Modern techniques in quantum algorithms

QTML 2025 Tutorial

Zane Marius Rossi

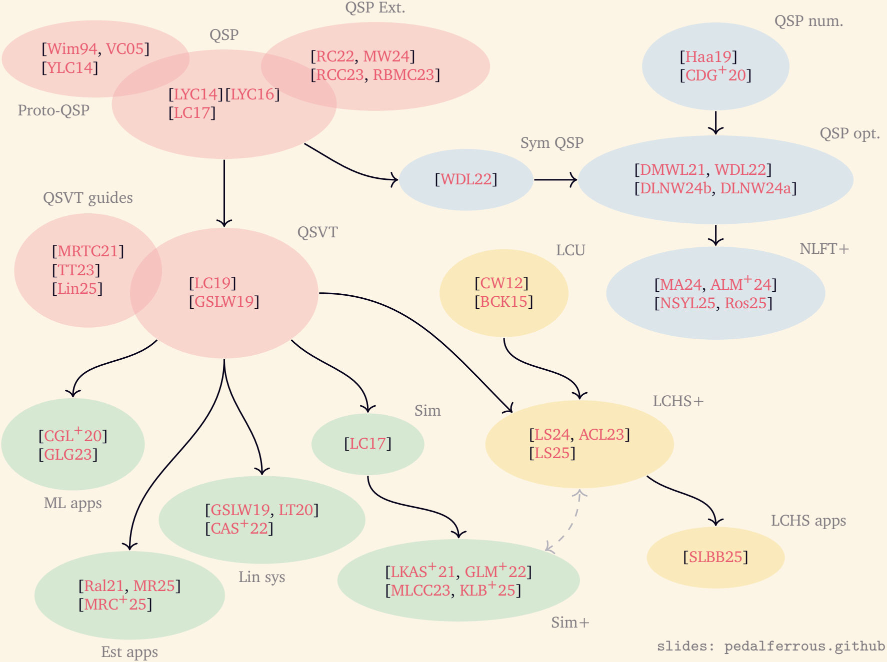
The University of Tokyo

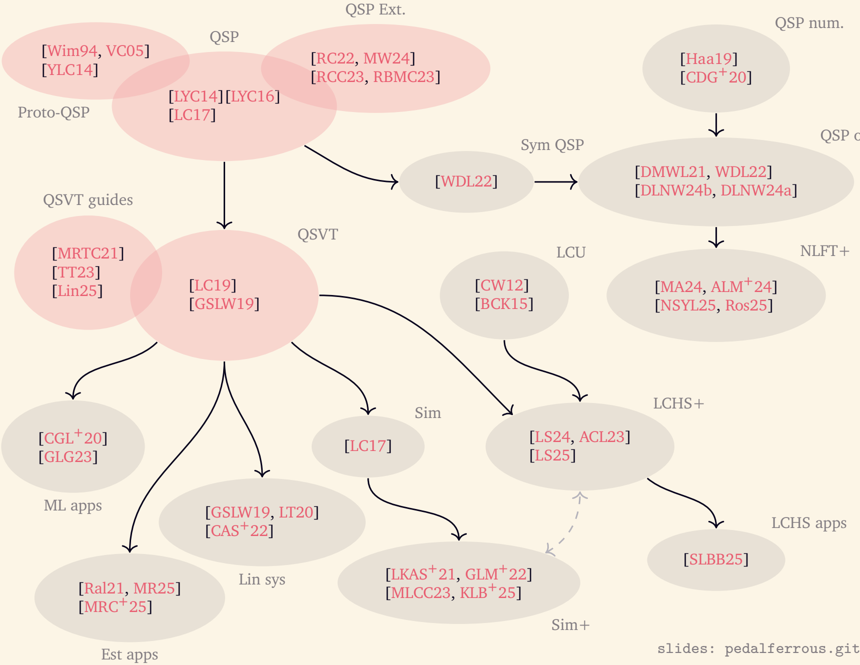
November 16, 2025

🎯 Today's aims

1. Motivation and crash-course for QSP[1] ★
2. Numerical interlude ★
3. Qubitization, block encodings, and QSVT[2] ★
4. Applications and the state of the art ★
5. The limits of QSVT, and alternative approaches ★
6. Open problems, recent progress, and outlook ★★★★

[1] Quantum signal processing
[2] Quantum singular value transformation


QSP Ext.
[RC22, MW24]
[RCC23, RBMC23]
QSP
[LYC14][LYC16]
[LC17]
[Wim94, VC05]
[YLC14]
Proto-QSP
QSP num.
[Haa19]
[CDG+20]
QSP opt.
[DMWL21, WDL22]
[DLNW24b, DLNW24a]
Sym QSP
[WDL22]
QSVT guides
[MRTC21]
[TT23]
[Lin25]
QSVT
[LC19]
[GSLW19]
LCU
[CW12]
[BCK15]
NLFT+
[MA24, ALM+24]
[NSYL25, Ros25]
[CGL+20]
[GLG23]
ML apps
Sim
[LC17]
LCHS+
[LS24, ACL23]
[LS25]
[GSLW19, LT20]
[CAS+22]
Lin sys
[Ral21, MR25]
[MRC+25]
Est apps
[LKAS+21, GLM+22]
[MLCC23, KLB+25]
Sim+
LCHS apps
[SLBB25]



slides: pedalferrous.github.io

⚛️ NMR and composite pulses

⚛️ NMR and composite pulses

What can we do with many copies of an ***unknown but consistent*** unitary process?[1] [Wim94, VC05, BHC04]

[1]E.g., Larmor precession in inhomogeneous magnetic field

⚛️ NMR and composite pulses

What can we do with many copies of an ***unknown but consistent*** unitary process?[1] [Wim94, VC05, BHC04]

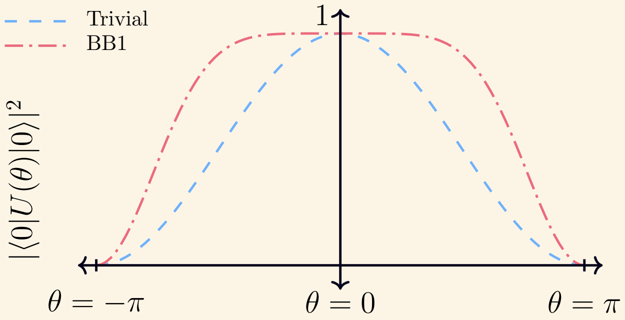
$$V(\theta), V(\theta), \ldots, V(\theta) \mapsto U(\theta)$$

[1] E.g., Larmor precession in inhomogeneous magnetic field

⚛️ NMR and composite pulses

What can we do with many copies of an ***unknown but consistent*** unitary process?[1] [Wim94, VC05, BHC04]

$$V(\theta), V(\theta), \ldots, V(\theta) \mapsto U(\theta)$$



[1] E.g., Larmor precession in inhomogeneous magnetic field

QSP is a tunable SU(2)-valued function

QSP is a tunable SU(2)-valued function

Single-qubit alternating circuit ansatz

$$\Phi \in \mathbb{R}^{n+1} \mapsto U_{\Phi}(x).$$

QSP is a tunable SU(2)-valued function

Single-qubit alternating circuit ansatz

$$\Phi \in \mathbb{R}^{n+1} \mapsto U_{\Phi}(x).$$

Unitary oracle is **structured**: $W(x) = e^{i\cos^{-1}(x)\sigma_x}$

QSP is a tunable SU(2)-valued function

Single-qubit alternating circuit ansatz

$$\Phi \in \mathbb{R}^{n+1} \mapsto U_{\Phi}(x).$$

Unitary oracle is **structured**: $W(x) = e^{i\cos^{-1}(x)\sigma_x}$

We program real **phases** $\Phi = \{\phi_0, \ldots, \phi_n\}$ to condition output unitary on **signal** x in a precise way

QSP is a tunable SU(2)-valued function

Single-qubit alternating circuit ansatz

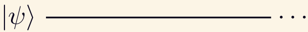
$$\Phi \in \mathbb{R}^{n+1} \mapsto U_{\Phi}(x).$$

Unitary oracle is **structured**: $W(x) = e^{i\cos^{-1}(x)\sigma_x}$

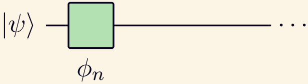
We program real **phases** $\Phi = \{\phi_0, \ldots, \phi_n\}$ to condition output unitary on **signal** x in a precise way

$$W(x) = \begin{bmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{bmatrix}, \qquad e^{i\phi} = \begin{bmatrix} e^{i\phi} & \\ & e^{-i\phi} \end{bmatrix}.$$

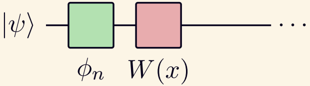
Properties of the QSP ansatz



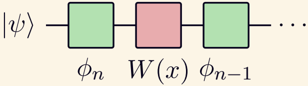
Properties of the QSP ansatz



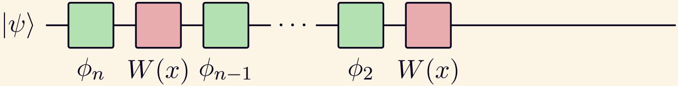
Properties of the QSP ansatz



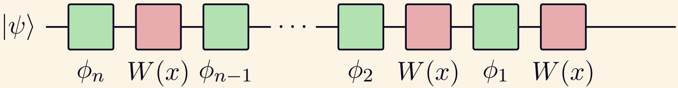
Properties of the QSP ansatz



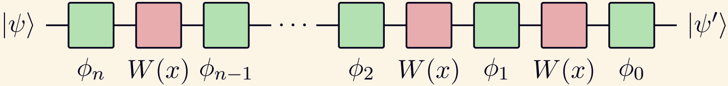
Properties of the QSP ansatz



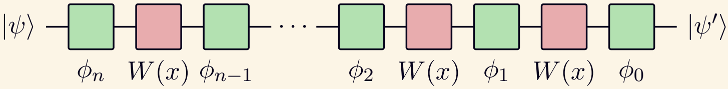
Properties of the QSP ansatz



Properties of the QSP ansatz



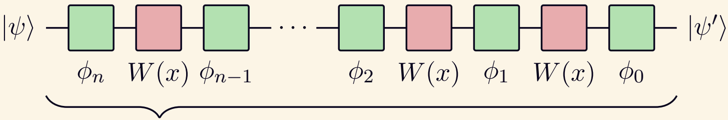
Properties of the QSP ansatz



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

Can map $P \leftrightarrow \Phi$ efficiently; like classical filter!

Properties of the QSP ansatz

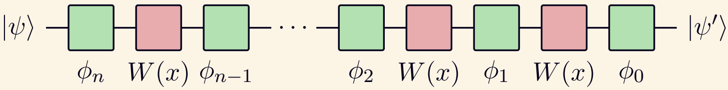


$$U(\Phi, H) = e^{i\phi_0 Z}\prod_{j=1}^{k} W(H)\, e^{i\phi_j Z} = \begin{array}{cc} \begin{bmatrix} P(H) & iQ(H)\sqrt{1-H^2} \\ iQ(H)^*\sqrt{1-H^2} & P(H)^* \end{bmatrix} & \begin{array}{c} |0\rangle \\ |1\rangle \end{array} \\ \begin{array}{cc} \quad |0\rangle \quad & \quad\quad |1\rangle \quad \end{array}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

Can map $P \leftrightarrow \Phi$ efficiently; like classical filter!

Claim: can replace $x \in \mathbb{R}$ with normal H, and more! I.e., a block-matrix form of $U_\Phi(H)$.

Properties of the QSP ansatz



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

Can map $P \leftrightarrow \Phi$ efficiently; like classical filter!

Claim: can replace $x \in \mathbb{R}$ with normal H, and more! I.e., a block-matrix form of $U_\Phi(H)$.

Later: the theory of *block encodings*!

⚛️ The BBl Protocol[1] as QSP

[1] [Wim94]

⚛️ The BB1 Protocol[1] as QSP

$$U(\theta) = e^{i\pi\sigma_z/2} V(\theta) e^{-i\eta\sigma_z} V(\theta) e^{2i\eta\sigma_z} V^2(\theta) e^{-2i\eta\sigma_z} V(\theta) e^{i\eta\sigma_z}$$

[1] [Wim94]

⚛️ The BB1 Protocol[1] as QSP

$$U(\theta) = e^{i\pi\sigma_z/2} V(\theta) e^{-i\eta\sigma_z} V(\theta) e^{2i\eta\sigma_z} V^2(\theta) e^{-2i\eta\sigma_z} V(\theta) e^{i\eta\sigma_z}$$

For $\eta = (1/2)\cos^{-1}(-1/4) \approx 0.912$ this yields:

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

$$|P(x)|^2 = \frac{1}{8}(30x^2 - 45x^4 + 35x^6 - 15x^8 + 3x^{10}).$$

[1] [Wim94]

⚛️ The BB1 Protocol[1] as QSP

$$U(\theta) = e^{i\pi\sigma_z/2} V(\theta) e^{-i\eta\sigma_z} V(\theta) e^{2i\eta\sigma_z} V^2(\theta) e^{-2i\eta\sigma_z} V(\theta) e^{i\eta\sigma_z}$$

For $\eta = (1/2)\cos^{-1}(-1/4) \approx 0.912$ this yields:

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

$$|P(x)|^2 = \frac{1}{8}(30x^2 - 45x^4 + 35x^6 - 15x^8 + 3x^{10}).$$

For small θ, $|\langle 0|U(\theta)|0\rangle|^2$ expands $1 + (5/512)\theta^6 + \ldots$

[1] [Wim94]

Achievable polynomials

Achievable polynomials

Two components can be chosen (mostly) freely; e.g., $C(x) = 0$, and $|A(x)|^2 \leq 1$ on $x \in [-1, 1]$; measurement in $|\pm\rangle$ basis yields $p_+ = |A(x)|^2$

Achievable polynomials

Two components can be chosen (mostly) freely; e.g., $C(x) = 0$, and $|A(x)|^2 \le 1$ on $x \in [-1, 1]$; measurement in $|\pm\rangle$ basis yields $p_+ = |A(x)|^2$

$$U_\Phi(x) = \begin{bmatrix} A(x) + iB(x) & i\sqrt{1-x^2}[C(x) + iD(x)] \\ * & * \end{bmatrix}$$

Achievable polynomials

Two components can be chosen (mostly) freely; e.g., $C(x) = 0$, and $|A(x)|^2 \leq 1$ on $x \in [-1, 1]$; measurement in $|\pm\rangle$ basis yields $p_+ = |A(x)|^2$

$$U_\Phi(x) = \begin{bmatrix} A(x) + iB(x) & i\sqrt{1-x^2}[C(x) + iD(x)] \\ * & * \end{bmatrix}$$

Proof method: specify components, **complete** unitary, and **layer strip** phases by induction

Achievable polynomials

Two components can be chosen (mostly) freely; e.g., $C(x) = 0$, and $|A(x)|^2 \leq 1$ on $x \in [-1, 1]$; measurement in $|\pm\rangle$ basis yields $p_+ = |A(x)|^2$

$$U_\Phi(x) = \begin{bmatrix} A(x) + iB(x) & i\sqrt{1-x^2}[C(x) + iD(x)] \\ * & * \end{bmatrix}$$

Proof method: specify components, complete unitary, and layer strip phases by induction

⚠️ For other parameterizations ($x \mapsto [z + z^{-1}]/2$) or ansätze, this can change!

QSP ansätze and functional constraints

$$|P|^2 + (1 - x^2)|Q|^2 = 1.$$

The circuit depth for P of degree d is d. Moreover, $x \mapsto -x$ changes signs of P, Q, and $x \in \{\pm 1, 0\}$ induce constraints

QSP ansätze and functional constraints

$$|A|^2 + (1 - x^2)|C|^2 \leq 1, \quad A, C \in \mathbb{R}[x].$$

The circuit depth for P of degree d is d. Moreover, $x \mapsto -x$ changes signs of P, Q, and $x \in \{\pm 1, 0\}$ induce constraints

Partial specification (**A**, **C**) up to inequality is simpler!

QSP ansätze and functional constraints

$$|A|^2 + (1 - x^2)|C|^2 \leq 1, \quad A, C \in \mathbb{R}[x].$$

The circuit depth for P of degree d is d. Moreover, $x \mapsto -x$ changes signs of P, Q, and $x \in \{\pm 1, 0\}$ induce constraints

Partial specification (**A**, **C**) up to inequality is simpler!

Circumvention:[1] additional qubits + LCU [CW12], or change phases to arbitrary SU(2) elements [MW24]

[1]To be discussed later!

QSP ansätze and functional constraints

$$|A|^2 + (1 - x^2)|C|^2 \leq 1, \quad A, C \in \mathbb{R}[x].$$

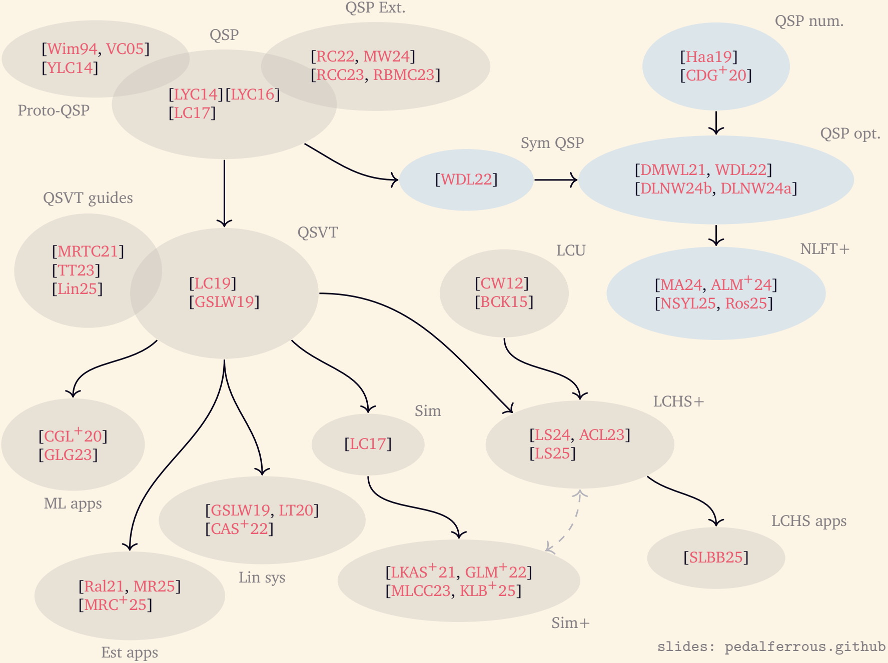
The circuit depth for P of degree d is d. Moreover, $x \mapsto -x$ changes signs of P, Q, and $x \in \{\pm 1, 0\}$ induce constraints

Partial specification (**A**, **C**) up to inequality is simpler!

Circumvention:[1] additional qubits + LCU [CW12], or change phases to arbitrary SU(2) elements [MW24]

Finding A, C reduces to *norm-constrained polynomial approximation or interpolation!*

[1]To be discussed later!



[Wim94, VC05]
[YLC14]
Proto-QSP
QSP
[LYC14][LYC16]
[LC17]
QSP Ext.
[RC22, MW24]
[RCC23, RBMC23]
QSP num.
[Haa19]
[CDG+20]
Sym QSP
[WDL22]
QSP opt.
[DMWL21, WDL22]
[DLNW24b, DLNW24a]
NLFT+
[MA24, ALM+24]
[NSYL25, Ros25]
QSVT guides
[MRTC21]
[TT23]
[Lin25]
QSVT
[LC19]
[GSLW19]
LCU
[CW12]
[BCK15]
[CGL+20]
[GLG23]
ML apps
Sim
[LC17]
LCHS+
[LS24, ACL23]
[LS25]
[GSLW19, LT20]
[CAS+22]
Lin sys
[Ral21, MR25]
[MRC+25]
Est apps
[LKAS+21, GLM+22]
[MLCC23, KLB+25]
Sim+
LCHS apps
[SLBB25]

Numerical methods in QSP

Numerical methods in QSP

Extensive study of phase finding algorithms[1]

[1] [Haa19, CDG$^+$20, WDL22, MA24, ALM$^+$24, NSYL25].

Numerical methods in QSP

Extensive study of phase finding algorithms[1]

$$(f, g) \mapsto (A, C), \text{ then}^2\ (A, C) \mapsto (P, Q), \text{ then } (P, Q) \mapsto \Phi.$$

[1] [Haa19, CDG$^+$20, WDL22, MA24, ALM$^+$24, NSYL25].
[2] By the Fejér-Riesz theorem [PS98]

Numerical methods in QSP

Extensive study of phase finding algorithms[1]

$$(f,g) \mapsto (A,C), \text{ then}^2\ (A,C) \mapsto (P,Q), \text{ then } (P,Q) \mapsto \Phi.$$

$$(P_n, Q_n) \mapsto \left\{ (P_{n-1}, Q_{n-1}), \phi_n \right\} \text{ and induct.}$$

[1] [Haa19, CDG+20, WDL22, MA24, ALM+24, NSYL25].

[2] By the Fejér-Riesz theorem [PS98]

Numerical methods in QSP

Extensive study of phase finding algorithms[1]

$$(f, g) \mapsto (A, C), \text{ then}^{2} \ (A, C) \mapsto (P, Q), \text{ then } (P, Q) \mapsto \Phi.$$

$$(P_n, Q_n) \mapsto \left\{ (P_{n-1}, Q_{n-1}), \phi_n \right\} \text{ and induct.}$$

$$\Phi^{t+1} = \Phi^t - [DF(\Phi^t)]^{-1}(F(\Phi^t) - A) \text{ quasi-Newton.}$$

[1] [Haa19, CDG$^+$20, WDL22, MA24, ALM$^+$24, NSYL25].

[2] By the Fejér-Riesz theorem [PS98]

Numerical methods in QSP

Extensive study of phase finding algorithms[1]

$$(f,g) \mapsto (A,C), \text{ then}^2\ (A,C) \mapsto (P,Q), \text{ then } (P,Q) \mapsto \Phi.$$

$$(P_n, Q_n) \mapsto \left\{(P_{n-1}, Q_{n-1}), \phi_n\right\} \text{ and induct.}$$

$$\Phi^{t+1} = \Phi^t - [DF(\Phi^t)]^{-1}(F(\Phi^t) - A) \text{ quasi-Newton.}$$

When $\Phi = \Phi^R$ and proper initialization, this works[3] for almost all $||f||_\infty \leq 1$. Proof known for *Szegő functions*!

[1] [Haa19, CDG$^+$20, WDL22, MA24, ALM$^+$24, NSYL25].

[2] By the Fejér-Riesz theorem [PS98]

[3] [DMWL21, DLNW24b, DLNW24a]

Numerical methods in QSP

Extensive study of phase finding algorithms[1]

$$(f, g) \mapsto (A, C), \text{ then}^2\ (A, C) \mapsto (P, Q), \text{ then } (P, Q) \mapsto \Phi.$$

$$(P_n, Q_n) \mapsto \left\{ (P_{n-1}, Q_{n-1}), \phi_n \right\} \text{ and induct.}$$

$$\Phi^{t+1} = \Phi^t - [DF(\Phi^t)]^{-1}(F(\Phi^t) - A) \text{ quasi-Newton.}$$

When $\Phi = \Phi^R$ and proper initialization, this works[3] for almost all $||f||_\infty \leq 1$. Proof known for *Szegő functions*!

✨ QSPPACK: `https://github.com/qsppack/QSPPACK`
✨ pyQSP: `https://github.com/ichuang/pyqsp`.

[1] [Haa19, CDG+20, WDL22, MA24, ALM+24, NSYL25].
[2] By the Fejér-Riesz theorem [PS98]
[3] [DMWL21, DLNW24b, DLNW24a]

What we know: QSP *usefully*[1] characterizes achievable SU(2)-valued functions

What we want to know: How can QSP *usefully* transform quantum systems comprising *many qubits*?

[1]Stable, efficient classical companion algorithms

⏱️

What we know: QSP *usefully*[1] characterizes achievable SU(2)-valued functions

What we want to know: How can QSP *usefully* transform quantum systems comprising *many qubits*?

Up next: finding qubits inside large unitaries

[1] Stable, efficient classical companion algorithms

💡 Revisiting the Grover iterate

💡 Revisiting the Grover iterate

The ability to reflect about a quantum state is powerful[1]

[1] [Sze04, Gro05, Reg06, YLC14]

💡 Revisiting the Grover iterate

The ability to reflect about a quantum state is powerful[1]

The product of two reflections generates a rotation in a related plane [Jor75]

[1] [Sze04, Gro05, Reg06, YLC14]

💡 Revisiting the Grover iterate

The ability to reflect about a quantum state is powerful[1]

The product of two reflections generates a rotation in a related plane [Jor75]

$$|s\rangle = \xi\,|t\rangle + \sqrt{1-\xi^2}\,|t^{\perp}\rangle,$$
$$|t^{\perp}\rangle = (1-\xi^2)^{-1}\Big[\,|s\rangle - \langle t|s\rangle\,|t\rangle\Big].$$

The Grover iterate alternately reflects about the target state $|t\rangle$ and the uniform superposition $|s\rangle$

[1] [Sze04, Gro05, Reg06, YLC14]

💡 Revisiting the Grover iterate

The ability to reflect about a quantum state is powerful[1]

The product of two reflections generates a rotation in a related plane [Jor75]

$$|s\rangle = \xi\,|t\rangle + \sqrt{1-\xi^2}\,|t^\perp\rangle,$$
$$|t^\perp\rangle = (1-\xi^2)^{-1}\Big[\,|s\rangle - \langle t\,|\,s\rangle\,|t\rangle\Big].$$

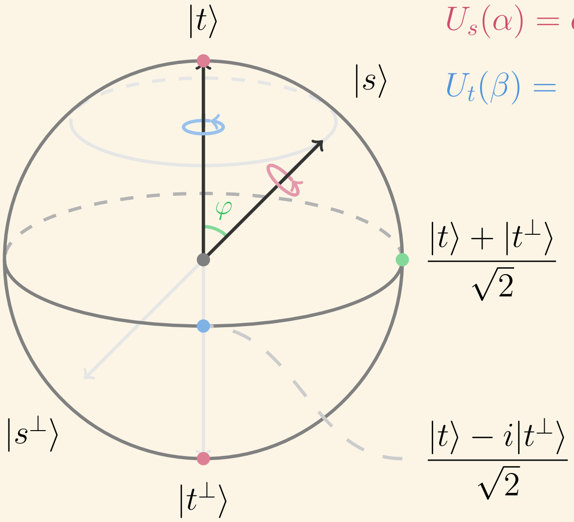
The Grover iterate alternately reflects about the target state $|t\rangle$ and the uniform superposition $|s\rangle$

$$-S_s(\pi)S_t(\pi) = \begin{bmatrix} -1+2\xi^2 & -2\xi\sqrt{1-\xi^2} \\ -2\xi\sqrt{1-\xi^2} & 1-2\xi^2 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

[1] [Sze04, Gro05, Reg06, YLC14]

$$U_s(\alpha) = e^{i\alpha(2|s\rangle\langle s| - I)}$$

$$U_t(\beta) = e^{i\beta\,(2|t\rangle\langle t| - I)}$$



$$S_s(\alpha) = I - (1 - e^{i\alpha}\,|s\rangle\langle s|), \qquad S_t(\beta) = I - (1 - e^{i\beta}\,|t\rangle\langle t|).$$

Block encoding [TT23]

Block encoding [TT23] Let $A \in \mathbb{C}^{r\times c}$ for $\alpha, \varepsilon > 0$.
Let $B_{L,1} \in \mathbb{C}^{d\times r}$, $B_{R,1} \in \mathbb{C}^{d\times c}$ have *orthonormal columns*.

Block encoding [TT23] Let $A \in \mathbb{C}^{r\times c}$ for $\alpha, \varepsilon > 0$.
Let $B_{L,1} \in \mathbb{C}^{d\times r}$, $B_{R,1} \in \mathbb{C}^{d\times c}$ have *orthonormal columns*.
A unitary $U \in \mathbb{C}^{d\times d}$ is an (α, ε)-block encoding of A if

$$\left\| A - \alpha B_{L,1}^{\dagger} U B_{R,1} \right\|_{\mathrm{op}} \leq \varepsilon.$$

Block encoding [TT23] Let $A \in \mathbb{C}^{r \times c}$ for $\alpha, \varepsilon > 0$.
Let $B_{L,1} \in \mathbb{C}^{d \times r}$, $B_{R,1} \in \mathbb{C}^{d \times c}$ have *orthonormal columns*.
A unitary $U \in \mathbb{C}^{d \times d}$ is an (α, ε)-block encoding of A if

$$\left\| A - \alpha B_{L,1}^{\dagger} U B_{R,1} \right\|_{\mathrm{op}} \leq \varepsilon.$$

We denote $\Pi_L = B_{L,1} B_{L,1}^{\dagger}$ and $\Pi_R = B_{R,1} B_{R,1}^{\dagger}$, the orthogonal projectors onto the span of the columns of $B_{R/L,1}$.

Block encoding [TT23] Let $A \in \mathbb{C}^{r\times c}$ for $\alpha, \varepsilon > 0$. Let $B_{L,1} \in \mathbb{C}^{d\times r}$, $B_{R,1} \in \mathbb{C}^{d\times c}$ have *orthonormal columns*. A unitary $U \in \mathbb{C}^{d\times d}$ is an (α, ε)-block encoding of A if

$$\left\| A - \alpha B_{L,1}^{\dagger} U B_{R,1} \right\|_{\mathrm{op}} \leq \varepsilon.$$

We denote $\Pi_L = B_{L,1} B_{L,1}^{\dagger}$ and $\Pi_R = B_{R,1} B_{R,1}^{\dagger}$, the orthogonal projectors onto the span of the columns of $B_{R/L,1}$.

E.g., a $(1,0)$ block encoding, with $B_L = (B_{L,1}, B_{L,2})$ and $B_R = (B_{R,1}, B_{R,2})$ the 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 (α, a, ε)-block encoding of A [GSLW19]

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

where A is s-qubit, and U is $(s+a)$ qubit, and, e.g., $(\langle 0|^{\otimes a} \otimes I) = B_{L,1}^{\dagger}$ and $\Pi_L = |0\rangle\langle 0|^{\otimes a} \otimes I$.

An (α, a, ε)-block encoding of A [GSLW19]

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

where A is s-qubit, and U is $(s+a)$ qubit, and, e.g., $(\langle 0|^{\otimes a} \otimes I) = B_{L,1}^{\dagger}$ and $\Pi_L = |0\rangle\langle 0|^{\otimes a} \otimes I$.

Sometimes say $\Pi_L U \Pi_R = A$, ignoring zero blocks of U.
Then A has a singular value decomposition

$$A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|, \qquad \text{maps red to green!}$$

An (α, a, ε)-block encoding of A [GSLW19]

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

where A is s-qubit, and U is $(s+a)$ qubit, and, e.g., $(\langle 0|^{\otimes a} \otimes I) = B^\dagger_{L,1}$ and $\Pi_L = |0\rangle\langle 0|^{\otimes a} \otimes I$.

Sometimes say $\Pi_L U \Pi_R = A$, ignoring zero blocks of U. Then A has a singular value decomposition

$$A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|, \qquad \text{maps red to green!}$$

Observe action of U on a **right** singular vector

$$U|\psi_i\rangle = \Pi_L U \Pi_R |\psi_i\rangle + (1 - \Pi_L) U |\psi_i\rangle \tag{1}$$

$$= \xi_i |\psi_i\rangle + \sqrt{1 - \xi_i^2}\, |*\rangle \tag{2}$$

Let Π_L, Π_R, U, A as before, ξ_i the i-th singular value of A, k the largest index where $\xi_k = 1$, and $r = \mathrm{rank}(A)$.

$$A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|$$

Let Π_L, Π_R, U, A as before, ξ_i the i-th singular value of A, k the largest index where $\xi_k = 1$, and $r = \mathrm{rank}(A)$.

$$A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|$$

$d = \mathrm{rank}(\Pi_R)$, $\tilde{d} = \mathrm{rank}(\Pi_L)$; $|\psi_i\rangle / |\tilde{\psi}_i\rangle$ are the **right**/**left** singular vectors of A (bases for $\mathrm{img}(\Pi_R)$, $\mathrm{img}(\Pi_L)$).[1]

[1]Note also $\mathscr{H}^\perp / \tilde{\mathscr{H}}^\perp$, where action of U is unspecified.

Let Π_L, Π_R, U, A as before, ξ_i the i-th singular value of A, k the largest index where $\xi_k = 1$, and $r = \mathrm{rank}(A)$.

$$A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|$$

$d = \mathrm{rank}(\Pi_R)$, $\tilde{d} = \mathrm{rank}(\Pi_L)$; $|\psi_i\rangle / |\tilde{\psi}_i\rangle$ are the **right**/**left** singular vectors of A (bases for $\mathrm{img}(\Pi_R)$, $\mathrm{img}(\Pi_L)$).[1]

$$|\psi_i^{\perp}\rangle \propto (I - \Pi_R) U^{\dagger} |\tilde{\psi}_i\rangle,$$
$$|\tilde{\psi}_i^{\perp}\rangle \propto (I - \Pi_L) U |\psi_i\rangle.$$

[1]Note also $\mathscr{H}^{\perp} / \tilde{\mathscr{H}}^{\perp}$, where action of U is unspecified.

Let Π_L, Π_R, U, A as before, ξ_i the i-th singular value of A, k the largest index where $\xi_k = 1$, and $r = \mathrm{rank}(A)$.

$$A = \sum_i \xi_i |\tilde{\psi}_i\rangle\langle\psi_i|$$

$d = \mathrm{rank}(\Pi_R)$, $\tilde{d} = \mathrm{rank}(\Pi_L)$; $|\psi_i\rangle / |\tilde{\psi}_i\rangle$ are the **right**/**left** singular vectors of A (bases for $\mathrm{img}(\Pi_R)$, $\mathrm{img}(\Pi_L)$).[1]

$$|\psi_i^\perp\rangle \propto (I - \Pi_R) U^\dagger |\tilde{\psi}_i\rangle,$$
$$|\tilde{\psi}_i^\perp\rangle \propto (I - \Pi_L) U |\psi_i\rangle.$$

$$\begin{aligned}
&\mathscr{H}_i = \mathrm{span}(|\psi_i\rangle), && \tilde{\mathscr{H}}_i = \mathrm{span}(|\tilde{\psi}_i\rangle), && i \in [k], \\
&\mathscr{H}_i = \mathrm{span}(|\psi_i\rangle, |\psi_i^\perp\rangle), && \tilde{\mathscr{H}}_i = \mathrm{span}(|\tilde{\psi}_i\rangle, |\tilde{\psi}_i^\perp\rangle), && i \in [r]\backslash[k], \\
&\mathscr{H}_i^R = \mathrm{span}(|\psi_i\rangle), && \tilde{\mathscr{H}}_i^R = \mathrm{span}(U|\psi_i\rangle), && i \in [d]\backslash[r], \\
&\mathscr{H}_i^L = \mathrm{span}(U^\dagger|\tilde{\psi}_i\rangle), && \tilde{\mathscr{H}}_i^L = \mathrm{span}(|\tilde{\psi}_i\rangle), && i \in [\tilde{d}]\backslash[r].
\end{aligned}$$

[1]Note also $\mathscr{H}^\perp / \tilde{\mathscr{H}}^\perp$, where action of U is unspecified.

Qubitization: we have to show $\mathscr{H}_i$, $\mathscr{H}_i^R$, and $\mathscr{H}_i^L$ and so on are pairwise orthogonal:

Qubitization: we have to show $\mathscr{H}_i$, $\mathscr{H}_i^R$, and $\mathscr{H}_i^L$ and so on are pairwise orthogonal:

$$\langle \psi_i | \psi_j \rangle = \delta_{ij}, \quad \mathscr{H}_i \perp \mathscr{H}_j, \tag{3}$$

$$\langle \tilde{\psi}_i | \tilde{\psi}_j \rangle = \delta_{ij}, \quad \tilde{\mathscr{H}}_i \perp \tilde{\mathscr{H}}_j, \tag{4}$$

$$\langle \psi_i^\perp | \psi_j^\perp \rangle = \langle \tilde{\psi}_i^\perp | \tilde{\psi}_j^\perp \rangle = \delta_{ij}, \quad (\mathscr{H}_i)^\perp \perp (\tilde{\mathscr{H}}_j)^\perp, \tag{5}$$

$$\langle \psi_i | \psi_j^\perp \rangle = \langle \tilde{\psi}_i | \tilde{\psi}_j^\perp \rangle = 0, \quad */* \perp */*, \tag{6}$$

$$\langle \psi_i | U^\dagger | \tilde{\psi}_j \rangle = 0, \quad U^\dagger | \tilde{\psi}_j \rangle \in \mathscr{H}_j^L, \tag{7}$$

$$\langle \psi_i^\perp | U^\dagger | \tilde{\psi}_j \rangle = 0, \quad U^\dagger | \tilde{\psi}_j \rangle \in \mathscr{H}_j^L. \tag{8}$$

Qubitization: we have to show $\mathscr{H}_i$, $\mathscr{H}_i^R$, and $\mathscr{H}_i^L$ and so on are pairwise orthogonal:

$$\langle \psi_i | \psi_j \rangle = \delta_{ij}, \quad \mathscr{H}_i \perp \mathscr{H}_j, \tag{3}$$

$$\langle \tilde{\psi}_i | \tilde{\psi}_j \rangle = \delta_{ij}, \quad \tilde{\mathscr{H}}_i \perp \tilde{\mathscr{H}}_j, \tag{4}$$

$$\langle \psi_i^\perp | \psi_j^\perp \rangle = \langle \tilde{\psi}_i^\perp | \tilde{\psi}_j^\perp \rangle = \delta_{ij}, \quad (\mathscr{H}_i)^\perp \perp (\tilde{\mathscr{H}}_j)^\perp, \tag{5}$$

$$\langle \psi_i | \psi_j^\perp \rangle = \langle \tilde{\psi}_i | \tilde{\psi}_j^\perp \rangle = 0, \quad */* \perp */*, \tag{6}$$

$$\langle \psi_i | U^\dagger | \tilde{\psi}_j \rangle = 0, \quad U^\dagger | \tilde{\psi}_j \rangle \in \mathscr{H}_j^L, \tag{7}$$

$$\langle \psi_i^\perp | U^\dagger | \tilde{\psi}_j \rangle = 0, \quad U^\dagger | \tilde{\psi}_j \rangle \in \mathscr{H}_j^L. \tag{8}$$

The first three from singular vector orthogonality; note that $\langle \tilde{\psi}_i | U \Pi_R U^\dagger | \tilde{\psi}_j \rangle$ can be replaced by $\langle \tilde{\psi}_i | A A^\dagger | \tilde{\psi}_j \rangle$. The final three follow from definition of projectors.

The components of the QSVT circuit[1]

$$U = \bigoplus_{i\in[k]} [\xi_i]_{\tilde{\mathscr{H}}_i}^{\mathscr{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} \xi_i & \sqrt{1-\xi_i^2} \\ \sqrt{1-\xi_i^2} & \xi_i \end{bmatrix}_{\tilde{\mathscr{H}}_i}^{\mathscr{H}_i} \oplus$$

$$\bigoplus_{i\in[d]\setminus[r]} [1]_{\tilde{\mathscr{H}}_i^R}^{\mathscr{H}_i^R} \oplus \bigoplus_{i\in[\tilde{d}]\setminus[r]} [1]_{\tilde{\mathscr{H}}_i^L}^{\mathscr{H}_i^L} \oplus [*]_{\tilde{\mathscr{H}}^\perp}^{\mathscr{H}^\perp},$$

$$e^{i\phi(2\Pi_R - I)} = \bigoplus_{i\in[k]} [e^{i\phi}]_{\mathscr{H}_i}^{\mathscr{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{bmatrix}_{\mathscr{H}_i}^{\mathscr{H}_i} \oplus$$

$$\bigoplus_{i\in[d]\setminus[r]} [e^{i\phi}]_{\mathscr{H}_i^R}^{\mathscr{H}_i^R} \oplus \bigoplus_{i\in[d]\setminus[r]} [e^{-i\phi}]_{\mathscr{H}_i^L}^{\mathscr{H}_i^L} \oplus [*]_{\mathscr{H}^\perp}^{\mathscr{H}^\perp},$$

[1]Our notation maps top subspace to bottom: $\mathbf{r} \downarrow \mathbf{g}$

***QSVT* [GSLW19]:** Let $\Phi = \{\phi_j\}_{j\in[n]} \in \mathbb{R}^n$; the QSVT protocol associated with Φ and block encoding U has circuit (taking n odd):

$$U_\Phi \equiv e^{i\phi_1(2\Pi_L - I)} U \prod_{j=1}^{(n-1)/2} e^{i\phi_{2j}(2\Pi_R - I)} U^\dagger e^{i\phi_{2j+1}(2\Pi_L - I)} U.$$

***QSVT* [GSLW19]:** Let $\Phi = \{\phi_j\}_{j\in[n]} \in \mathbb{R}^n$; the QSVT protocol associated with Φ and block encoding U has circuit (taking n odd):

$$U_\Phi \equiv e^{i\phi_1(2\Pi_L - I)} U \prod_{j=1}^{(n-1)/2} e^{i\phi_{2j}(2\Pi_R - I)} U^\dagger e^{i\phi_{2j+1}(2\Pi_L - I)} U.$$

Simultaneous QSP with phases Φ within exponentially many invariant subspaces, given U, $U^\dagger$, single-qubit gates, and $\Pi_{R/L}$-controlled NOT.

***QSVT* [GSLW19]:** Let $\Phi = \{\phi_j\}_{j\in[n]} \in \mathbb{R}^n$; the QSVT protocol associated with Φ and block encoding U has circuit (taking n odd):

$$U_\Phi \equiv e^{i\phi_1(2\Pi_L - I)} U \prod_{j=1}^{(n-1)/2} e^{i\phi_{2j}(2\Pi_R - I)} U^\dagger e^{i\phi_{2j+1}(2\Pi_L - I)} U.$$

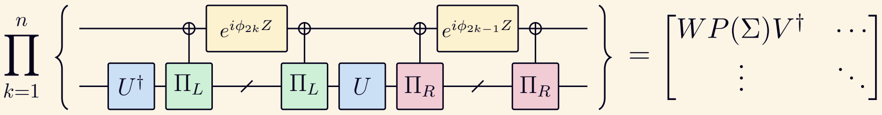
Simultaneous QSP with phases Φ within exponentially many invariant subspaces, given U, $U^\dagger$, single-qubit gates, and $\Pi_{R/L}$-controlled NOT.

$$U_\Phi = \bigoplus_{i\in[k]} [P(\xi_i)]_{\tilde{\mathscr{H}}_i}^{\mathscr{H}_i} \oplus \bigoplus_{i\in[r]\setminus[k]} \begin{bmatrix} P(\xi_i) & * \\ * & * \end{bmatrix}_{\tilde{\mathscr{H}}_i}^{\mathscr{H}_i} \oplus \bigoplus_{i\in[d]\setminus[r]} [e^{i\phi_0}]_{\tilde{\mathscr{H}}_i^R}^{\mathscr{H}_i^R} \oplus \bigoplus_{i\in[\tilde{d}]\setminus[r]} [e^{-i\phi_0}]_{\tilde{\mathscr{H}}_i^L}^{\mathscr{H}_i^L} \oplus [*]_{\tilde{\mathscr{H}}^\perp}^{\mathscr{H}^\perp},$$

***QSVT* [GSLW19]:** Let $\Phi = \{\phi_j\}_{j\in[n]} \in \mathbb{R}^n$; the QSVT protocol associated with Φ and block encoding U has circuit (taking n odd):

$$U_\Phi \equiv e^{i\phi_1(2\Pi_L - I)} U \prod_{j=1}^{(n-1)/2} e^{i\phi_{2j}(2\Pi_R - I)} U^\dagger e^{i\phi_{2j+1}(2\Pi_L - I)} U.$$

Simultaneous QSP with phases Φ within exponentially many invariant subspaces, given U, $U^\dagger$, single-qubit gates, and $\Pi_{R/L}$-controlled NOT.



Another approach:[1] *the cosine-sine decomposition*

[1]The CSD is also useful in gate synthesis! [Tan25]

Another approach:[1] *the cosine-sine decomposition*

It turns out one can produce *simultaneous* SVDs for multiple unitaries at once: $U_{ij} = V_i D_{ij} W_j^\dagger$ [TT23, PW94]

[1] The CSD is also useful in gate synthesis! [Tan25]

Another approach:[1] *the cosine-sine decomposition*

It turns out one can produce *simultaneous* SVDs for multiple unitaries at once: $U_{ij} = V_i D_{ij} W_j^\dagger$ [TT23, PW94]

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

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

[1]The CSD is also useful in gate synthesis! [Tan25]

Another approach:[1] *the cosine-sine decomposition*

It turns out one can produce *simultaneous* SVDs for multiple unitaries at once: $U_{ij} = V_i D_{ij} W_j^\dagger$ [TT23, PW94]

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

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

Then there exist unitaries $V_i \in \mathbb{C}^{r_i \times r_i}$ and $W_j \in \mathbb{C}^{c_j \times c_j}$ s.t.

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

where each D_{ij} is diagonal in $\mathbb{C}^{r_i \times c_j}$, possibly zero-padded.

[1]The CSD is also useful in gate synthesis! [Tan25]

Specifically, we can write D as:

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

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

where C, S are square w/ entries in $(0,1)$, and $C^2 + S^2 = I$

Cosine-sine decomposition proof sketch

1. Start with the SVD of one block $U_{11} = V_1 D_{11} W_1^\dagger$

2. Compute QR decompositions of $U_{21} W_1$ and $U_{12}^\dagger V_1$, giving 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 unitary (whose rows and columns must be orthonormal), this forces the entries of D_{12}, D_{21} to be diagonal and $C^2 + S^2 = I$

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

The limits of qubitization

⏱️

The limits of qubitization

In general, invariant subspaces for two reflections are only **one- and two-dimensional** [Jor75]

No good notion for *quditization*

⏱️

The limits of qubitization

In general, invariant subspaces for two reflections are only **one- and two-dimensional** [Jor75]

No good notion for *quditization*

Weaker variants of G-SVD do exist, applying to many matrices at once [PW94]

⏱️

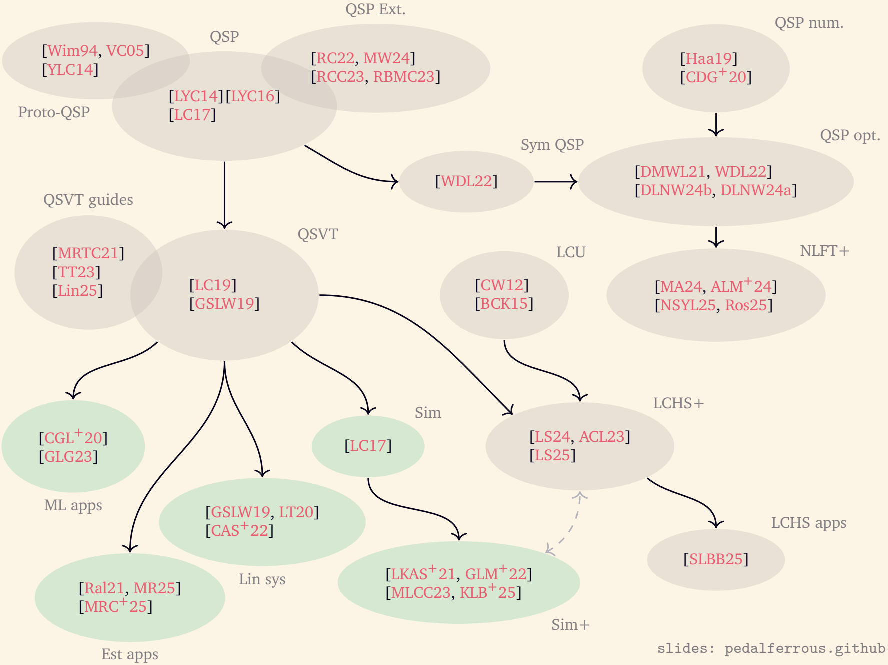
The limits of qubitization

In general, invariant subspaces for two reflections are only **one- and two-dimensional** [Jor75]

No good notion for *quditization*

Weaker variants of G-SVD do exist, applying to many matrices at once [PW94]

Question: Weaker qubitization? Can we transform Jordan blocks [LS24] ? Irrep labels? Oracle-marked subspaces?


Proto-QSP
[Wim94, VC05]
[YLC14]
QSP
[LYC14][LYC16]
[LC17]
QSP Ext.
[RC22, MW24]
[RCC23, RBMC23]
QSP num.
[Haa19]
[CDG+20]
Sym QSP
[WDL22]
QSP opt.
[DMWL21, WDL22]
[DLNW24b, DLNW24a]
QSVT guides
[MRTC21]
[TT23]
[Lin25]
QSVT
[LC19]
[GSLW19]
LCU
[CW12]
[BCK15]
NLFT+
[MA24, ALM+24]
[NSYL25, Ros25]
ML apps
[CGL+20]
[GLG23]
Sim
[LC17]
LCHS+
[LS24, ACL23]
[LS25]
Lin sys
[GSLW19, LT20]
[CAS+22]
Est apps
[Ral21, MR25]
[MRC+25]
Sim+
[LKAS+21, GLM+22]
[MLCC23, KLB+25]
LCHS apps
[SLBB25]

QSP is all* you need[1]

[1][MRTC21]

QSP is all* you need[1]

Matrix functions for large* linear operators

[1] [MRTC21]

QSP is all* you need[1]

Matrix functions for large* linear operators

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

1 [MRTC21]

QSP is all* you need[1]

Matrix functions for large* linear operators

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

Search:
Input **Grover oracle**, **apply constant function**
Low energy projection:
Input **Hamiltonian**, **apply bandpass function**
Inversion:
Input **sparse linear sys**, **apply $1/x$ approximation**
Simulation:
Input **Hamiltonian**, **apply trig function** …

[1] [MRTC21]

QSP is all* you need[1]

Matrix functions for large* linear operators

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

Search:
Input **Grover oracle**, **apply** **constant function**
Low energy projection:
Input **Hamiltonian**, **apply** **bandpass function**
Inversion:
Input **sparse linear sys**, **apply** **$1/x$ approximation**
Simulation:
Input **Hamiltonian**, **apply** **trig function** …

🌞 ***Changing the polynomial changes the algorithm*** 🌞

[1] [MRTC21]

QSVT in practice

QSVT in practice

Simulation

Linear system solving

Estimation tasks

QSVT in practice

Simulation

Linear system solving

Estimation tasks

👉 QSVT is most useful when demanding **precise, coherent** answers to **discrete or adaptive** questions[1]

[1] E.g., cryptography [LMS21], modified QPE [Ral21].

QSVT in practice

Simulation

Linear system solving

Estimation tasks

👉 QSVT is most useful when demanding **precise, coherent** answers to **discrete or adaptive** questions[1]

⚠️ QSVT captures BQP-complete problems; if you ask for too much, it'll be expensive

[1] E.g., cryptography [LMS21], modified QPE [Ral21].

Block encodings, revisited

Block encodings, revisited

1. Approximating generic unitaries is expensive [KSV02]

Block encodings, revisited

1. Approximating generic unitaries is expensive [KSV02]

2. Blocks have spectral norm bound of one; projecting onto proper block requires amplification [GSLW19]

Block encodings, revisited

1. Approximating generic unitaries is expensive [KSV02]

2. Blocks have spectral norm bound of one; projecting onto proper block requires amplification [GSLW19]

Moral: Block encodings circuits have to be efficient, and subnormalization can't be too severe

Block encodings, revisited

1. Approximating generic unitaries is expensive [KSV02]

2. Blocks have spectral norm bound of one; projecting onto proper block requires amplification [GSLW19]

Moral: Block encodings circuits have to be efficient, and subnormalization can't be too severe

⚠️: Constant factors seem to matter [KREO25]; error propagation requires care [TT23, GSLW19]

When is block encoding asymptotically cheap?

When is block encoding asymptotically cheap?

Controlled access, sparsity, purification unitary [GSLW19], displacement structure [CLVBY24], Hamiltonian evolution [LKAS+21], state copies [LMR14] ...

When is block encoding asymptotically cheap?

Controlled access, sparsity, purification unitary [GSLW19], displacement structure [CLVBY24], Hamiltonian evolution [LKAS+21], state copies [LMR14] ...

$$\{\rho, \rho, \ldots, \rho\} \mapsto e^{i\rho t} \mapsto \begin{bmatrix} \rho & * \\ * & * \end{bmatrix}$$

When is block encoding asymptotically cheap?

Controlled access, sparsity, purification unitary [GSLW19], displacement structure [CLVBY24], Hamiltonian evolution [LKAS+21], state copies [LMR14] ...

$$\{\rho, \rho, \ldots, \rho\} \mapsto e^{i\rho t} \mapsto \begin{bmatrix} \rho & * \\ * & * \end{bmatrix}$$

Multiple quantitative investigations for realistic instances [SCC24, CLVBY24, KREO25]

When is block encoding asymptotically cheap?

Controlled access, sparsity, purification unitary [GSLW19], displacement structure [CLVBY24], Hamiltonian evolution [LKAS+21], state copies [LMR14] ...

$$\{\rho, \rho, \ldots, \rho\} \mapsto e^{i\rho t} \mapsto \begin{bmatrix} \rho & * \\ * & * \end{bmatrix}$$

Multiple quantitative investigations for realistic instances [SCC24, CLVBY24, KREO25]

Beyond instantiation, diverse methods for (approximately) combining block encodings [LW19, VG25]

Is QSVT the end of quantum algorithms?

Is QSVT the end of quantum algorithms?

🤷‍♂️ 🙅‍♂️ 🤦‍♂️

Is QSVT the end of quantum algorithms?

🤷‍♂️ 🙅‍♂️ 🤦‍♂️

Helps once you've converted problem to spectral mapping & ensured efficient block encoding

Is QSVT the end of quantum algorithms?

🤷‍♂️ 🙅‍♂️ 🤦‍♂️

Helps once you’ve converted problem to spectral mapping & ensured efficient block encoding

QSVT already captures BQP-complete problems

Is QSVT the end of quantum algorithms?

🤷‍♂️ 🙅‍♂️ 🤦‍♂️

Helps once you've converted problem to spectral mapping & ensured efficient block encoding

QSVT already captures BQP-complete problems

Highly coherent, though space efficient

Is QSVT the end of quantum algorithms?

🤷‍♂️ 🙅‍♂️ 🤦‍♂️

Helps once you've converted problem to spectral mapping & ensured efficient block encoding

QSVT already captures BQP-complete problems

Highly coherent, though space efficient

👉 **A clean abstraction**; use of diverse functional analytic, representation theoretic, and numerical linear algebraic tools is made clear

📌 A note on dequantization and QSVT

📌 A note on dequantization and QSVT

Moral: If matrix is low rank, block encoding is QRAM-based,[1] or your required precision is constant, then sketching methods work [CGL+20, GLG23, EHG25]

[1] E.g., recommendation systems, principal component analysis, supervised clustering, support vector machines, low-rank regression, semidefinite program solving, etc.

📌 A note on dequantization and QSVT

Moral: If matrix is low rank, block encoding is QRAM-based,[1] or your required precision is constant, then sketching methods work [CGL+20, GLG23, EHG25]

(Over)sampling and query access: analogues to quantum state preparation assumptions

[1] E.g., recommendation systems, principal component analysis, supervised clustering, support vector machines, low-rank regression, semidefinite program solving, etc.

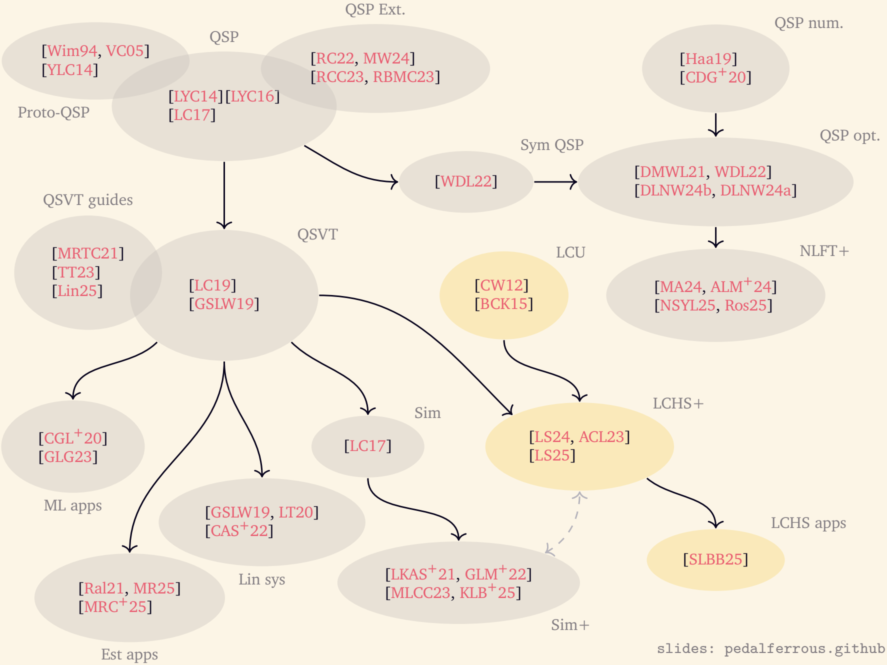
📌 A note on dequantization and QSVT

Moral: If matrix is low rank, block encoding is QRAM-based,[1] or your required precision is constant, then sketching methods work [CGL+20, GLG23, EHG25]

(Over)sampling and query access: analogues to quantum state preparation assumptions

In practice, there may be large polynomial separations even in these cases, but improvement is rapid!

[1] E.g., recommendation systems, principal component analysis, supervised clustering, support vector machines, low-rank regression, semidefinite program solving, etc.



Proto-QSP
[Wim94, VC05]
[YLC14]
QSP
[LYC14][LYC16]
[LC17]
QSP Ext.
[RC22, MW24]
[RCC23, RBMC23]
QSP num.
[Haa19]
[CDG+20]
Sym QSP
[WDL22]
QSP opt.
[DMWL21, WDL22]
[DLNW24b, DLNW24a]
QSVT guides
[MRTC21]
[TT23]
[Lin25]
QSVT
[LC19]
[GSLW19]
LCU
[CW12]
[BCK15]
NLFT+
[MA24, ALM+24]
[NSYL25, Ros25]
ML apps
[CGL+20]
[GLG23]
Sim
[LC17]
LCHS+
[LS24, ACL23]
[LS25]
Lin sys
[GSLW19, LT20]
[CAS+22]
Est apps
[Ral21, MR25]
[MRC+25]
Sim+
[LKAS+21, GLM+22]
[MLCC23, KLB+25]
LCHS apps
[SLBB25]

👪 The great family of block encoding algorithms

QSP [LC17], QSVT [GSLW19], QEP [LS24], G-QSP [MW24], M-QSP [RC22], LCU [CW12], LCHS [ACL23], randomized, parallelized, and hybrid variants!

👪 The great family of block encoding algorithms

QSP [LC17], QSVT [GSLW19], QEP [LS24], G-QSP [MW24], M-QSP [RC22], LCU [CW12], LCHS [ACL23], randomized, parallelized, and hybrid variants!

Unifying aspect: precise control of subsystem dynamics

🧑‍🧑‍🧒 The great family of block encoding algorithms

QSP [LC17], QSVT [GSLW19], QEP [LS24], G-QSP [MW24], M-QSP [RC22], LCU [CW12], LCHS [ACL23], randomized, parallelized, and hybrid variants!

Unifying aspect: precise control of subsystem dynamics

⚠️ Comparison not always possible, and choices between these can depend on architecture and instance size!

Linear combination of unitaries (LCU) [CW12, BCK15]

Linear combination of unitaries (LCU) [CW12, BCK15]

Originally formulated for efficient Hamiltonian simulation

Linear combination of unitaries (LCU) [CW12, BCK15]

Originally formulated for efficient Hamiltonian simulation

Let $V = \sum_k \alpha_k U_k$ for $\alpha_k > 0$ and U_k unitary. Let W be any unitary that acts as

$$W|0\rangle = \frac{1}{\sqrt{\alpha}} \sum_k \sqrt{\alpha_k}\,|k\rangle, \quad \alpha \equiv \sum_k \alpha_k.$$

Then

$$W^\dagger \left[\sum_k |k\rangle\langle k| \otimes U_k \right] W \equiv W^\dagger U W$$

is an $(\alpha, *, 0)$ block encoding of V. Often W, U are referred to as the PREPARE and SELECT operations.

LCU, QSVT, and Hamiltonian simulation

LCU, QSVT, and Hamiltonian simulation

Desired transformation written as a sum of block-encoded unitaries $\implies$ **LCU describes costs**

LCU, QSVT, and Hamiltonian simulation

Desired transformation written as a sum of block-encoded unitaries $\implies$ **LCU describes costs**

Time-independent Hamiltonian sim for d-term LCU

Method	aux qubits	query comp
LCU	$\lceil \log d \rceil \log(\alpha t/\varepsilon)$	$\lceil \alpha t \rceil \log(\alpha t/\varepsilon)$
QSVT	$\log d$	$\alpha t + \log(1/\varepsilon)$

LCU, QSVT, and Hamiltonian simulation

Desired transformation written as a sum of block-encoded unitaries $\implies$ **LCU describes costs**

Time-independent Hamiltonian sim for d-term LCU

Method	aux qubits	query comp
LCU	$\lceil \log d \rceil \log(\alpha t/\varepsilon)$	$\lceil \alpha t \rceil \log(\alpha t/\varepsilon)$
QSVT	$\log d$	$\alpha t + \log(1/\varepsilon)$

QSVT insight: quick approximation of entire functions

$$\cos(xt) = J_0(t) + 2\sum_{k=1}^{\infty}(-1)^k J_{2k}(t) T_{2k}(x),$$

$$\sin(xt) = 2\sum_{k=0}^{\infty}(-1)^k J_{2k+1}(t) T_{2k+1}(x),$$

Simulation beyond QSVT: LCHS

Simulation beyond QSVT: LCHS

$$\dot{u}(t) = -A(t)u(t) + b(t), \quad u(0) = u_0.$$

Simulation beyond QSVT: LCHS

$$\dot{u}(t) = -A(t)u(t) + b(t), \quad u(0) = u_0.$$

Approximating $u(t)$ relies on time-ordered integral:

$$u(t) = U_0(t)u_0 + \int_0^t U_s(t)b(s)\,ds,$$
$$U_s(t) \equiv \mathcal{T}\exp\left[\int_s^t A(s')\,ds'\right].$$

Simulation beyond QSVT: LCHS

$$\dot{u}(t) = -A(t)u(t) + b(t), \quad u(0) = u_0.$$

Approximating $u(t)$ relies on time-ordered integral:

$$u(t) = U_0(t)u_0 + \int_0^t U_s(t)b(s)\,ds,$$
$$U_s(t) \equiv \mathscr{T}\exp\left[\int_s^t A(s')\,ds'\right].$$

This is **not a matrix function**, but it is linear combination of **Hermitian** Ham sims [ACL23, LS25, SLBB25]

Simulation beyond QSVT: LCHS

$$\dot{u}(t) = -A(t)u(t) + b(t), \quad u(0) = u_0.$$

Approximating $u(t)$ relies on time-ordered integral:

$$u(t) = U_0(t)u_0 + \int_0^t U_s(t)b(s)\,ds,$$

$$U_s(t) \equiv \mathcal{T}\exp\left[\int_s^t A(s')\,ds'\right].$$

This is **not a matrix function**, but it is linear combination of **Hermitian** Ham sims [ACL23, LS25, SLBB25]

$$\forall x \geq 0,\ \ e^{-x} = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \hat{f}(k)\,e^{-ikx}\,dk$$

$$\Longrightarrow \ \forall t \geq 0,\ \ e^{-(L+iH)t} = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \hat{f}(k)\,e^{-i(kL+H)t}\,dk.$$

LCHS: utility and complexity

LCHS: utility and complexity

$$u_r \alpha t \log(1/\varepsilon), \quad u_r = \|u\|_{L^\infty} / |u(t)|$$

LCHS: utility and complexity

$$u_r \alpha t \log(1/\varepsilon), \quad u_r = \|u\|_{L^\infty} / |u(t)|$$

In addition to pole, $\hat{f}$ is chosen to have its inverse Fourier coefficients decay exponentially

LCHS: utility and complexity

$$u_r \alpha t \log(1/\varepsilon), \quad u_r = \|u\|_{L\infty} / |u(t)|$$

In addition to pole, $\hat{f}$ is chosen to have its inverse Fourier coefficients decay exponentially

The key insight of [LS25] is that this decay only needs occur on infinite strip, not the half plane

LCHS: utility and complexity

$$u_r \alpha t \log(1/\varepsilon), \quad u_r = \|u\|_{L^\infty} / |u(t)|$$

In addition to pole, $\hat{f}$ is chosen to have its inverse Fourier coefficients decay exponentially

The key insight of [LS25] is that this decay only needs occur on infinite strip, not the half plane

⚠️ **Integral quadrature can be technical [ACL23, TW14]**

LCHS: utility and complexity

$$u_r \alpha t \log(1/\varepsilon), \quad u_r = \|u\|_{L^\infty} / |u(t)|$$

In addition to pole, $\hat{f}$ is chosen to have its inverse Fourier coefficients decay exponentially

The key insight of [LS25] is that this decay only needs occur on infinite strip, not the half plane

⚠️ **Integral quadrature can be technical [ACL23, TW14]**

Beyond e^{-x}, approximations for $1/x$ permit linear matrix equation solvers [SLBB25], e.g., $AX - XB = C$

The quantum linear system problem: $Ax = b$

The quantum linear system problem: $Ax = b$

⚠️ QSVT exhibits suboptimal condition num dependence

The quantum linear system problem: $Ax = b$

⚠️ QSVT exhibits suboptimal condition num dependence

Algorithm	*Query comp.*	*N.b.*
HHL [HHL09]	$O(\kappa^2/\varepsilon)$	VTAA
LCU [CKS17]	$O(\kappa^2 \operatorname{polylog}(\varepsilon^{-1}))$	VTAA
QSVT [GSLW19]	$O(\kappa^2 \log(\varepsilon^{-1}))$	
VTAA [Amb10]	$O(\kappa \operatorname{polylog}(\kappa\varepsilon^{-1}))$	Overhead
Adiabatic [CAS+22]	$O(\kappa \operatorname{polylog}(\varepsilon^{-1}))$	
Eig. filter [LT20]	$O(\kappa \log(\varepsilon^{-1}))$	Adiabatic

The quantum linear system problem: $Ax = b$

⚠️ QSVT exhibits suboptimal condition num dependence

Algorithm	*Query comp.*	*N.b.*
HHL [HHL09]	$O(\kappa^2/\varepsilon)$	VTAA
LCU [CKS17]	$O(\kappa^2 \operatorname{polylog}(\varepsilon^{-1}))$	VTAA
QSVT [GSLW19]	$O(\kappa^2 \log(\varepsilon^{-1}))$	
VTAA [Amb10]	$O(\kappa \operatorname{polylog}(\kappa\varepsilon^{-1}))$	Overhead
Adiabatic [CAS+22]	$O(\kappa \operatorname{polylog}(\varepsilon^{-1}))$	
Eig. filter [LT20]	$O(\kappa \log(\varepsilon^{-1}))$	Adiabatic

Moral: block encoding A^{-1} requires subnormalizing by κ, meaning $O(\kappa)$ amp, and thus $O(\kappa^2)$ query comp.

⚖️ Tradeoffs in the great family

⚖️ Tradeoffs in the great family

1. QSVT depends on $\|f\|_\infty$, while LCU depends on $|f|_1$

⚖️ Tradeoffs in the great family

1. QSVT depends on $\| f \|_\infty$, while LCU depends on $| f |_1$
2. Bounded approx introduces logarithmic overhead

⚖️ Tradeoffs in the great family

1. QSVT depends on $||f||_\infty$, while LCU depends on $|f|_1$
2. Bounded approx introduces logarithmic overhead
3. Space use in QSVT (*post block encoding*) is constant in poly degree, while LCU's is logarithmic[1]

[1]For entire functions, this is doubly-log in error.

⚖️ Tradeoffs in the great family

1. QSVT depends on $\|f\|_\infty$, while LCU depends on $|f|_1$
2. Bounded approx introduces logarithmic overhead
3. Space use in QSVT (*post block encoding*) is constant in poly degree, while LCU's is logarithmic[1]
4. Ham sim for QSVT+ scales with spectral norm; in-place commutator-scaling (fancy) Trotter methods are (nearly) linear in time [CST+21, LKW19]

[1]For entire functions, this is doubly-log in error.

⚖️ Tradeoffs in the great family

1. QSVT depends on $\|f\|_\infty$, while LCU depends on $|f|_1$
2. Bounded approx introduces logarithmic overhead
3. Space use in QSVT (*post block encoding*) is constant in poly degree, while LCU's is logarithmic[1]
4. Ham sim for QSVT+ scales with spectral norm; in-place commutator-scaling (fancy) Trotter methods are (nearly) linear in time [CST+21, LKW19]

⚠️ **The utility of $\log(1/\varepsilon)$ error dependence outside subroutines is debatable**

[1]For entire functions, this is doubly-log in error.

🚡 Open avenues in the near-term

🚡 Open avenues in the near-term

Moving beyond spectral mapping	Weaker access models and hybrid methods	More flexible proof techniques

🚡 Open avenues in the near-term

Moving beyond spectral mapping	Weaker access models and hybrid methods	More flexible proof techniques

Past examples include **LCHS** [LS25], **multiproduct formulas** [LKW19], and **SKTs for QSP** [Ros25]

🚡 Open avenues in the near-term

Moving beyond spectral mapping	Weaker access models and hybrid methods	More flexible proof techniques

Past examples include **LCHS** [LS25], **multiproduct formulas** [LKW19], and **SKTs for QSP** [Ros25]

We need more *end-to-end analysis* for *realistic problem instances*! Most operations can be fuzzed!

🤖 Block encoding algorithms & machine learning

🤖 Block encoding algorithms & machine learning

Great platform for dequantization efforts [CGL+20]

🤖 Block encoding algorithms & machine learning

Great platform for dequantization efforts [CGL+20]

Investigations into quantum analogues of universal approximators [PSLNnGS+21, PSCLGFL20, Ros25]

🤖 Block encoding algorithms & machine learning

Great platform for dequantization efforts [CGL+20]

Investigations into quantum analogues of universal approximators [PSLNnGS+21, PSCLGFL20, Ros25]

Plenty of work left in realizing matrix manipulations efficiently (e.g., inference step in transformers [GYC+25])

🤖 Block encoding algorithms & machine learning

Great platform for dequantization efforts [CGL+20]

Investigations into quantum analogues of universal approximators [[illegible], GC+21, PSCLGFL20, Ros25]

⚠️ **As always, *caveat lector***

Plenty of work left in realizing matrix manipulations efficiently (e.g., inference step in transformers [GYC+25])

🏡 Take-home

🏡 Take-home

QSP/QSVT are one offshoot of a growing family of quantum algorithms for **efficiently manipulating sub-blocks of unitary matrices**

🏡 Take-home

QSP/QSVT are one offshoot of a growing family of quantum algorithms for **efficiently manipulating sub-blocks of unitary matrices**

These algorithms and their hybrids enable importation of *vast techniques from applied mathematics*

🏡 Take-home

QSP/QSVT are one offshoot of a growing family of quantum algorithms for **efficiently manipulating sub-blocks of unitary matrices**

These algorithms and their hybrids enable importation of *vast techniques from applied mathematics*

Regimes in which one method quantitatively outperforms another are *complex and evolving*

🏡 Take-home

QSP/QSVT are one offshoot of a growing family of quantum algorithms for **efficiently manipulating sub-blocks of unitary matrices**

These algorithms and their hybrids enable importation of *vast techniques from applied mathematics*

Regimes in which one method quantitatively outperforms another are *complex and evolving*

We have *limited tools beyond qubitization* for nonlinear transformations of subsystem dynamics

🏡 Take-home

QSP/QSVT are one offshoot of a growing family of quantum algorithms for **efficiently manipulating sub-blocks of unitary matrices**

These algorithms and their hybrids enable importation of *vast techniques from applied mathematics*

We're not yet sufficiently optimistic about these algorithms!

Regimes in which a method quantitatively outperforms another are *complex and evolving*

We have *limited tools beyond qubitization* for nonlinear transformations of subsystem dynamics fin

[ACL23] Dong An, Andrew M. Childs, and Lin Lin. Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters, 2023.

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

[Amb10] Andris Ambainis. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations, 2010.

[BCK15] Dominic W Berry, Andrew M Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th annual symposium on foundations of computer science*, pages 792–809. IEEE, 2015.

[BHC04] Kenneth R. Brown, Aram W. Harrow, and Isaac L. Chuang. Arbitrarily accurate composite pulse sequences. *Phys. Rev. A*, 70:052318, Nov 2004.

[CAS+22] Pedro C.S. Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry. Optimal scaling quantum linear-systems solver via discrete adiabatic theorem. *PRX Quantum*, 3:040303, 2022.

[CDG+20] Rui Chao, Dawei Ding, Andras Gilyen, Cupjin Huang, and Mario Szegedy. Finding angles for quantum signal processing with machine precision. *arXiv preprint arXiv:2003.02831*, 2020.

[CGL+20] Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 387 – 400, New York, NY, USA, 2020. Association for Computing Machinery.

[CKS17] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision. *SIAM Journal on Computing*, 46(6):1920 – 1950, 2017.

[CLVBY24] Daan Camps, Lin Lin, Roel Van Beeumen, and Chao Yang. Explicit quantum circuits for block encodings of certain sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 45(1):801–827, 2024.

[CST+21] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, Feb 2021.

[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.

[DLNW24a] Yulong Dong, Lin Lin, Hongkang Ni, and Jiasu Wang. Infinite quantum signal processing. *Quantum*, 8:1558, 2024.

[DLNW24b] Yulong Dong, Lin Lin, Hongkang Ni, and Jiasu Wang. Robust iterative method for symmetric quantum signal processing in all parameter regimes. *SIAM Journal on Scientific Computing*, 46(5):A2951–A2971, 2024.

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

[EHG25] Roman Edenhofer, Atsuya Hasegawa, and François Le Gall. Dequantization and hardness of spectral sum estimation, 2025.

[GLG23] Sevag Gharibian and François Le Gall. Dequantizing the quantum singular value transformation: Hardness and applications to quantum chemistry and the quantum pcp conjecture. *SIAM Journal on Computing*, 52(4):1009 – 1038, August 2023.

[GLM+22] András Gilyén, Seth Lloyd, Iman Marvian, Yihui Quek, and Mark M. Wilde. Quantum algorithm for petz recovery channels and pretty good measurements. *Physical Review Letters*, 128(22), 2022.

[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.

[GYC+25] Naixu Guo, Zhan Yu, Matthew Choi, Yizhan Han, Aman Agrawal, Kouhei Nakaji, Alán Aspuru-Guzik, and Patrick Rebentrost. Quantum transformer: Accelerating model inference via quantum linear algebra, 2025.

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

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), 2009.

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

[KLB+25] Robbie King, Guang Hao Low, Ryan Babbush, Rolando D. Somma, and Nicholas C. Rubin. Quantum simulation with sum-of-squares spectral amplification, 2025.

[KREO25] Parker Kuklinski, Benjamin Rempfer, Justin Elenewski, and Kevin Obenland. Efficient block-encodings require structure, 2025.

[KSV02] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

[LC17] G. H. Low and I. L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, 2017.

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

[Lin25] Lin Lin. Mathematical and numerical analysis of quantum signal processing, 2025.

[LKAS+21] Seth Lloyd, Bobak T. Kiani, David R. M. Arvidsson-Shukur, Samuel Bosch, Giacomo De Palma, William M. Kaminsky, Zi-Wen Liu, and Milad Marvian. Hamiltonian singular value transformation and inverse block encoding. *arXiv preprint arXiv:2104.01410*, 2021.

[LKW19] Guang Hao Low, Vadym Kliuchnikov, and Nathan Wiebe. Well-conditioned multiproduct hamiltonian simulation, 2019.

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631 – 633, July 2014.

[LMS21] Alex Lombardi, Fermi Ma, and Nicholas Spooner. Post-quantum zero knowledge, revisited (or: How to do quantum rewinding undetectably), 2021.

[LS24] Guang Hao Low and Yuan Su. Quantum eigenvalue processing. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, page 1051 – 1062. IEEE, October 2024.

[LS25] Guang Hao Low and Rolando D. Somma. Optimal quantum simulation of linear non-unitary dynamics, 2025.

[LSS+25] Yuan Liu, Shraddha Singh, Kevin C. Smith, Eleanor Crane, John M. Martyn, Alec Eickbusch, Alexander Schuckert, Richard D. Li, Jasmine Sinanan-Singh, Micheline B. Soley, Takahiro Tsunoda, Isaac L. Chuang, Nathan Wiebe, and Steven M. Girvin. Hybrid oscillator-qubit quantum processors: Instruction set architectures, abstract machine models, and applications. *PRX Quantum*, 2025.

[LT20] Lin Lin and Yu Tong. Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems. *Quantum*, 4:361, November 2020.

[LW19] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture, 2019.

[LYC14] Guang Hao Low, Theodore J Yoder, and Isaac L Chuang. Optimal arbitrarily accurate composite pulse sequences. *Phys. Rev. A*, 89(2):022341, 2014.

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

[MA24] C. Thiele M. Alexis, G. Mnatsakanyan. Quantum signal processing and nonlinear Fourier analysis. *Rev. Mat. Complut.*, pages 655–694, 2024.

[MLCC23] John M. Martyn, Yuan Liu, Zachary E. Chin, and Isaac L. Chuang. Efficient fully-coherent quantum signal processing algorithms for real-time dynamics simulation. *The Journal of Chemical Physics*, 158(2), 2023.

[MR25] John M. Martyn and Patrick Rall. Halving the cost of quantum algorithms with randomization. *npj Quantum Information*, 11(1), 2025.

[MRC+25] John M. Martyn, Zane M. Rossi, Kevin Z. Cheng, Yuan Liu, and Isaac L. Chuang. Parallel quantum signal processing via polynomial factorization. *Quantum*, 9:1834, August 2025.

[MRTC21] John M Martyn, Zane M Rossi, Andrew K Tan, and Isaac L Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4):040203, 2021.

[MS24] Ashley Montanaro and Changpeng Shao. Quantum and classical query complexities of functions of matrices. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC ’ 24, page 573 – 584. ACM, 2024.

[MW24] Danial Motlagh and Nathan Wiebe. Generalized quantum signal processing. *PRX Quantum*, 5:020368, 2024.

[NSYL25] Hongkang Ni, Rahul Sarkar, Lexing Ying, and Lin Lin. Inverse nonlinear fast fourier transform on su(2) with applications to quantum signal processing, 2025.

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

[PSCLGFL20] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

[PSLNnGS+21] Adrián Pérez-Salinas, David López-Núñez, Artur García-Sáez, P. Forn-Díaz, and José I. Latorre. One qubit as a universal approximant. *Phys. Rev. A*, 104:012405, 2021.

[PW94] C. Paige and M. Wei. History and generality of the CS decomposition. *Linear Algebra and its Applications*, 208-209:303–326, 1994.

[Ral21] Patrick Rall. Faster Coherent Quantum Algorithms for Phase, Energy, and Amplitude Estimation. *Quantum*, 5:566, October 2021.

[RBMC23] Zane M. Rossi, Victor M. Bastidas, William J. Munro, and Isaac L. Chuang. Quantum signal processing with continuous variables, 2023.

[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.

[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.

[Ros25] Zane M. Rossi. A solovay-kitaev theorem for quantum signal processing, 2025.

[SCC24] Christoph Sünderhauf, Earl Campbell, and Joan Camps. Block-encoding structured matrices for data input in quantum computing. *Quantum*, 8:1226, 2024.

[SLBB25] Rolando D. Somma, Guang Hao Low, Dominic W. Berry, and Ryan Babbush. Quantum algorithm for linear matrix equations, 2025.

[Sze04] Mario Szegedy. Quantum speed-up of markov chain based algorithms. In *In Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 32–41, 2004.

[Tan25] Xinyu Tan. Unitary synthesis with fewer t gates, 2025.

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

[TW14] Lloyd N. Trefethen and J. A. C. Weideman. The exponentially convergent trapezoidal rule. *SIAM Review*, 56(3):385–458, 2014.

[VC05] Lieven MK Vandersypen and Isaac L Chuang. NMR techniques for quantum control and computation. *RMP*, 76(4):1037, 2005.

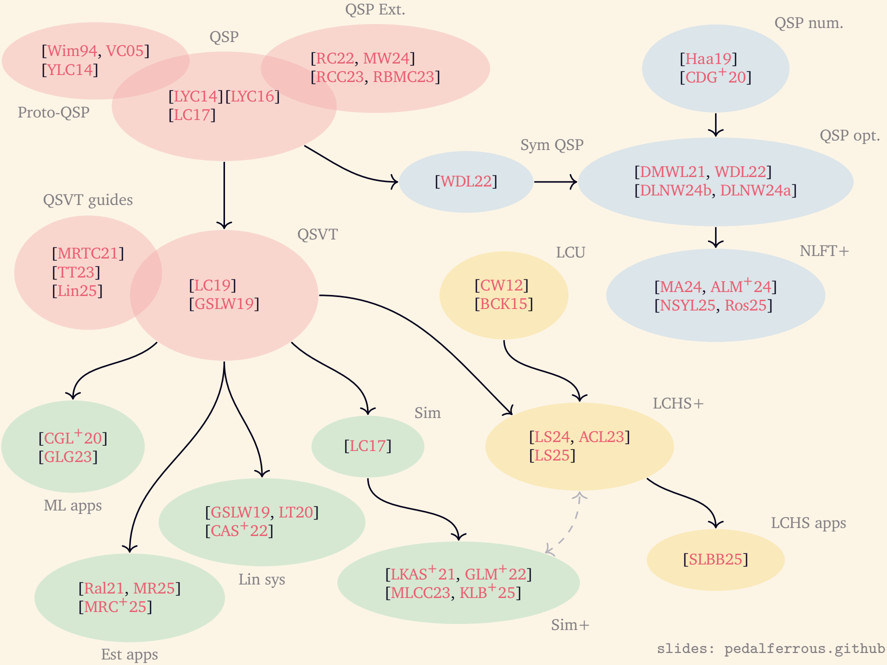
[VG25] Francisca Vasconcelos and András Gilyén. Methods for reducing ancilla-overhead in block encodings, 2025.

[WDL22] Jiasu Wang, Yulong Dong, and Lin Lin. On the energy landscape of symmetric quantum signal processing. *Quantum*, 6:850, 2022.

[Wim94] Stephen Wimperis. Broadband, narrowband, and passband composite pulses for use in advanced NMR experiments. *Journal of Magnetic Resonance, Series A*, 109(2):221–231, 1994.

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

✨ slides++



QSP Ext.
QSP
[Wim94, VC05]
[YLC14]
Proto-QSP
[LYC14][LYC16]
[LC17]
[RC22, MW24]
[RCC23, RBMC23]
QSP num.
[Haa19]
[CDG+20]
Sym QSP
[WDL22]
QSP opt.
[DMWL21, WDL22]
[DLNW24b, DLNW24a]
QSVT guides
[MRTC21]
[TT23]
[Lin25]
QSVT
[LC19]
[GSLW19]
LCU
[CW12]
[BCK15]
NLFT+
[MA24, ALM+24]
[NSYL25, Ros25]
[CGL+20]
[GLG23]
ML apps
Sim
[LC17]
LCHS+
[LS24, ACL23]
[LS25]
[GSLW19, LT20]
[CAS+22]
Lin sys
[LKAS+21, GLM+22]
[MLCC23, KLB+25]
Sim+
LCHS apps
[SLBB25]
[Ral21, MR25]
[MRC+25]
Est apps

Szegő functions

A real-valued measureable even function $f : [0,1] \to [-1,1]$ is called Szegő if it satisfies

$$\int_0^1 \log\left|1 - f(x)^2\right| \frac{dx}{\sqrt{1-x^2}} > -\infty \tag{9}$$

Szegő functions come with a norm showing they're a subset of square-summable

The work of [ALM+24] shows that Szegő functions satisfying $\|f\|_\infty \leq 1-\eta$ admit unique QSP phases, and produce a unitary with a matrix element converging to f in a nice way, and that

$$\|\Phi - \Phi'\|_\infty \leq O(\eta^{-3}) \|f - f'\|_S. \tag{10}$$

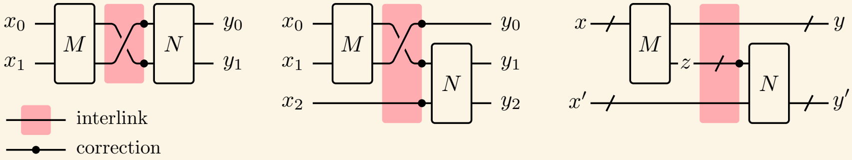
👹 Hybrid methods in block encoding algorithms

Hybrid oscillator-qubit systems in experiment [LSS+25]

LCU of product formulas [CST+21, LKW19]

Parallel, randomized, and self-composed QSP/QSVT [RCC23, MRC+25, MR25]

Approximate or accelerated products for block encodings [LW19, VG25]



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; [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 $\langle i|f(A)|j\rangle \pm \varepsilon/4$.