{i}\sin(d\theta) & -\mathbf{i}\cos(d\theta) \end{bmatrix}$ where $x = \cos(\theta)$
- $0 = \mathrm{Re}\left(U_d(x, \Phi)_{1,1}\right)$

Examples of phase factors

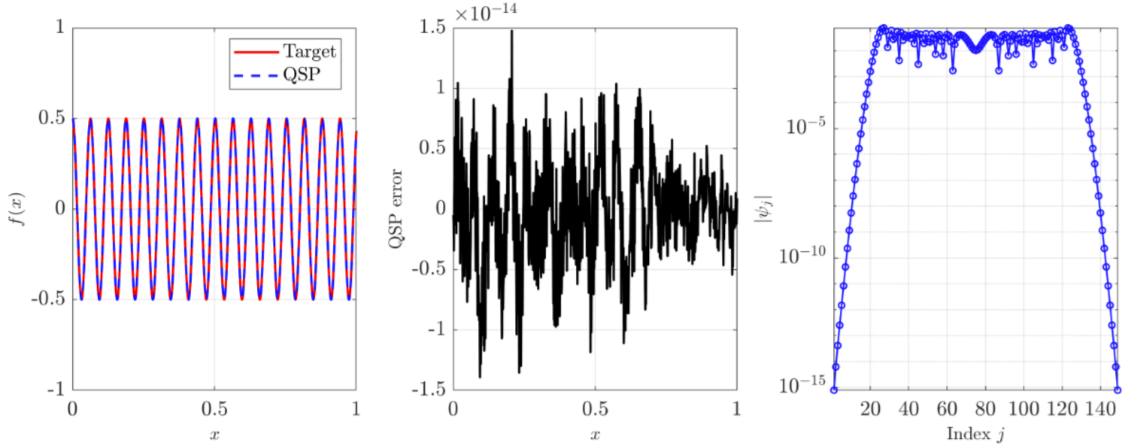
For $\Phi = (\phi_0, \phi_1, \dots, \phi_d) \in \mathbb{R}^{d+1}$, define the QSP operator in W-convention:

$$U_d(x,\Phi) := e^{\mathrm{i}\phi_0 Z}\prod_{j=1}^{d}\left(W(x)e^{\mathrm{i}\phi_j Z}\right), \qquad W(x) = e^{\mathrm{i}\arccos(x)X} = \begin{bmatrix} x & \mathrm{i}\sqrt{1-x^2} \\ \mathrm{i}\sqrt{1-x^2} & x \end{bmatrix}$$

- $W(x) = e^{-\mathrm{i}\frac{\pi}{4}Z} O(x) e^{\mathrm{i}\frac{\pi}{4}Z}$, where $O(x) = \begin{bmatrix} x & -\sqrt{1-x^2} \\ \sqrt{1-x^2} & x \end{bmatrix}$ as in our previous definition

Example 3: trigonometric functions $\left(f(x) = \frac{1}{2}\cos(100x)\right)$

- $d = 150$



- Symmetric
- Decaying fast

Symmetric phase factors

- Due to the parity constraint, the target polynomial has degree of freedom $\tilde{d} = \left\lceil \frac{d+1}{2} \right\rceil$
- On the other hand, Φ has $\mathrm{dof} = d$
- We define the symmetric phase factors: $\Phi = (\phi_0, \phi_1, \dots, \phi_1, \phi_0) \in D_d$, where

$$D_d = \begin{cases} \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^{\frac{d}{2}} \times [-\pi, \pi) \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^{\frac{d}{2}}, & d \text{ is odd} \\ \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^{d+1}, & d \text{ is even} \end{cases}$$

Existence of phase factors

Theorem (Gilyén et al. '18). Let $f(x) \in \mathbb{R}[x]$ be a degree-d polynomial for some $d \geq 1$ such that

- $f(x)$ has parity $d \bmod 2$
- $\|f(x)\|_{[-1,1]} \leq 1$

Then there exists phase factors Φ such that $f(x) = \mathrm{Re}\big(U_d(x, \Phi)_{1,1}\big)$

However, the choice of Φ is highly non-unique. There are combinatorically many global optimal solutions

Uniqueness of phase factors

Theorem (Wang-Dong-Lin ’22). For any $P(x) \in \mathbb{C}[x]$ and $Q(x) \in \mathbb{R}[x]$ satisfying

1. $\deg(P) = d, \deg(Q) = d - 1$
2. P has parity $(d \bmod 2)$ and Q has parity $(d - 1 \bmod 2)$
3. $|P(x)|^2 + (1 - x^2)|Q(x)|^2 = 1$ for all $x \in [-1,1]$
4. If d is odd, then the leading coefficient of Q is positive

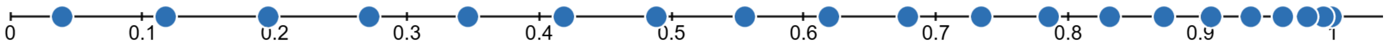
there exists a unique set of symmetric phase factors $\Phi := (\phi_0, \phi_1, \dots, \phi_1, \phi_0) \in D_d$ such that

$$U_d(x, \Phi) = \begin{bmatrix} P(x) & \mathbf{i}Q(x)\sqrt{1 - x^2} \\ \mathbf{i}Q(x)\sqrt{1 - x^2} & P^*(x) \end{bmatrix}$$

- Still highly non-unique if $\mathrm{Im}(P)$ and Q are not specified

Finding phase factors via optimization

- Parity: only $\tilde{d} = \left\lceil \frac{d+1}{2} \right\rceil$ degree of freedom to determine $f(x)$
- Sampling on Chebyshev nodes $x_k = \cos\left(\frac{2k-1}{4\tilde{d}}\pi\right)$, for $k = 1, \dots, \tilde{d}$

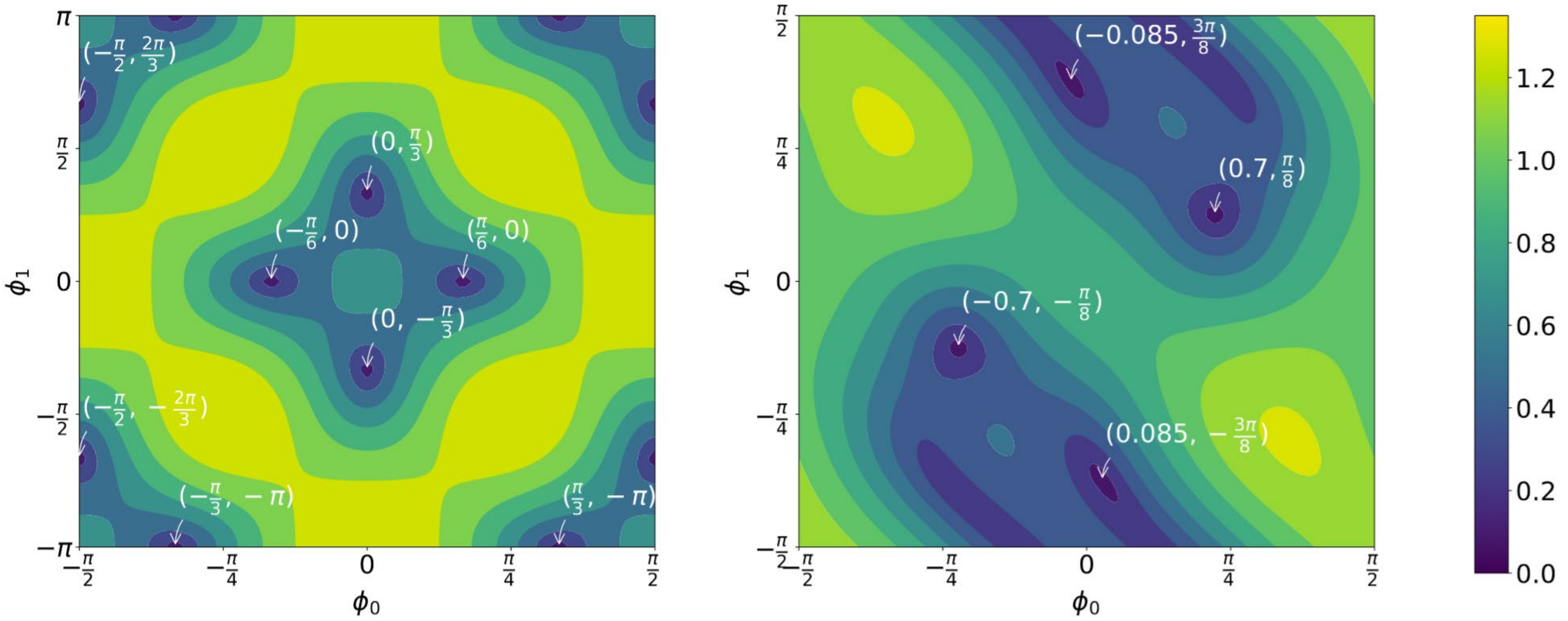


- Define the loss function:

$$\Phi^\star = \arg\min_{\Phi \in D_d} F(\Phi), \qquad F(\Phi) \coloneqq \frac{1}{\tilde{d}} \sum_{k=1}^{\tilde{d}} \left| \mathrm{Re}\big(U_d(x_k, \Phi)_{1,1}\big) - f(x_k) \right|^2$$

- Global minimum: $F(\Phi^\star) = 0$

Optimization landscape



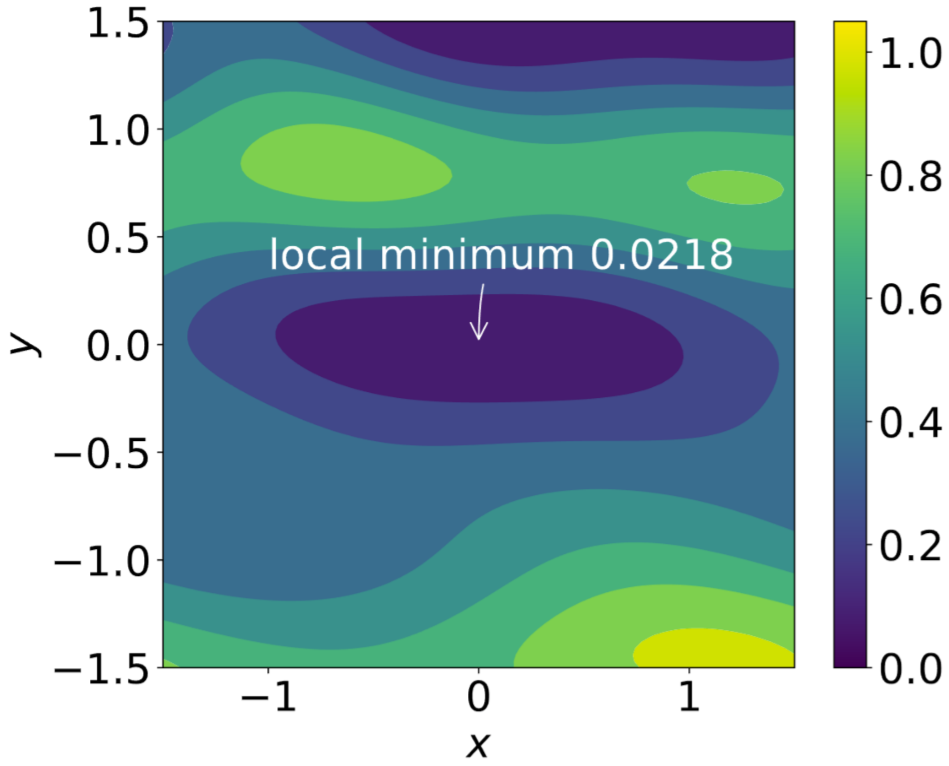
$$f(x) = x^2 - \frac{1}{2}$$

$$f(x) = \frac{1}{\sqrt{3}}x^3 - \frac{2}{\sqrt{3}}x$$

Only global minima

Local minima exists (and there are many)

If slightly increase the degree d



Characterizing all global minimizers

Theorem (Wang-Dong-Lin ’22). There is a bijection between global minimizers and all QSP-admissible $\big(P(x), Q(x)\big)$ pairs with $\mathrm{Re}(P)(x) = f(x)$

- $P = f + \mathbf{i} \cdot \mathrm{Im}(P)$
- $\mathrm{Im}(P), Q \in \mathbb{R}[x]$ are called the complementary polynomials
- By the normalization condition:

$$|P|^2 + (1 - x^2)Q^2 = 1 \quad \Longrightarrow \quad 1 - f^2 = \mathrm{Im}(P)^2 + (1 - x^2)Q^2$$

- For $x = \cos\theta = \frac{1}{2}\left(e^{\mathbf{i}\theta} + e^{-\mathbf{i}\theta}\right) = \frac{1}{2}(z + z^{-1})$, $1 - f((z + z^{-1})/2)^2$ can be expressed as a Laurent polynomial $\mathbb{C}[z, z^{-1}]$, which determines all roots of RHS
- All possible $\mathrm{Im}(P)$ and Q can be reconstructed from the roots

Magic initial guess

Fixed initial guess $\Phi_0 = \left(\frac{\pi}{4}, 0, \dots, 0, \frac{\pi}{4}\right)$

- Used in *qsppack* for all examples
- Robust for all target functions seen so far!
- One special solution for the all-zero function $f = 0$

Wang-Dong-Lin ’22: when $\|f\|_\infty \lesssim d^{-1}$,

- A neighborhood around Φ_0 is strongly convex
- A global minimizer is contained in that region

Fixed point iteration

- We define a mapping from phase factors to the function's Chebyshev coefficients:

$$F(\Phi) = c \quad \text{s.t.} \quad \mathrm{Re}\big(U_d(x,\Phi)_{1,1}\big) = \sum_{j=0}^{d} c_j T_j(x)$$

- Start from $\Phi_0 = \left(\frac{\pi}{4}, 0, \dots, 0, \frac{\pi}{4}\right)$
- Fixed point iteration:

$$\Phi_{t+1} = \Phi_t - \frac{1}{2}\left(F(\Phi_t) - c^\star\right)$$

Dong-Lin-Ni-Wang '24: when $\|c^\star\|_1 \approx 1$, the FPI algorithm converges linearly to the global minimizer

Connection to nonlinear Fourier transform (NLFT)

For any compactly supported sequence $\gamma \in \ell_0(\mathbb{Z})$, whose support lies in $[m, n] \subset \mathbb{Z}$, its nonlinear Fourier transform is defined as:

$$\overset{\frown}{\gamma}(z) := \prod_{k=m}^{n} \left(\frac{1}{\sqrt{1 + |\gamma_k|^2}} \begin{bmatrix} 1 & \gamma_k z^k \\ -\overline{\gamma_k} z^{-k} & 1 \end{bmatrix} \right) = \begin{bmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{bmatrix}$$

where $a(z), b(z)$ are Laurent polynomials and $a^*(z) = \overline{a\left(\overline{z}^{-1}\right)}$

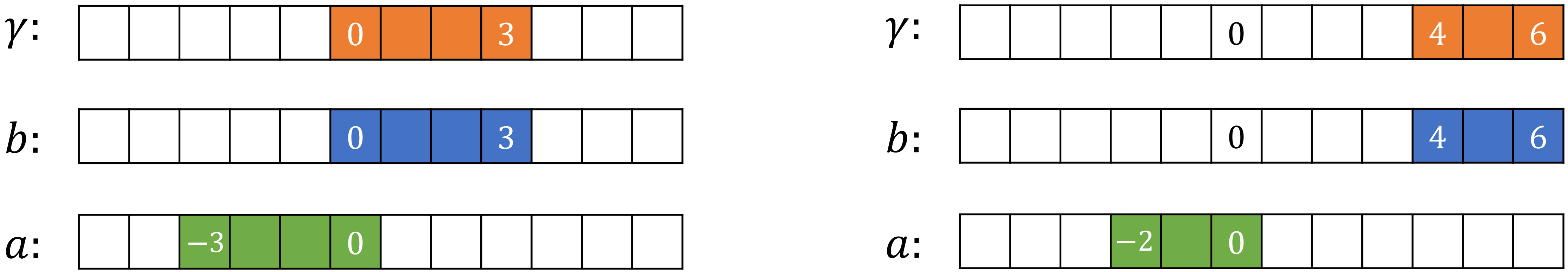
Finding phase factors via inverse NLFT

- For the degree-d target polynomial $f(x)$, let $b\left(e^{2\mathbf{i}\theta}\right) = e^{\mathbf{i}d\theta} f(\cos\theta)$
- We can find Laurent polynomial $a(z)$, $b(z)$ with $aa^* + bb^* = 1$ by the Weiss algorithm
- Apply inverse nonlinear Fourier transform to obtain γ and define $\phi_k = \arctan \gamma_k$ for $k = 0, \dots, d$
- $f(x) = \mathrm{Im}\left(U_d(x_k, \Phi)_{1,1}\right)$

See the recent survey (Lin '25, *arXiv:2510.00443*)

NLFT properties

- **P1:** If $\operatorname{supp}(\gamma) = [m, n]$, then $\operatorname{supp}(b) = [m, n]$ and $\operatorname{supp}(a) = [m - n, 0]$



- **P2:** If $\gamma' = \gamma_{\to 1}$, then $a' = a$ and $b' = zb$

Layer stripping algorithm

Suppose the unknown γ is supported on $[0, n-1]$

$$\widehat{[\gamma_0, \gamma_1, \dots, \gamma_{n-1}]}(z) = \prod_{k=0}^{n-1} \left(\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{bmatrix} 1 & \gamma_k z^k \\ -\overline{\gamma_k} z^{-k} & 1 \end{bmatrix} \right) = \begin{bmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{bmatrix}$$

$$= \frac{1}{\sqrt{1+|\gamma_0|^2}} \begin{bmatrix} 1 & \gamma_0 \\ -\overline{\gamma_0} & 1 \end{bmatrix} \cdot \widehat{[0, \gamma_1, \dots, \gamma_{n-1}]}(z)$$

- If $\widehat{[\gamma_1, \gamma_2, \dots, \gamma_{n-1}]}(z) = \begin{bmatrix} a_1 & b_1 \\ -b_1^* & a_1^* \end{bmatrix}$, then $\widehat{[0, \gamma_1, \dots, \gamma_{n-1}]}(z) = \begin{bmatrix} a_1 & zb_1 \\ -(zb_1)^* & a_1^* \end{bmatrix}$ by **P2**
- We get a linear equation:

$$\frac{1}{\sqrt{1+|\gamma_0|^2}} \begin{bmatrix} 1 & \gamma_0 \\ -\overline{\gamma_0} & 1 \end{bmatrix} \begin{bmatrix} a_1(z) & zb_1(z) \\ -z^{-1}b_1^*(z) & a_1^*(z) \end{bmatrix} = \begin{bmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} a_1(z) & zb_1(z) \\ -z^{-1}b_1^*(z) & a_1^*(z) \end{bmatrix} = \frac{1}{\sqrt{1+|\gamma_0|^2}} \begin{bmatrix} 1 & -\gamma_0 \\ \overline{\gamma_0} & 1 \end{bmatrix} \begin{bmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{bmatrix}$$

Layer stripping algorithm

Suppose $\overbrace{[\gamma_0, \dots, \gamma_{n-1}]}(z) = \begin{bmatrix} a & b \\ -b^* & a^* \end{bmatrix}$ and $\overbrace{[\gamma_1, \dots, \gamma_{n-1}]}(z) = \begin{bmatrix} a_1 & b_1 \\ -b_1^* & a_1^* \end{bmatrix}$

Then

$$a_1^*(z) = \frac{a^*(z) + \overline{\gamma_0} b(z)}{\sqrt{1 + |\gamma_0|^2}}, \qquad b_1(z) = \frac{-\gamma_0 a^*(z) + b(z)}{z\sqrt{1 + |\gamma_0|^2}}$$

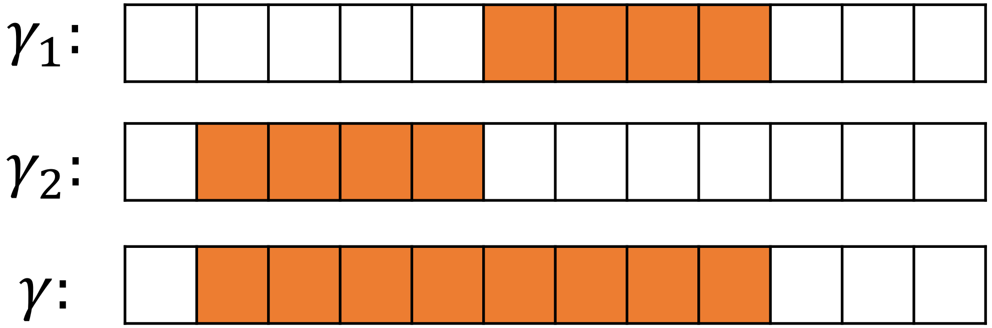
- By **P1**, $b_1(z)$ is a polynomial, which implies that

$$\gamma_0 = \frac{b(0)}{a^*(0)}$$

- Using (a_1, b_1), we can apply the same procedure to recover γ_1 and (a_2, b_2) for $\overbrace{[\gamma_2, \dots, \gamma_{n-1}]}(z)$, and then $\gamma_3, \gamma_4, \dots$
- Complexity: $\mathcal{O}(n^2)$

Fast nonlinear Fourier transform

- **P3:** If $\mathrm{supp}(\gamma_1)$ is on the left of $\mathrm{supp}(\gamma_2)$, then $\overbrace{\gamma_1 \circ \gamma_2}(z) = \overbrace{\gamma_1}(z)\overbrace{\gamma_2}(z)$



- **Idea:** strip $n/2$ layers at a time

$$\overbrace{[\gamma_0, \gamma_1, \dots, \gamma_{m-1}]}\ \overbrace{[0, \dots, 0, \gamma_m, \dots, \gamma_{n-1}]} = \overbrace{[\gamma_0, \gamma_1, \dots, \gamma_{n-1}]}$$

$$\begin{bmatrix} \eta_m^* & \xi_m \\ -\xi_m^* & \eta_m \end{bmatrix} \qquad \begin{bmatrix} a_m & z^m b_m \\ -z^{-m} b_m^* & a_m^* \end{bmatrix} \quad = \quad \begin{bmatrix} a & b \\ -b^* & a^* \end{bmatrix}$$

$$\begin{bmatrix} z_m b_m(z) \\ a_m^*(z) \end{bmatrix} = \begin{bmatrix} \eta_m(z) & -\xi_m(z) \\ \xi_m^*(z) & \eta_m^*(z) \end{bmatrix} \begin{bmatrix} b_0(z) \\ a_0^*(z) \end{bmatrix}$$

- Polynomial multiplication via FFT
- Total complexity $\mathcal{O}(n \log n)$