Quantum singular value transformation: theory and practice

Zane Marius Rossi

The University of Tokyo

✏️ *An overview of QSP/QSVT literature*†

Early work: focused on Hamiltonian simulation, composite pulses. [YLC14, LYC16, LC19, Haa19].

Broad and pedagogical works on QSVT: general reference. [GSLW19], [MRTC21].

For a CS reader: connected to numerical linear algebra. [TT23].

For a math reader: connected to nonlinear Fourier theory. [AMT23, ALM+24].

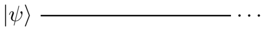
Generalizations, extensions, variants: recent progress in simplifying analysis and relaxing input assumptions. [MW23, RC22, WDL21, DLNW22, RCC23].

† Green text indicates a recommended entry-level paper.

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_{\Phi}(x)$.
Oracle access to structured unitary $W(x) = e^{i \cos^{-1}(x) \sigma_x}$.

User 'programs' $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x

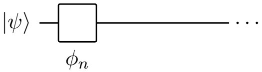


[1] [LYC16,LC17]

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_\Phi(x)$.
Oracle access to structured unitary $W(x) = e^{i \cos^{-1}(x)\sigma_x}$.

User 'programs' $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x

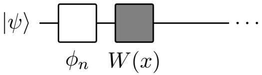


[1] [LYC16,LC17]

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_{\Phi}(x)$.
Oracle access to structured unitary $W(x) = e^{i \cos^{-1}(x) \sigma_x}$.

User ‘programs’ $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x

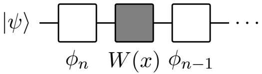


[1] [LYC16,LC17]

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_\Phi(x)$.
Oracle access to structured unitary $W(x) = e^{i \cos^{-1}(x)\sigma_x}$.

User 'programs' $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x

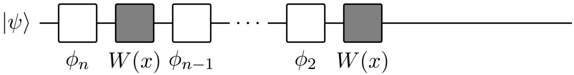


[1] [LYC16,LC17]

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_{\Phi}(x)$.
Oracle access to structured unitary $W(x) = e^{i \cos^{-1}(x) \sigma_x}$.

User 'programs' $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x

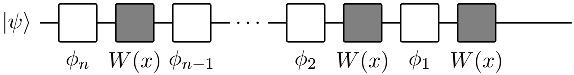


[1] [LYC16,LC17]

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_{\Phi}(x)$.
Oracle access to structured unitary $W(x) = e^{i\cos^{-1}(x)\sigma_x}$.

User 'programs' $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x

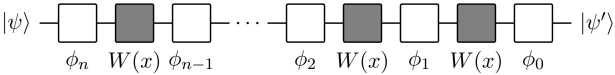


[1] [LYC16,LC17]

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_\Phi(x)$.
Oracle access to structured unitary $W(x) = e^{i\cos^{-1}(x)\sigma_x}$.

User 'programs' $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x

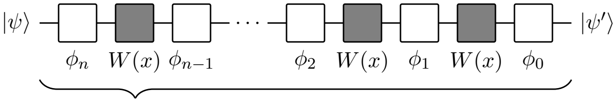


[1] [LYC16,LC17]

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_\Phi(x)$.
Oracle access to structured unitary $W(x) = e^{i\cos^{-1}(x)\sigma_x}$.

User 'programs' $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x



$$U(\Phi, x) = e^{i\phi_0\sigma_z}\prod_{j=1}^{k} W(x)\, e^{i\phi_j\sigma_z} = \begin{bmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ(x)^*\sqrt{1-x^2} & P(x)^* \end{bmatrix}$$

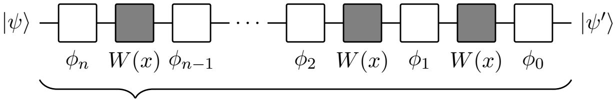
Can go $P \mapsto \Phi$ and $\Phi \mapsto P$ efficiently; just like classical filter![1]

[1] [LYC16,LC17]

Quantum signal processing (QSP) is an SU(2)-valued map

Single-qubit alternating circuit taking $\Phi \in \mathbb{R}^{n+1}$ to $U_\Phi(x)$.
Oracle access to structured unitary $W(x) = e^{i \cos^{-1}(x)\sigma_x}$.

User 'programs' $\Phi = \{\phi_0, \cdots, \phi_n\}$ to condition unitary on signal x



$$U(\Phi, H) = e^{i\phi_0 Z}\prod_{j=1}^{k} W(H)\, e^{i\phi_j Z} = \begin{bmatrix} P(H) & iQ(H)\sqrt{1-H^2} \\ iQ(H)^*\sqrt{1-H^2} & P(H)^* \end{bmatrix} \begin{matrix} |0\rangle \\ |1\rangle \end{matrix}$$

$$\qquad\qquad\qquad\qquad |0\rangle \qquad\qquad\qquad |1\rangle$$

Can go $P \mapsto \Phi$ and $\Phi \mapsto P$ efficiently; just like classical filter![1]

Claim: can replace[2] $x \in \mathbb{R}$ with $H = H^\dagger \in \mathbb{C}^{m\times m}$; block $U_\Phi(H)$.

[1] [LYC16,LC17] [2] [LC19, GSLW19]

📚 ***Talk outline and intent***

Part I: motivate and demystify QSVT by providing two ‘lifting arguments’ with commentary.

Part II: discuss reduction to QSP, and functional analytic tools that make this reduction worthwhile.

Part III: discuss common applications, guidelines, and recent extensions (multivar, randomized, functional programming, etc.).

Part I: Lifting arguments for QSVT

Block encoding; adapted from [TT23]

Let $A \in \mathbb{C}^{r\times c}$ and $\alpha, \varepsilon > 0$; a unitary $U \in \mathbb{C}^{d\times d}$ is an (α,ε)-block encoding of A if there exist $B_{L,1} \in \mathbb{C}^{d\times r}, B_{R,1} \in \mathbb{C}^{d\times c}$ with orthonormal columns s.t. $\|A - \alpha B_{L,1}^\dagger U B_{R,1}\|_{\mathsf{op}} \le \varepsilon$. We denote $B_{L,1}^\dagger B_{L,1} = \Pi_L$ and $B_{R,1}^\dagger B_{R,1} = \Pi_R$, orthogonal projectors.

Block encoding; adapted from [TT23]

Let $A \in \mathbb{C}^{r\times c}$ and $\alpha, \varepsilon > 0$; a unitary $U \in \mathbb{C}^{d\times d}$ is an (α, ε)-block encoding of A if there exist $B_{L,1} \in \mathbb{C}^{d\times r}, B_{R,1} \in \mathbb{C}^{d\times c}$ with orthonormal columns s.t. $\|A - \alpha B_{L,1}^{\dagger} U B_{R,1}\|_{\mathsf{op}} \leq \varepsilon$. We denote $B_{L,1}^{\dagger} B_{L,1} = \Pi_L$ and $B_{R,1}^{\dagger} B_{R,1} = \Pi_R$, orthogonal projectors.

U, a block matrix, contains something ε-close to αA in its top left sub-block. Taking $(1, 0)$ block encoding, with $B_L = (B_{L,1}, B_{L,2})$ and $B_R = (B_{R,1}, B_{R,2})$ unitary completions of $B_{L,1}, B_{R,1}$:

$$B_L^{\dagger} U B_R = \begin{bmatrix} A & * \\ * & * \end{bmatrix}, \quad B_L^{\dagger}(\Pi_L U \Pi_R) B_R = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}.$$

Block encoding; adapted from [TT23]

Let $A \in \mathbb{C}^{r\times c}$ and $\alpha, \varepsilon > 0$; a unitary $U \in \mathbb{C}^{d\times d}$ is an (α, ε)-block encoding of A if there exist $B_{L,1} \in \mathbb{C}^{d\times r}, B_{R,1} \in \mathbb{C}^{d\times c}$ with orthonormal columns s.t. $\|A - \alpha B_{L,1}^\dagger U B_{R,1}\|_{\mathrm{op}} \leq \varepsilon$. We denote $B_{L,1}^\dagger B_{L,1} = \Pi_L$ and $B_{R,1}^\dagger B_{R,1} = \Pi_R$, orthogonal projectors.

U, a block matrix, contains something ε-close to αA in its top left sub-block. Taking $(1, 0)$ block encoding, with $B_L = (B_{L,1}, B_{L,2})$ and $B_R = (B_{R,1}, B_{R,2})$ unitary completions of $B_{L,1}, B_{R,1}$:

$$B_L^\dagger U B_R = \begin{bmatrix} A & * \\ * & * \end{bmatrix}, \quad B_L^\dagger (\Pi_L U \Pi_R) B_R = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}.$$

An alternative definition is sometimes given, as in Def. 43 of [GSLW19], where a (α, a, ε)-block encoding of A satisfies

$$\|A - \alpha(\langle 0|^{\otimes a} \otimes I) U (|0\rangle^{\otimes a} \otimes I)\| \leq \varepsilon,$$

where A is an s-qubit operator, and U is an $(s + a)$ qubit unitary.

QSVT unitary; Def. 15 [GSLW19]

Let $\Phi = \{\phi_j\}_{j\in[n]} \in \mathbb{R}^n$; the QSVT protocol associated with Φ and a 2×2 block unitary U has circuit form (taking n even):

$$U_\Phi \equiv \prod_{j\in[n/2]} e^{i\phi_{2j-1}(2\Pi_R - I)} U^\dagger e^{i\phi_{2j}(2\Pi_L - I)} U.$$

QSVT unitary; Def. 15 [GSLW19]

Let $\Phi = \{\phi_j\}_{j\in[n]} \in \mathbb{R}^n$; the QSVT protocol associated with Φ and a 2×2 block unitary U has circuit form (taking n even):

$$U_\Phi \equiv \prod_{j\in[n/2]} e^{i\phi_{2j-1}(2\Pi_R - I)} U^\dagger e^{i\phi_{2j}(2\Pi_L - I)} U.$$

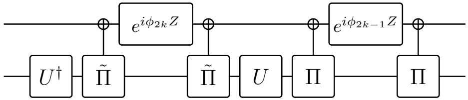
QSVT main theorem (informal)

Let $U \in \mathbb{C}^{d\times d}$ a block encoding of A, and let $\Phi \in \mathbb{R}^n$ such that its QSP protocol achieves $P(x) \in \mathbb{C}[x]$. Then (taking n even)

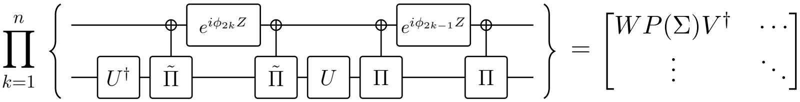
$$\Pi_R U_\Phi \Pi_R = \begin{bmatrix} P^{(SV)}(A) & 0 \\ 0 & 0 \end{bmatrix}.$$

In other words, within the block, he polynomial is the same as would have been applied by the QSP protocol for Φ.

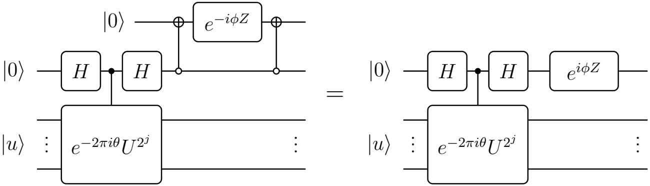
$$\prod_{k=1}^{n} \left\{ \right.$$



$$\left. \right\} = \begin{bmatrix} WP(\Sigma)V^\dagger & \cdots \\ \vdots & \ddots \end{bmatrix}$$



Note when image of projector is a single-qubit pure state, a trick allows for the direct recovery of simpler qubitization method.



🔑 ***The trick to making QSP useful***

The theory of QSP is basically ‘non-quantum‘; good for understanding, but how can we match quantum information processing tasks to its simple form?

🔑 *The trick to making QSP useful*

The theory of QSP is basically ‘non-quantum‘; good for understanding, but how can we match quantum information processing tasks to its simple form?

We’ll rely on a lifting argument, showing that interleaving large unitaries induces simple action in invariant subspaces.

🔑 ***The trick to making QSP useful***

The theory of QSP is basically 'non-quantum'; good for understanding, but how can we match quantum information processing tasks to its simple form?

We'll rely on a lifting argument, showing that interleaving large unitaries induces simple action in invariant subspaces.

This idea is not new, and appears in Grover search and QMA amplification [Gro05, Reg06]; the core observation has been known since 19^{th} century. [Jor75]

$$\frac{1}{\sqrt{N}}|m\rangle + \sqrt{\frac{N-1}{N}}|m^{\perp}\rangle \mapsto -\frac{1}{\sqrt{N}}|m\rangle + \sqrt{\frac{N-1}{N}}|m^{\perp}\rangle.$$

We can explicitly construct invariant subspaces. Let Π_R, Π_L, U, A as before, and k the largest index for which $\xi_k = 1$, where ξ_k is the k-th SV of A ordered by decreasing size, and $r = \text{rank}(A)$.

📌 **Recall:** $A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|$.

We can explicitly construct invariant subspaces. Let Π_R, Π_L, U, A as before, and k the largest index for which $\xi_k = 1$, where ξ_k is the k-th SV of A ordered by decreasing size, and $r = \mathrm{rank}(A)$.

📌 **Recall:** $A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|$.

$$\mathcal{H}_i = \mathrm{span}(|\psi_i\rangle), \qquad \tilde{\mathcal{H}}_i = \mathrm{span}(|\tilde{\psi}_i\rangle), \qquad i \in [k], \tag{1}$$

$$\mathcal{H}_i = \mathrm{span}(|\psi_i\rangle, |\psi_i^\perp\rangle), \qquad \tilde{\mathcal{H}}_i = \mathrm{span}(|\tilde{\psi}_i\rangle, |\tilde{\psi}_i^\perp\rangle), \qquad i \in [r]\backslash[k], \tag{2}$$

$$\mathcal{H}_i^R = \mathrm{span}(|\psi_i\rangle), \qquad \tilde{\mathcal{H}}_i^R = \mathrm{span}(U|\psi_i\rangle), \qquad i \in [d]\backslash[r], \tag{3}$$

$$\mathcal{H}_i^L = \mathrm{span}(U^\dagger|\tilde{\psi}_i\rangle), \qquad \tilde{\mathcal{H}}_i^L = \mathrm{span}(|\tilde{\psi}_i\rangle), \qquad i \in [\tilde{d}]\backslash[r]. \tag{4}$$

We can explicitly construct invariant subspaces. Let Π_R, Π_L, U, A as before, and k the largest index for which $\xi_k = 1$, where ξ_k is the k-th SV of A ordered by decreasing size, and $r = \text{rank}(A)$.

📌 **Recall:** $A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|$.

$$\mathcal{H}_i = \text{span}(|\psi_i\rangle), \qquad \tilde{\mathcal{H}}_i = \text{span}(|\tilde{\psi}_i\rangle), \qquad i \in [k], \tag{1}$$

$$\mathcal{H}_i = \text{span}(|\psi_i\rangle, |\psi_i^\perp\rangle), \qquad \tilde{\mathcal{H}}_i = \text{span}(|\tilde{\psi}_i\rangle, |\tilde{\psi}_i^\perp\rangle), \qquad i \in [r]\backslash[k], \tag{2}$$

$$\mathcal{H}_i^R = \text{span}(|\psi_i\rangle), \qquad \tilde{\mathcal{H}}_i^R = \text{span}(U|\psi_i\rangle), \qquad i \in [d]\backslash[r], \tag{3}$$

$$\mathcal{H}_i^L = \text{span}(U^\dagger|\tilde{\psi}_i\rangle), \qquad \tilde{\mathcal{H}}_i^L = \text{span}(|\tilde{\psi}_i\rangle), \qquad i \in [\tilde{d}]\backslash[r]. \tag{4}$$

Here $d = \text{rank}(\Pi_R)$, $\tilde{d} = \text{rank}(\Pi_L)$, and $|\psi_i\rangle$ and $|\tilde{\psi}_i\rangle$ are the right and left SVecs of A; i.e., orthonormal bases for $(img)(\Pi_R)$ and $\text{img}(\Pi_L)$. The $(\perp)$ superscript follows:

$$|\psi_i^\perp\rangle \equiv (\sqrt{1 - \xi_i^2})^{-1}(I - \Pi_R)U^\dagger|\tilde{\psi}_i\rangle, \tag{5}$$

$$|\tilde{\psi}_i^\perp\rangle \equiv (\sqrt{1 - \xi_i^2})^{-1}(I - \Pi_L)U|\psi_i\rangle. \tag{6}$$

Qubitization shows that $\mathcal{H}_i$, $\mathcal{H}_i^R$, and $\mathcal{H}_i^L$ are pairwise orthogonal, which we summarize and motivate:

Qubitization shows that $\mathcal{H}_i$, $\mathcal{H}_i^R$, and $\mathcal{H}_i^L$ are pairwise orthogonal, which we summarize and motivate:

$$\langle\psi_i|\psi_j\rangle = \delta_{ij}, \quad \mathcal{H}_i \perp \mathcal{H}_j, \tag{7}$$

$$\langle\tilde{\psi}_i|\tilde{\psi}_j\rangle = \delta_{ij}, \quad \tilde{\mathcal{H}}_i \perp \tilde{\mathcal{H}}_j, \tag{8}$$

$$\langle\psi_i^\perp|\psi_j^\perp\rangle = \langle\tilde{\psi}_i^\perp|\tilde{\psi}_j^\perp\rangle = \delta_{ij}, \quad (\mathcal{H}_i/\mathcal{H}_i)^\perp \perp (\tilde{\mathcal{H}}_j/\tilde{\mathcal{H}}_j)^\perp, \tag{9}$$

$$\langle\psi_i|\psi_j^\perp\rangle = \langle\tilde{\psi}_i|\tilde{\psi}_j^\perp\rangle = 0, \quad */* \perp */*, \tag{10}$$

$$\langle\psi_i|U^\dagger|\tilde{\psi}_j\rangle = 0, \quad U^\dagger|\tilde{\psi}_j\rangle \in \mathcal{H}_j^L, \tag{11}$$

$$\langle\psi_i^\perp|U^\dagger|\tilde{\psi}_j\rangle = 0, \quad U^\dagger|\tilde{\psi}_j\rangle \in \mathcal{H}_j^L. \tag{12}$$

Qubitization shows that $\mathcal{H}_i$, $\mathcal{H}_i^R$, and $\mathcal{H}_i^L$ are pairwise orthogonal, which we summarize and motivate:

$$\langle\psi_i|\psi_j\rangle = \delta_{ij}, \quad \mathcal{H}_i \perp \mathcal{H}_j, \tag{7}$$

$$\langle\tilde{\psi}_i|\tilde{\psi}_j\rangle = \delta_{ij}, \quad \tilde{\mathcal{H}}_i \perp \tilde{\mathcal{H}}_j, \tag{8}$$

$$\langle\psi_i^\perp|\psi_j^\perp\rangle = \langle\tilde{\psi}_i^\perp|\tilde{\psi}_j^\perp\rangle = \delta_{ij}, \quad (\mathcal{H}_i/\mathcal{H}_i)^\perp \perp (\tilde{\mathcal{H}}_j/\tilde{\mathcal{H}}_j)^\perp, \tag{9}$$

$$\langle\psi_i|\psi_j^\perp\rangle = \langle\tilde{\psi}_i|\tilde{\psi}_j^\perp\rangle = 0, \quad */* \perp */*, \tag{10}$$

$$\langle\psi_i|U^\dagger|\tilde{\psi}_j\rangle = 0, \quad U^\dagger|\tilde{\psi}_j\rangle \in \mathcal{H}_j^L, \tag{11}$$

$$\langle\psi_i^\perp|U^\dagger|\tilde{\psi}_j\rangle = 0, \quad U^\dagger|\tilde{\psi}_j\rangle \in \mathcal{H}_j^L. \tag{12}$$

The first three follow from the orthogonality of singular vectors; note that $\langle\tilde{\psi}_i|U\Pi_R U^\dagger|\psi_j\rangle$ can be replaced by $\langle\tilde{\psi}_i|AA^\dagger|\psi_j\rangle$ freely. The action of U is to take all $|\psi_i\rangle$ to their corresponding $|\tilde{\psi}_j\rangle$ vectors, and Π_R, Π_L project onto the span of the tilde and non-tilde orthogonal bases. The final three identities follow from the action of the projectors on vectors not in their images.

In ***qubitization***, large unitary breaks into direct sum of qubit-like maps. Brackets indicate map from *superscript to the subscript*:

$$U = \bigoplus_{i\in[k]} [\xi_i]^{\mathcal{H}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} \xi_i & \sqrt{1-\xi_i^2} \\ \sqrt{1-\xi_i^2} & \xi_i \end{bmatrix}^{\mathcal{H}_i}_{\tilde{\mathcal{H}}_i} \oplus [1]^{\mathcal{H}_i^R \oplus \mathcal{H}_i^L}_{\tilde{\mathcal{H}}_i^R \oplus \tilde{\mathcal{H}}_i^L} \oplus [*]^{\mathcal{H}^\perp}_{\tilde{\mathcal{H}}^\perp}, \tag{13}$$

$$e^{i\phi(2\Pi_R - I)} = \bigoplus_{i\in[k]} [e^{i\phi}]^{\mathcal{H}_i}_{\mathcal{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}^{\mathcal{H}_i}_{\mathcal{H}_i} \oplus [e^{i\phi}]^{\mathcal{H}_i^R}_{\mathcal{H}_i^R} \oplus [e^{-i\phi}]^{\mathcal{H}_i^L}_{\mathcal{H}_i^L} \oplus [*]^{\mathcal{H}^\perp}_{\mathcal{H}^\perp}, \tag{14}$$

$$e^{i\phi(2\Pi_L - I)} = \bigoplus_{i\in[k]} [e^{i\phi}]^{\tilde{\mathcal{H}}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}^{\tilde{\mathcal{H}}_i}_{\tilde{\mathcal{H}}_i} \oplus [e^{-i\phi}]^{\tilde{\mathcal{H}}_i^R}_{\tilde{\mathcal{H}}_i^R} \oplus [e^{i\phi}]^{\tilde{\mathcal{H}}_i^L}_{\tilde{\mathcal{H}}_i^L} \oplus [*]^{\tilde{\mathcal{H}}^\perp}_{\tilde{\mathcal{H}}^\perp}, \tag{15}$$

In ***qubitization***, large unitary breaks into direct sum of qubit-like maps. Brackets indicate map from *superscript to the subscript*:

$$U = \bigoplus_{i\in[k]} [\xi_i]^{\mathcal{H}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} \xi_i & \sqrt{1-\xi_i^2} \\ \sqrt{1-\xi_i^2} & \xi_i \end{bmatrix}^{\mathcal{H}_i}_{\tilde{\mathcal{H}}_i} \oplus [1]^{\mathcal{H}_i^R \oplus \mathcal{H}_i^L}_{\tilde{\mathcal{H}}_i^R \oplus \tilde{\mathcal{H}}_i^L} \oplus [*]^{\mathcal{H}^\perp}_{\tilde{\mathcal{H}}^\perp}, \tag{13}$$

$$e^{i\phi(2\Pi_R - I)} = \bigoplus_{i\in[k]} [e^{i\phi}]^{\mathcal{H}_i}_{\mathcal{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}^{\mathcal{H}_i}_{\mathcal{H}_i} \oplus [e^{i\phi}]^{\mathcal{H}_i^R}_{\mathcal{H}_i^R} \oplus [e^{-i\phi}]^{\mathcal{H}_i^L}_{\mathcal{H}_i^L} \oplus [*]^{\mathcal{H}^\perp}_{\mathcal{H}^\perp}, \tag{14}$$

$$e^{i\phi(2\Pi_L - I)} = \bigoplus_{i\in[k]} [e^{i\phi}]^{\tilde{\mathcal{H}}_i}_{\tilde{\mathcal{H}}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}^{\tilde{\mathcal{H}}_i}_{\tilde{\mathcal{H}}_i} \oplus [e^{-i\phi}]^{\tilde{\mathcal{H}}_i^R}_{\tilde{\mathcal{H}}_i^R} \oplus [e^{i\phi}]^{\tilde{\mathcal{H}}_i^L}_{\tilde{\mathcal{H}}_i^L} \oplus [*]^{\tilde{\mathcal{H}}^\perp}_{\tilde{\mathcal{H}}^\perp}, \tag{15}$$

This can be explicitly verified from the known relations among Π_L, Π_R, U, A. Important subspaces are non-trivial $\mathcal{H}_i, \tilde{\mathcal{H}}_i$.

In some sense, nothing besides U(2) operations could have happened in these subspaces! And this imposes constraints!

Alternatively, the ***cosine-sine decomposition*** arises for unitary U with 2×2 block form. It turns out one can produce *simultaneous* SVDs satisfying $U_{ij} = V_i D_{ij} W_j^\dagger$:

Alternatively, the ***cosine-sine decomposition*** arises for unitary U with 2×2 block form. It turns out one can produce *simultaneous* SVDs satisfying $U_{ij} = V_i D_{ij} W_j^\dagger$:

Cosine-sine decomposition (CSD) statement

Let $U \in \mathbb{C}^{d \times d}$ a unitary matrix partitioned into blocks of size $\{r_1, r_2\} \times \{c_1, c_2\}$:

$$\begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}, \quad \text{where } U_{ij} \in \mathbb{C}^{r_i \times c_j},$$

Then there exist unitaries $V_i \in \mathbb{C}^{r_i \times r_i}$ and $W_j \in \mathbb{C}^{c_j \times c_j}$ such that

$$\begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} = \begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix} \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} W_1 & \\ & W_2 \end{bmatrix}^\dagger,$$

where blanks are the zero matrix, and each D_{ij} is diagonal in $\mathbb{C}^{r_i \times c_j}$, possibly padded with zeros.

Specifically, we can write D in the form:

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} = \left[\begin{array}{ccc|ccc} 0 & & & I & & \\ & C & & & S & \\ & & I & & & 0 \\ \hline I & & & 0 & & \\ & S & & & -C & \\ & & 0 & & & -I \end{array}\right],$$

$$= \underbrace{\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}}_{\mathcal{X}_0 \to \mathcal{Y}_0} \oplus \underbrace{\begin{bmatrix} C & S \\ S & -C \end{bmatrix}}_{\mathcal{X}_C \to \mathcal{Y}_C} \oplus \underbrace{\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}}_{\mathcal{X}_1 \to \mathcal{Y}_1}.$$

where C, S, I are square diagonal matrices, and C, S have entries in the interval $(0, 1)$ on their diagonal, and $C^2 + S^2 = I$.

💡 *Idea of the proof of CSD*

(1) Start with the SVD of $U_{11} = V_1 D_{11} W_1^\dagger$, noting SVs in $[0, 1]$.

(2) Compute QR decompositions of $U_{21} W_1$ and $U_{12}^\dagger V_1$, which give V_2, W_2 to make these operators upper-diagonal with nonnegative diagonal entries:

$$\begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix}^\dagger \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} W_1 & \\ & W_2 \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & V_2^\dagger U_{22} W_2 \end{bmatrix}.$$

(3) Observing the rest of the overall unitary (whose rows and columns must be orthonormal), this forces the entries of D_{12}, D_{21} to satisfy the desired form: $C^2 + S^2 = I$.

(4) Finally, $W_2 \mapsto W_2'$ to correct D_{22} (free up to unitary).

Part II: QSP and functional analysis

After lifting, what's next?

Given reduction to QSP, understanding possible unitaries follows from understanding possible polynomials.

Usually want to control one SU(2) matrix element, leading to a *completion problem*: for poly $P(x)$, does there exist $Q(x)$ s.t.

$$\begin{bmatrix} P(x) & i\sqrt{1-x^2}Q(x) \\ i\sqrt{1-x^2}Q^*(x) & P^*(x) \end{bmatrix} \in \mathrm{SU}(2)\ ?$$

After lifting, what's next?

Given reduction to QSP, understanding possible unitaries follows from understanding possible polynomials.

Usually want to control one SU(2) matrix element, leading to a *completion problem*: for poly $P(x)$, does there exist $Q(x)$ s.t.

$$\begin{bmatrix} P(x) & i\sqrt{1-x^2}Q(x) \\ i\sqrt{1-x^2}Q^*(x) & P^*(x) \end{bmatrix} \in \mathrm{SU}(2)\ ?$$

This is only part of the story, but simplifies choosing $P(x)$; then

$$P(x) \rightarrow P(x), Q(x) \rightarrow \Phi \in \mathbb{R}^{n+1}.$$

For standard QSP, completion is equivalent to phase existence, and relies on simple fact of positive trigonometric polynomials.

⚙️ *Standard proof techniques in QSP*

Completion arguments rely on Fejér-Riesz lemma [PS98], which shows nonnegative trigonometric polynomials are squares. Proof follows from simple root analysis/pairing.

$$P(x) \geq 0 \text{ on } [-1, 1] \implies P(x) = |B(x)|^2 + (1 - x^2)|C(x)|^2.$$

Showing equivalence to existence of Φ follows by induction, finding ϕ s.t. $U(\Phi, x) = e^{i\phi\sigma_z} U(\Phi', x)$ with $|\Phi'| = |\Phi| - 1$.

⚙️ *Standard proof techniques in QSP*

Completion arguments rely on Fejér-Riesz lemma [PS98], which shows nonnegative trigonometric polynomials are squares. Proof follows from simple root analysis/pairing.

$$P(x) \geq 0 \text{ on } [-1,1] \implies P(x) = |B(x)|^2 + (1-x^2)|C(x)|^2.$$

Showing equivalence to existence of Φ follows by induction, finding ϕ s.t. $U(\Phi, x) = e^{i\phi\sigma_z}U(\Phi', x)$ with $|\Phi'| = |\Phi| - 1$.

⚠️ In infinite-length [DLNW22], multivariable [RC22], or nonlinear Fourier analysis [ALM$^+$24] setting, equivalent statements require careful algebraic geometric analysis.

💻 ***Classical algorithms paired with QSP***

Phase-finding methods for QSP; the ultimate goal is numerical stability, where the number of bits of precision required goes as $\log[1/\varepsilon]$ in desired fidelity.

(a) Initial, unstable, factorization-based, followed by iterative phase read-off; good to $n \approx 100$ [YLC14, LYC16]

(b) Laurent polynomial and Fourier methods; good to $n \approx 10^3$, though not using standard double precision. [Haa19]

(c) Optimization-based, iterative methods for restricted ansätze; good approx $n \approx 10^7$. [WDL21, DMWL21, AMT23, ALM$^+$24]

🛠️ *Numerical methods for QSP*

Current leading methods for phase-finding are iterative, Newton's method-like, and rely on symmetrizing ansatz. [DMWL21]

$$\|\Phi - \Phi'\|_{\infty} \leq C\eta^{-3}\|f - f'\|_{S}, \quad \|f\|_{\infty} \leq 1 - \eta.$$

Proof of convergence analyzes QSP Jacobian, shown to be Lipschitz continuous, and guaranteed not just for bounded ℓ_1-norm targets, but bounded ℓ_∞ targets! [ALM+24]

🛠️ *Numerical methods for QSP*

Current leading methods for phase-finding are iterative, Newton's method-like, and rely on symmetrizing ansatz. [DMWL21]

$$\|\Phi - \Phi'\|_\infty \leq C\eta^{-3}\|f - f'\|_S, \quad \|f\|_\infty \leq 1 - \eta.$$

Proof of convergence analyzes QSP Jacobian, shown to be Lipschitz continuous, and guaranteed not just for bounded ℓ_1-norm targets, but bounded ℓ_∞ targets! [ALM$^+$24]

Actively-developed numerical packages:

MATLB-based from Lin group:
✨ `QSPPACK: https://github.com/qsppack/QSPPACK`,
and Python-based from Chuang group:
✨ `pyQSP: https://github.com/ichuang/pyqsp`.

Part III: Applications and extensions

‼️ Applications: QSP is all* you need

‼️ Applications: QSP is all* you need

QSP and quantum singular value transformation (QSVT) compute ***matrix functions*** for large* linear operators [GSLW19].

$$A = \sum_k \xi_k |\tilde{\psi}_k\rangle\langle\psi_k| \xrightarrow[\text{QSVT}]{} \sum_k P(\xi_k)|\tilde{\psi}_k\rangle\langle\psi_k| = P(A)$$

‼️ Applications: QSP is all* you need

QSP and quantum singular value transformation (QSVT) compute ***matrix functions*** for large* linear operators [GSLW19].

$$A = \sum_k \xi_k |\tilde{\psi}_k\rangle\langle\psi_k| \underset{\text{QSVT}}{\longmapsto} \sum_k P(\xi_k)|\tilde{\psi}_k\rangle\langle\psi_k| = P(A)$$

Search: Input Grover oracle, apply constant function

Low energy proj: Input Hamiltonian, apply bandpass function

Inversion: Input sparse linear sys, apply $1/x$ approximation

Simulation: Input Hamiltonian, apply trigonometric function ...

‼️ Applications: QSP is all* you need

QSP and quantum singular value transformation (QSVT) compute ***matrix functions*** for large* linear operators [GSLW19].

$$A = \sum_k \xi_k |\tilde{\psi}_k\rangle\langle\psi_k| \underset{\text{QSVT}}{\longmapsto} \sum_k P(\xi_k)|\tilde{\psi}_k\rangle\langle\psi_k| = P(A)$$

Search: Input Grover oracle, apply constant function

Low energy proj: Input Hamiltonian, apply bandpass function

Inversion: Input sparse linear sys, apply $1/x$ approximation

Simulation: Input Hamiltonian, apply trigonometric function ...

Changing the polynomial changes the algorithm

🔨 ***Guidelines and standard applications for QSVT***

Question(s)

QSVT can do similar things to other quantum algorithms, so when should we use it? What are its strong attributes?

🔨 *Guidelines and standard applications for QSVT*

Question(s)

QSVT can do similar things to other quantum algorithms, so when should we use it? What are its strong attributes?

Input promises: For low-space phase-estimation, QSVT incredibly tuneable given promises on eigenvalue distribution. [Ral21]

State preparation: When approximating entire functions, e.g., exponential for Gibbs states, or trigonometric functions for simulation, smoothness guarantees exponential convergence. [GSLW19, GLM$^+$22]

Deep, coherent circuits: QSVT has constant space overhead, with success scaling as ℓ_∞ norm, as opposed to LCU, with logarithmic space and ℓ_1-norm scaling.

✨ ***Recent generalizations and extensions***

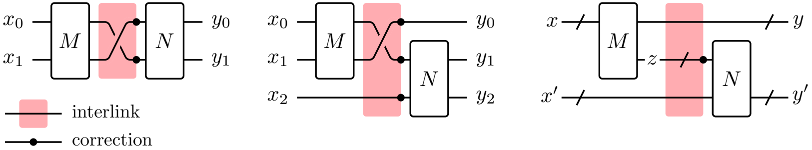
Restricting [WDL21] or expanding [MW23] circuit ansatz can improve numerical properties and flexibility of achieved transform.

Multivariable variants [RC22, RC23, BWSS23, GLW24] can compute joint functions and make bosonic simulation simpler.

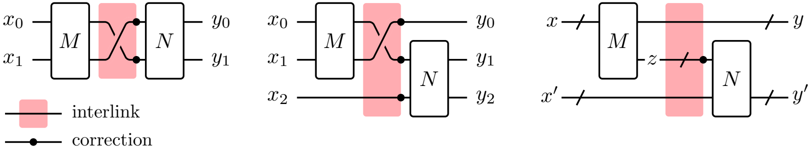
QSVT can be modularly composed [RCC23, MF23, GLW24] in a functional way, simplifying protocol design.

The theory of nonlinear Fourier analysis captures behavior of QSP [AMT23, ALM^{+}24], and furnishes convergence proofs for phase-finding algorithms.

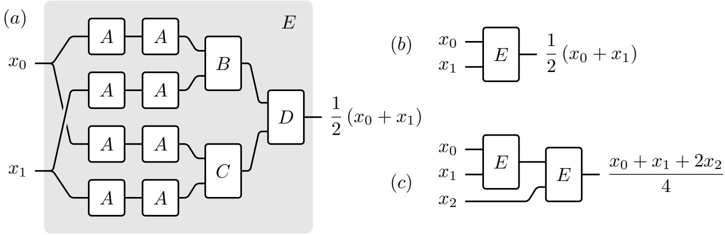
E.g., QSP-like modules can be combined, **if we can enforce** QSP-like behavior in special subspaces:



E.g., QSP-like modules can be combined, **if we can enforce** QSP-like behavior in special subspaces:



Once these properties have been (approximately) established, algorithm design can be usefully abstracted:



🔗 *QSP/M-QSP: permitted block encoding functionals*

Exposited in [RCC23, GLW24, MF23].

	Exact	Approx	Query comp	Norm scale
$\mathbb{Q}$-power	❌	✅	$\delta^{-1} \log \varepsilon^{-1}$	$\lVert * \rVert_\infty$
Inversion	❌	✅	$\delta^{-1} \log \varepsilon^{-1}$	$\lVert * \rVert_\infty$
Composition	⚠️ †	✅	$d_1 d_2 \log \varepsilon^{-1}$	$\lVert * \rVert_\infty$
Sum	⚠️	✅	$(d_1 + d_2) \log \varepsilon^{-1}$	$\lVert * \rVert_\infty$
Product	⚠️	✅	$d_1 d_2 \log \varepsilon^{-1}$	$\lVert * \rVert_\infty$

† Here ⚠️ means exact for non-trivial strict subsets of possible polynomials of degree d_1, d_2.
Complexity and norm scaling are given for approximative methods for $x \in [-1 + \delta, -\delta] \cup [\delta, 1 - \delta]$.

🔗 QSP/M-QSP: permitted block encoding functionals

Exposited in [RCC23, GLW24, MF23].

	Exact	Approx	Query comp	Norm scale
$\mathbb{Q}$-power	❌	✅	$\delta^{-1} \log \varepsilon^{-1}$	$\|*\|_\infty$
Inversion	❌	✅	$\delta^{-1} \log \varepsilon^{-1}$	$\|*\|_\infty$
Composition	⚠️ †	✅	$d_1 d_2 \log \varepsilon^{-1}$	$\|*\|_\infty$
Sum	⚠️	✅	$(d_1 + d_2) \log \varepsilon^{-1}$	$\|*\|_\infty$
Product	⚠️	✅	$d_1 d_2 \log \varepsilon^{-1}$	$\|*\|_\infty$

📌 In comparison, linear combination of unitaries (LCU) [CW12] (1) depends on $\|*\|_1$, (2) uses logarithmic not constant additional space and (3) can exhibit quadratically worse query complexity.*

† Here ⚠️ means exact for non-trivial strict subsets of possible polynomials of degree d_1, d_2.
Complexity and norm scaling are given for approximative methods for $x \in [-1+\delta, -\delta] \cup [\delta, 1-\delta]$.

✨ *Work challenging input assumptions and ansatz form*

Parallelized QSP [MRC+24] can trade-off circuit depth for width.

Randomized QSP [MR24] can lower circuit depth.

Classical feedback-based QSP [DAN24] for calibration tasks.

Higher-order tasks (rational powers, inversion, composition, sums/products) are sensitive to resource model and target (approximate, non-deterministic, etc.). [RCC23]

✨ ***Work challenging input assumptions and ansatz form***

Parallelized QSP [MRC$^+$24] can trade-off circuit depth for width.

Randomized QSP [MR24] can lower circuit depth.

Classical feedback-based QSP [DAN24] for calibration tasks.

Higher-order tasks (rational powers, inversion, composition, sums/products) are sensitive to resource model and target (approximate, non-deterministic, etc.). [RCC23]

Looking ahead

Applying QSP/QSVT to different resource models requires suitably weakening lifting argument, modifying completion argument, and applying new approximation techniques. ***How do we flesh-out a fuzzy, functional model of QSP/QSVT-like algorithms?***

[ALM+24] Michel Alexis, Lin Lin, Gevorg Mnatsakanyan, Christoph Thiele, and Jiasu Wang. Infinite quantum signal processing for arbitrary szegö functions, 2024.

[AMT23] Michel Alexis, Gevorg Mnatsakanyan, and Christoph Thiele. Quantum signal processing and nonlinear fourier analysis, 2023.

[BWSS23] Yonah Borns-Weil, Tahsin Saffat, and Zachary Stier. A quantum algorithm for functions of multiple commuting Hermitian matrices. *arXiv preprint, arXiv:2302.11139*, 2023.

[CW12] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Info. Comput.*, 12(11–12):901–924, Nov 2012.

[DAN24] Yulong Dong, Dong An, and Murphy Yuezhen Niu. Feedforward quantum singular value transformation, 2024.

[DLNW22] Yulong Dong, Lin Lin, Hongkang Ni, and Jiasu Wang. Infinite quantum signal processing. *arXiv preprint, arXiv:2209.10162*, 2022.

[DMWL21] Yulong Dong, Xiang Meng, K. Birgitta Whaley, and Lin Lin. Efficient phase-factor evaluation in quantum signal processing. *Phys. Rev. A*, 103(4), Apr 2021.

[GLM+22] András Gilyén, Seth Lloyd, Iman Marvian, Yihui Quek, and Mark M Wilde. Quantum algorithm for Petz recovery channels and pretty good measurements. *Phys. Rev. Lett.*, 128(22):220502, 2022.

[GLW24] Niladri Gomes, Hokiat Lim, and Nathan Wiebe. Multivariable qsp and bosonic quantum simulation using iterated quantum signal processing, 2024.

[Gro05] Lov K Grover. Fixed-point quantum search. *Phys. Rev. Lett.*, 95(15):150501, 2005.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019.

[Haa19] Jeongwan Haah. Product decomposition of periodic functions in quantum signal processing. *Quantum*, 3:190, Oct 2019.

[Jor75] Camille Jordan. Essai sur la géométrie à n dimensions. *Bulletin de la Société mathématique de France*, 3:103–174, 1875.

[LC19] G. H. Low and I. L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, Jul 2019.

[LYC16] G. H. Low, T. J. Yoder, and I. L. Chuang. Methodology of resonant equiangular composite quantum gates. *Phys. Rev. X*, 6:041067, 2016.

[MF23] Kaoru Mizuta and Keisuke Fujii. Recursive quantum eigenvalue/singular-value transformation: Analytic construction of matrix sign function by Newton iteration. *arXiv preprint, arXiv:2304.13330*, 2023.

[MR24] John M. Martyn and Patrick Rall. Halving the cost of quantum algorithms with randomization, 2024.

[MRC+24] John M. Martyn, Zane M. Rossi, Kevin Z. Cheng, Yuan Liu, and Isaac L. Chuang. Parallel quantum signal processing via polynomial factorization, 2024.

[MRTC21] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4), dec 2021.

[MW23] Danial Motlagh and Nathan Wiebe. Generalized quantum signal processing. *arXiv preprint, arXiv:2308.01501*, 2023.

[PS98] George Polya and Gabor Szegö. *Problems and Theorems in Analysis II*. Springer, Berlin, 1998.

[Ral21] Patrick Rall. Faster coherent quantum algorithms for phase, energy, and amplitude estimation. *Quantum*, 5:566, Oct 2021.

[RC22] Zane M. Rossi and Isaac L. Chuang. Multivariable quantum signal processing (M-QSP): prophecies of the two-headed oracle. *Quantum*, 6:811, Sep 2022.

[RC23] Zane M. Rossi and Isaac L. Chuang. Semantic embedding for quantum algorithms. *arXiv preprint, arXiv:2304.14392*, 2023.

[RCC23] Zane M. Rossi, Jack L. Ceroni, and Isaac L. Chuang. Modular quantum signal processing in many variables. *arXiv preprint arXiv:2309.16665*, 2023.

[Reg06] Oded Regev. Fast amplification of QMA (lecture notes), 2006. https://cims.nyu.edu/~regev/teaching/quantum_fall_2005/ln/qma.pdf.

[TT23] Ewin Tang and Kevin Tian. A CS guide to the quantum singular value transformation. *arXiv preprint, arXiv:2302.14324*, 2023.

[WDL21] Jiasu Wang, Yulong Dong, and Lin Lin. On the energy landscape of symmetric quantum signal processing. *arXiv preprint, arXiv:2110.04993*, 2021.

[YLC14] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. Fixed-point quantum search with an optimal number of queries. *Phys. Rev. Lett.*, 113(21):210501, 2014.

✨ Backup slides

✨ On the optimality of QSP/QSVT

Lower bound for eig. transformation; Thm. 73 [GSLW19]
Let $I \subseteq [-1,1]$, $a \geq 1$ and suppose U is a $(1,a,0)$-block encoding of an unknown Hermitian matrix H with the promise that the spectrum of H lies within I. Let $f : I \to \mathbb{R}$, and suppose access to a quantum circuit V that implements a $(1,b,\varepsilon)$-block encoding of $f(H)$ using T applications of U for all U satisfying the promise. Then for all $x \neq y \in I \cap [-1/2, 1/2]$ we have that

$$T = \Omega\left[\frac{|f(x) - f(y)| - 2\varepsilon}{|x-y|}\right]$$

Lower bound for quantum matrix functions; from [MS24]
For any continuous function $f(x) : [-1,1] \to [-1,1]$, there is a 2-sparse Hermitian matrix A with $|A| \leq 1$ and two indices i,j such that $\Omega(\widetilde{\deg}_\varepsilon(f))$ queries to A are required in order to compute $\langle i|f(A)|j\rangle \pm \varepsilon/4$.