Quantum Singular Value Transformation

A unified framework for quantum algorithms

Jinge Bao

School of Informatics
University of Edinburgh

CDT QI Spring School

May 26, 2026

Topics to cover

- Lec 1: Quantum singular value transform (Tue)
- Lec 2: Quantum query lower bound (Wed)
- Lec 3: Quantum fine-grained complexity (Fri)

This lecture

1. Block Encodings
2. Quantum Signal Processing
3. Quantum Singular Value Transformation
4. Applications
5. Wrap-up

Goal. Understand the framework that unifies the major quantum algorithms.

- **Block encoding** — the right input model for matrices on a quantum computer.
- **QSP** — a single qubit + alternating rotations realises any admissible polynomial.
- **QSVT** — lift QSP to matrices: act on all singular values simultaneously.
- **Applications** — Hamiltonian simulation rover and HHL become choices of polynomial.

The unifying question

Given a state $|\psi\rangle$ and access to an operator A, how do we apply $f(A)$ to the state, query-efficiently?

$$|\psi\rangle \longmapsto f(A)\,|\psi\rangle\,.$$

The unifying question

Given a state $|\psi\rangle$ and access to an operator A, how do we apply $f(A)$ to the state, query-efficiently?

$$|\psi\rangle \longmapsto f(A)\,|\psi\rangle\,.$$

Many quantum algorithms have this shape with different f:

- **Grover.** f projects onto solutions.
- **HHL.** $f(x) = 1/x$.
- **Hamiltonian simulation.** $f(x) = e^{-itx}$.
- **Amplitude estimation.** A cos-shape locked to an unknown amplitude.

The unifying question

Given a state $|\psi\rangle$ and access to an operator A, how do we apply $f(A)$ to the state, query-efficiently?

$$|\psi\rangle \longmapsto f(A)\,|\psi\rangle\,.$$

Many quantum algorithms have this shape with different f:

- **Grover.** f projects onto solutions.
- **HHL.** $f(x) = 1/x$.
- **Hamiltonian simulation.** $f(x) = e^{-itx}$.
- **Amplitude estimation.** A cos-shape locked to an unknown amplitude.

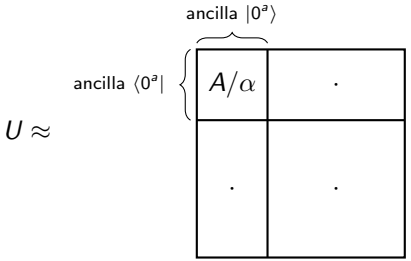
Answer: QSVT. Pick a polynomial P *approximating* f, apply it via a *block encoding* of A.

$\Rightarrow$ Quantum algorithm design reduces to *approximation theory*.

Block encodings: a uniform interface for matrix access

Definition. A unitary U on a ancilla + n system qubits is an (α, a, ε)-*block encoding* of A if

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

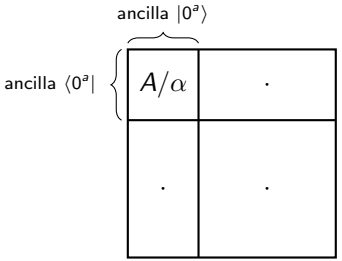


Block encodings: a uniform interface for matrix access

Definition. A unitary U on a ancilla + n system qubits is an (α, a, ε)-*block encoding* of A if

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

$U \approx$



Why this object.

- Standardises "access to a non-unitary matrix".
- Closed under products, sums, and tensor products.
- Many natural input models give block encodings.

Building block encodings

Sparse-access oracles. For s-sparse A with $\|A\|_{\max} \leq 1$, oracles for nonzero locations and entries give an $(s,\ O(\log N),\ 0)$-block encoding using $O(1)$ oracle queries (Childs '10; Berry–Childs–Kothari '15).

Building block encodings

Sparse-access oracles. For s-sparse A with $\|A\|_{\max} \leq 1$, oracles for nonzero locations and entries give an $(s,\ O(\log N),\ 0)$-block encoding using $O(1)$ oracle queries (Childs '10; Berry–Childs–Kothari '15).

Linear combination of unitaries (LCU). If $A = \sum_{k=1}^{L} \alpha_k U_k$ with $\alpha_k > 0$, $\alpha := \sum_k \alpha_k$:

$$\mathrm{PREP}\,|0\rangle = \tfrac{1}{\sqrt{\alpha}} \sum_k \sqrt{\alpha_k}\,|k\rangle\,, \qquad \mathrm{SEL} = \sum_k |k\rangle\langle k| \otimes U_k.$$

Then $(\mathrm{PREP}^\dagger \otimes \mathrm{Id})\,\mathrm{SEL}\,(\mathrm{PREP} \otimes \mathrm{Id})$ is an $(\alpha, \lceil \log L \rceil, 0)$-block encoding.

Building block encodings

Sparse-access oracles. For s-sparse A with $\|A\|_{\max} \leq 1$, oracles for nonzero locations and entries give an $(s,\ O(\log N),\ 0)$-block encoding using $O(1)$ oracle queries (Childs '10; Berry–Childs–Kothari '15).

Linear combination of unitaries (LCU). If $A = \sum_{k=1}^{L} \alpha_k U_k$ with $\alpha_k > 0$, $\alpha := \sum_k \alpha_k$:

$$\mathrm{PREP}\,|0\rangle = \tfrac{1}{\sqrt{\alpha}} \sum_k \sqrt{\alpha_k}\,|k\rangle\,, \qquad \mathrm{SEL} = \sum_k |k\rangle\langle k| \otimes U_k.$$

Then $(\mathrm{PREP}^\dagger \otimes \mathrm{Id})\,\mathrm{SEL}\,(\mathrm{PREP} \otimes \mathrm{Id})$ is an $(\alpha, \lceil \log L \rceil, 0)$-block encoding.

Density matrix. A state-prep unitary for $|\psi\rangle$ gives a $(1, n, 0)$-block encoding of $|\psi\rangle\langle\psi|$.

Building block encodings

Sparse-access oracles. For s-sparse A with $\|A\|_{\max} \leq 1$, oracles for nonzero locations and entries give an $(s,\ O(\log N),\ 0)$-block encoding using $O(1)$ oracle queries (Childs '10; Berry–Childs–Kothari '15).

Linear combination of unitaries (LCU). If $A = \sum_{k=1}^{L} \alpha_k U_k$ with $\alpha_k > 0$, $\alpha := \sum_k \alpha_k$:

$$\mathrm{PREP}\,|0\rangle = \tfrac{1}{\sqrt{\alpha}} \sum_k \sqrt{\alpha_k}\,|k\rangle\,, \qquad \mathrm{SEL} = \sum_k |k\rangle\langle k| \otimes U_k.$$

Then $(\mathrm{PREP}^\dagger \otimes \mathrm{Id})\,\mathrm{SEL}\,(\mathrm{PREP} \otimes \mathrm{Id})$ is an $(\alpha, \lceil \log L \rceil, 0)$-block encoding.

Density matrix. A state-prep unitary for $|\psi\rangle$ gives a $(1, n, 0)$-block encoding of $|\psi\rangle\langle\psi|$.

Hamiltonian from Pauli decomposition. $H = \sum_j \alpha_j P_j$ is a direct LCU, block-encoded with $\alpha = \sum_j |\alpha_j|$.

Quantum signal processing: a single qubit, alternating rotations

Signal unitary. For $x \in [-1, 1]$,

$$W(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} = e^{i \arccos(x) X}.$$

Quantum signal processing: a single qubit, alternating rotations

Signal unitary. For $x \in [-1, 1]$,

$$W(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} = e^{i \arccos(x) X}.$$

Phase unitary. For $\phi \in \mathbb{R}$,

$$D(\phi) = e^{i\phi Z}$$

Quantum signal processing: a single qubit, alternating rotations

Signal unitary. For $x \in [-1, 1]$,

$$W(x) \;=\; \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} \;=\; e^{i \arccos(x) X}.$$

Phase unitary. For $\phi \in \mathbb{R}$,

$$D(\phi) \;=\; e^{i\phi Z}$$

Processing unitary. Choose $\vec{\phi} = (\phi_0, \phi_1, \ldots, \phi_d) \in \mathbb{R}^{d+1}$ and form

$$U_{\vec{\phi}}(x) \;=\; D(\phi_0) \prod_{k=1}^{d} \big(W(x)\, D(\phi_k)\big).$$

Quantum signal processing: a single qubit, alternating rotations

Signal unitary. For $x \in [-1, 1]$,

$$W(x) \;=\; \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} \;=\; e^{i \arccos(x) X}.$$

Phase unitary. For $\phi \in \mathbb{R}$,

$$D(\phi) \;=\; e^{i\phi Z}$$

Processing unitary. Choose $\vec{\phi} = (\phi_0, \phi_1, \ldots, \phi_d) \in \mathbb{R}^{d+1}$ and form

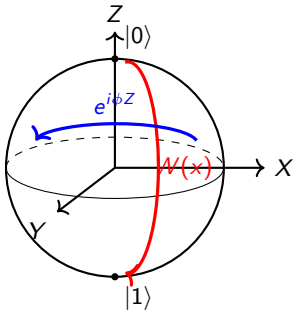
$$U_{\vec{\phi}}(x) \;=\; D(\phi_0) \prod_{k=1}^{d} \big(W(x)\, D(\phi_k)\big).$$

Lemma. The top-left entry of $U_{\vec{\phi}}(x)$ is a polynomial $P(x) \in \mathbb{C}[x]$ of degree $\leq d$ with $|P(x)| \leq 1$ on $[-1, 1]$.

d phase rotations interleaved with d signal calls $\Rightarrow$ degree-d polynomial.

QSP on the Bloch sphere

$W(x)$ rotates around X by $-2\arccos(x)$. $D(\phi)$ rotates around Z by -2ϕ.



Alternating X- and Z-rotations on a single qubit; $\langle 0| U_{\vec{\phi}}(x) |0\rangle$ encodes the resulting polynomial.

QSP achievability (Low–Chuang '17; GSLW '19)

Theorem. Any $P \in \mathbb{R}[x]$ of degree $\leq d$, definite parity, $|P| \leq 1$ on $[-1, 1]$ is realised by some $\vec{\phi}$ of length $d + 1$.

Phase computation. Given P, the phases $\vec{\phi}$ are computed classically in poly(d) (Haah '19, Dong–Meng–Whaley–Lin '21).

Lifting QSP to matrices: QSVT (GSLW '19)

Let U be an $(\alpha, a, 0)$-block encoding of A, with SVD

$$A/\alpha = \sum_k \sigma_k \, |u_k\rangle\langle v_k| \,, \qquad \sigma_k \in [0, 1].$$

Lifting QSP to matrices: QSVT (GSLW '19)

Let U be an $(\alpha, a, 0)$-block encoding of A, with SVD

$$A/\alpha \;=\; \sum_k \sigma_k \; |u_k\rangle\langle v_k| \,, \qquad \sigma_k \in [0,1].$$

Theorem (Gilyén–Su–Low–Wiebe '19). For real P of definite parity, degree d, $|P| \leq 1$ on $[-1,1]$, there is a circuit using d uses of $U, U^\dagger$ and $O(d(a+1))$ extra gates that block-encodes

$$P^{\mathrm{SV}}(A/\alpha) \;:=\; \begin{cases} \sum_k P(\sigma_k)\; |u_k\rangle\langle v_k| & d \text{ odd}, \\ \sum_k P(\sigma_k)\; |v_k\rangle\langle v_k| & d \text{ even}. \end{cases}$$

Lifting QSP to matrices: QSVT (GSLW '19)

Let U be an $(\alpha, a, 0)$-block encoding of A, with SVD

$$A/\alpha \;=\; \sum_k \sigma_k \, |u_k\rangle\langle v_k| \,, \qquad \sigma_k \in [0,1].$$

Theorem (Gilyén–Su–Low–Wiebe '19). For real P of definite parity, degree d, $|P| \leq 1$ on $[-1,1]$, there is a circuit using d uses of $U, U^\dagger$ and $O(d(a+1))$ extra gates that block-encodes

$$P^{\mathrm{SV}}(A/\alpha) \;:=\; \begin{cases} \sum_k P(\sigma_k)\, |u_k\rangle\langle v_k| & d \text{ odd}, \\ \sum_k P(\sigma_k)\, |v_k\rangle\langle v_k| & d \text{ even}. \end{cases}$$

In words. What a degree-d polynomial does to a scalar in $[-1,1]$, QSVT does to every singular value of a block-encoded matrix, using d oracle calls.

QSVT circuit

Let $\Pi = |0^a\rangle\langle 0^a| \otimes \mathrm{Id}$. Define the projector-controlled phase

$$\Pi_\phi \;=\; e^{i\phi\,(2\Pi - \mathrm{Id})}.$$

This applies phase $e^{i\phi}$ to the $\Pi = 1$ subspace and $e^{-i\phi}$ to the $\Pi = 0$ subspace (proof in appendix).

QSVT circuit

Let $\Pi = |0^a\rangle\langle 0^a| \otimes \mathrm{Id}$. Define the projector-controlled phase

$$\Pi_\phi \;=\; e^{i\phi\,(2\Pi - \mathrm{Id})}.$$

This applies phase $e^{i\phi}$ to the $\Pi = 1$ subspace and $e^{-i\phi}$ to the $\Pi = 0$ subspace (proof in appendix).

QSVT circuit (even d).

$$U_{\vec{\phi}} \;=\; \Pi_{\phi_0} \prod_{k=1}^{d/2} \left(U^\dagger\, \Pi_{\phi_{2k-1}}\, U\, \Pi_{\phi_{2k}}\right).$$

Odd d: alternate one extra U at the front.

QSVT circuit

Let $\Pi = |0^a\rangle\langle 0^a| \otimes \mathrm{Id}$. Define the projector-controlled phase

$$\Pi_\phi \;=\; e^{i\phi\,(2\Pi - \mathrm{Id})}.$$

This applies phase $e^{i\phi}$ to the $\Pi = 1$ subspace and $e^{-i\phi}$ to the $\Pi = 0$ subspace (proof in appendix).

QSVT circuit (even d).

$$U_{\vec{\phi}} \;=\; \Pi_{\phi_0} \prod_{k=1}^{d/2} \left(U^\dagger\, \Pi_{\phi_{2k-1}}\, U\, \Pi_{\phi_{2k}}\right).$$

Odd d: alternate one extra U at the front.

Counting. d uses of $U/U^\dagger$, $d+1$ projector-controlled phases. The latter are a-controlled Pauli rotations: $O(a)$ gates each.

QSVT circuit

Let $\Pi = |0^a\rangle\langle 0^a| \otimes \mathrm{Id}$. Define the projector-controlled phase

$$\Pi_\phi \;=\; e^{i\phi\,(2\Pi - \mathrm{Id})}.$$

This applies phase $e^{i\phi}$ to the $\Pi = 1$ subspace and $e^{-i\phi}$ to the $\Pi = 0$ subspace (proof in appendix).

QSVT circuit (even d).

$$U_{\vec{\phi}} \;=\; \Pi_{\phi_0} \prod_{k=1}^{d/2} \left(U^\dagger\, \Pi_{\phi_{2k-1}}\, U\, \Pi_{\phi_{2k}}\right).$$

Odd d: alternate one extra U at the front.

Counting. d uses of $U/U^\dagger$, $d+1$ projector-controlled phases. The latter are a-controlled Pauli rotations: $O(a)$ gates each.

Why it works. The block-encoded operator A/α acts inside a family of 2D invariant subspaces, in each of which it works like the QSP.

Qubitization step 1: what U does to an eigenvector

Hermitian case. We prove the result for Hermitian A (eigendecomposition), which covers all applications in this lecture.

The general SVD case uses the Cosine-Sine decomposition (GSLW '19).

Qubitization step 1: what U does to an eigenvector

Hermitian case. We prove the result for Hermitian A (eigendecomposition), which covers all applications in this lecture.

The general SVD case uses the Cosine-Sine decomposition (GSLW '19).

Let A/α be Hermitian with eigendecomposition $A/\alpha = \sum_k \lambda_k |v_k\rangle\langle v_k|$, $\lambda_k \in [-1, 1]$.

Embed: $|\tilde{v}_k\rangle := |0^a\rangle |v_k\rangle$.

By the block-encoding definition, $(\langle 0^a| \otimes \mathrm{Id})\, U\, (|0^a\rangle \otimes \mathrm{Id}) = A/\alpha$, so the $|0^a\rangle$-component of $U|\tilde{v}_k\rangle$ is $\lambda_k |v_k\rangle$.

Qubitization step 1: what U does to an eigenvector

Hermitian case. We prove the result for Hermitian A (eigendecomposition), which covers all applications in this lecture.

The general SVD case uses the Cosine-Sine decomposition (GSLW '19).

Let A/α be Hermitian with eigendecomposition $A/\alpha = \sum_k \lambda_k |v_k\rangle\langle v_k|$, $\lambda_k \in [-1, 1]$.

Embed: $|\tilde{v}_k\rangle := |0^a\rangle |v_k\rangle$.

By the block-encoding definition, $(\langle 0^a| \otimes \mathrm{Id})\, U\, (|0^a\rangle \otimes \mathrm{Id}) = A/\alpha$, so the $|0^a\rangle$-component of $U|\tilde{v}_k\rangle$ is $\lambda_k |v_k\rangle$.

Since U is unitary, the remainder has norm $\sqrt{1-\lambda_k^2}$ and lies outside the $|0^a\rangle$ subspace:

$$U|\tilde{v}_k\rangle \;=\; \lambda_k\, |\tilde{v}_k\rangle \;+\; \sqrt{1-\lambda_k^2}\, |\perp_k\rangle\,, \qquad \Pi|\perp_k\rangle = 0.$$

Qubitization step 2: the 2D invariant subspace

$\mathcal{H}_k := \mathrm{span}\{|\tilde{v}_k\rangle\,,\,|\perp_k\rangle\}$ is invariant under both U and Π_ϕ:

Qubitization step 2: the 2D invariant subspace

$\mathcal{H}_k := \mathrm{span}\{|\tilde{v}_k\rangle\,,\,|\perp_k\rangle\}$ is invariant under both U and Π_ϕ:

U in this basis.

$$U\big|_{\mathcal{H}_k} = \begin{pmatrix} \lambda_k & \sqrt{1-\lambda_k^2} \\ \sqrt{1-\lambda_k^2} & -\lambda_k \end{pmatrix}.$$

This is a reflection ($\det = -1$), *not* the same as the QSP signal $W(\lambda_k)$ (which has $\det = +1$).

Qubitization step 2: the 2D invariant subspace

$\mathcal{H}_k := \mathrm{span}\{|\tilde{v}_k\rangle\,,\ |\perp_k\rangle\}$ is invariant under both U and Π_ϕ:

U in this basis.

$$U\big|_{\mathcal{H}_k} = \begin{pmatrix} \lambda_k & \sqrt{1-\lambda_k^2} \\ \sqrt{1-\lambda_k^2} & -\lambda_k \end{pmatrix}.$$

This is a reflection ($\det = -1$), *not* the same as the QSP signal $W(\lambda_k)$ (which has $\det = +1$).

Π_ϕ in this basis.

$$\Pi_\phi\big|_{\mathcal{H}_k} = \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix} = D(\phi), \quad \text{exactly the QSP phase rotation.}$$

$\Rightarrow$ Inside each $\mathcal{H}_k$, the QSVT circuit is a QSP sequence with signal λ_k.

QSVT = QSP run in parallel

Inside each $\mathcal{H}_k$, the QSVT alternation $\Pi_\phi U^\dagger \Pi_\phi U$ acts exactly like QSP rotations after a basis change.

QSVT = QSP run in parallel

Inside each $\mathcal{H}_k$, the QSVT alternation $\Pi_\phi U^\dagger \Pi_\phi U$ acts exactly like QSP rotations after a basis change.

Consequence. Applying the full circuit $U_{\vec{\phi}}$:

- On $|\tilde{v}_k\rangle$: produces $P(\lambda_k)\,|\tilde{v}_k\rangle$ + (orthogonal junk).
- On any input $|0^a\rangle\,|\psi\rangle$ with $|\psi\rangle = \sum_k \beta_k\,|v_k\rangle$: produces $|0^a\rangle \sum_k \beta_k P(\lambda_k)\,|v_k\rangle$ in the encoded block, plus garbage on orthogonal ancilla states.

Thus $U_{\vec{\phi}}$ is a block encoding of $P(A/\alpha) = \sum_k P(\lambda_k)\,|v_k\rangle\langle v_k|$. $\square$

Polynomial approximation is all you need

Three workhorses:

- **Sign function.** $\exists\, P$ of degree $O(\delta^{-1}\log(1/\varepsilon))$ with $|P(x) - \mathrm{sgn}(x)| \leq \varepsilon$ on $[-1, -\delta] \cup [\delta, 1]$.

Polynomial approximation is all you need

Three workhorses:

- **Sign function.** $\exists\, P$ of degree $O(\delta^{-1}\log(1/\varepsilon))$ with $|P(x) - \mathrm{sgn}(x)| \leq \varepsilon$ on $[-1,-\delta] \cup [\delta, 1]$.
- **Inverse function.** On $[\kappa^{-1}, 1]$, $\exists\, P$ of degree $O(\kappa \log(\kappa/\varepsilon))$ with $|P(x) - 1/(2\kappa x)| \leq \varepsilon$.

Polynomial approximation is all you need

Three workhorses:

- **Sign function.** $\exists\, P$ of degree $O(\delta^{-1}\log(1/\varepsilon))$ with $|P(x) - \mathrm{sgn}(x)| \le \varepsilon$ on $[-1,-\delta]\cup[\delta,1]$.
- **Inverse function.** On $[\kappa^{-1},1]$, $\exists\, P$ of degree $O(\kappa\log(\kappa/\varepsilon))$ with $|P(x) - 1/(2\kappa x)| \le \varepsilon$.
- **Cosine / sine (Jacobi–Anger).**

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

truncate at $d = O\big(|t| + \log(1/\varepsilon)/\log(\log(1/\varepsilon)/|t|)\big)$.

Polynomial approximation is all you need

Three workhorses:

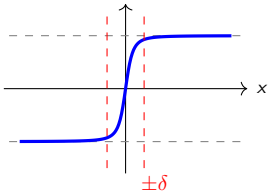
- **Sign function.** $\exists\, P$ of degree $O(\delta^{-1}\log(1/\varepsilon))$ with $|P(x) - \mathrm{sgn}(x)| \le \varepsilon$ on $[-1,-\delta]\cup[\delta,1]$.
- **Inverse function.** On $[\kappa^{-1},1]$, $\exists\, P$ of degree $O(\kappa\log(\kappa/\varepsilon))$ with $|P(x) - 1/(2\kappa x)| \le \varepsilon$.
- **Cosine / sine (Jacobi–Anger).**

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

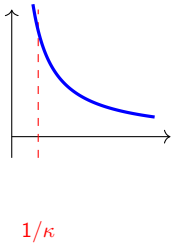
truncate at $d = O\big(|t| + \log(1/\varepsilon)/\log(\log(1/\varepsilon)/|t|)\big)$.

Each function: smooth $\Rightarrow$ exponentially decaying Chebyshev coefficients $\Rightarrow$ low-degree polynomial approximation.

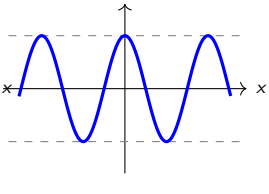
Polynomial gallery



Smoothed sgn(x)



$1/x$ on $[\kappa^{-1}, 1]$



cos(tx)

- sgn $\rightarrow$ amplitude amplification & eigenvalue threshold.
- $1/x \rightarrow$ matrix inversion (HHL).
- cos / sin $\rightarrow$ Hamiltonian simulation.

Application I: Hamiltonian simulation — the idea

Problem. Given a Hamiltonian H with eigendecomposition $H = \sum_k E_k |v_k\rangle\langle v_k|$, implement time evolution e^{-iHt} on a quantum state $|\psi\rangle$.

Application I: Hamiltonian simulation — the idea

Problem. Given a Hamiltonian H with eigendecomposition $H = \sum_k E_k |v_k\rangle\langle v_k|$, implement time evolution e^{-iHt} on a quantum state $|\psi\rangle$.

What does e^{-iHt} do? It applies $f(E_k) = e^{-iE_k t}$ to each eigenvalue:

$$|\psi\rangle = \sum_k \beta_k |v_k\rangle \quad \longmapsto \quad e^{-iHt}|\psi\rangle = \sum_k e^{-iE_k t}\beta_k |v_k\rangle .$$

Application I: Hamiltonian simulation — the idea

Problem. Given a Hamiltonian H with eigendecomposition $H = \sum_k E_k |v_k\rangle\langle v_k|$, implement time evolution e^{-iHt} on a quantum state $|\psi\rangle$.

What does e^{-iHt} do? It applies $f(E_k) = e^{-iE_k t}$ to each eigenvalue:

$$|\psi\rangle = \sum_k \beta_k |v_k\rangle \quad \longmapsto \quad e^{-iHt}|\psi\rangle = \sum_k e^{-iE_k t}\beta_k |v_k\rangle .$$

Key insight. This is a singular value transformation! We need to apply $f(x) = e^{-ixt}$ to each eigenvalue. QSVT can apply any polynomial — so approximate e^{-ixt} by a polynomial.

Application I: Hamiltonian simulation — the idea

Problem. Given a Hamiltonian H with eigendecomposition $H = \sum_k E_k |v_k\rangle\langle v_k|$, implement time evolution e^{-iHt} on a quantum state $|\psi\rangle$.

What does e^{-iHt} do? It applies $f(E_k) = e^{-iE_k t}$ to each eigenvalue:

$$|\psi\rangle = \sum_k \beta_k |v_k\rangle \quad \longmapsto \quad e^{-iHt}|\psi\rangle = \sum_k e^{-iE_k t}\beta_k |v_k\rangle .$$

Key insight. This is a singular value transformation! We need to apply $f(x) = e^{-ixt}$ to each eigenvalue. QSVT can apply any polynomial — so approximate e^{-ixt} by a polynomial.

Which polynomial? Use Euler's formula: $e^{-ixt} = \cos(xt) - i\sin(xt)$. Both $\cos(xt)$ and $\sin(xt)$ are well-approximated by Jacobi–Anger (Chebyshev) expansions of degree $d = O(\alpha t + \log(1/\varepsilon))$.

Application I: Hamiltonian simulation — the idea

Problem. Given a Hamiltonian H with eigendecomposition $H = \sum_k E_k |v_k\rangle\langle v_k|$, implement time evolution e^{-iHt} on a quantum state $|\psi\rangle$.

What does e^{-iHt} do? It applies $f(E_k) = e^{-iE_k t}$ to each eigenvalue:

$$|\psi\rangle = \sum_k \beta_k |v_k\rangle \quad \longmapsto \quad e^{-iHt}|\psi\rangle = \sum_k e^{-iE_k t}\beta_k |v_k\rangle .$$

Key insight. This is a singular value transformation! We need to apply $f(x) = e^{-ixt}$ to each eigenvalue. QSVT can apply any polynomial — so approximate e^{-ixt} by a polynomial.

Which polynomial? Use Euler's formula: $e^{-ixt} = \cos(xt) - i\sin(xt)$. Both $\cos(xt)$ and $\sin(xt)$ are well-approximated by Jacobi–Anger (Chebyshev) expansions of degree $d = O(\alpha t + \log(1/\varepsilon))$.

Why two polynomials? QSVT applies *real* polynomials with definite parity. cos is even, and sin is odd — each is handled by a separate QSVT circuit, then combined via LCU: $e^{-iHt} = \cos(Ht) - i\sin(Ht)$.

Application: Hamiltonian simulation — the recipe

Setup. H Hermitian, $\|H\| \leq 1$, $(\alpha, a, 0)$-block encoding U_H.

Application: Hamiltonian simulation — the recipe

Setup. H Hermitian, $\|H\| \leq 1$, $(\alpha, a, 0)$-block encoding U_H.

Recipe.

1. Approximate $\cos(\alpha t x)$ and $\sin(\alpha t x)$ by Jacobi–Anger truncations of degree $d = O(\alpha t + \log(1/\varepsilon))$.

Application: Hamiltonian simulation — the recipe

Setup. H Hermitian, $\|H\| \leq 1$, $(\alpha, a, 0)$-block encoding U_H.

Recipe.

1. Approximate $\cos(\alpha t x)$ and $\sin(\alpha t x)$ by Jacobi–Anger truncations of degree $d = O(\alpha t + \log(1/\varepsilon))$.
2. QSVT with even-parity phases $\Rightarrow$ block encoding of $\cos(Ht)$.

Application: Hamiltonian simulation — the recipe

Setup. H Hermitian, $\|H\| \leq 1$, $(\alpha, a, 0)$-block encoding U_H.

Recipe.

1. Approximate $\cos(\alpha t x)$ and $\sin(\alpha t x)$ by Jacobi–Anger truncations of degree $d = O(\alpha t + \log(1/\varepsilon))$.
2. QSVT with even-parity phases $\Rightarrow$ block encoding of $\cos(Ht)$.
3. QSVT with odd-parity phases $\Rightarrow$ block encoding of $\sin(Ht)$.

Application: Hamiltonian simulation — the recipe

Setup. H Hermitian, $\|H\| \leq 1$, $(\alpha, a, 0)$-block encoding U_H.

Recipe.

1. Approximate $\cos(\alpha t x)$ and $\sin(\alpha t x)$ by Jacobi–Anger truncations of degree $d = O(\alpha t + \log(1/\varepsilon))$.
2. QSVT with even-parity phases $\Rightarrow$ block encoding of $\cos(Ht)$.
3. QSVT with odd-parity phases $\Rightarrow$ block encoding of $\sin(Ht)$.
4. LCU combination: $e^{-iHt} = \cos(Ht) - i \sin(Ht)$.

Application: Hamiltonian simulation — the recipe

Setup. H Hermitian, $\|H\| \leq 1$, $(\alpha, a, 0)$-block encoding U_H.

Recipe.

1. Approximate $\cos(\alpha t x)$ and $\sin(\alpha t x)$ by Jacobi–Anger truncations of degree $d = O(\alpha t + \log(1/\varepsilon))$.
2. QSVT with even-parity phases $\Rightarrow$ block encoding of $\cos(Ht)$.
3. QSVT with odd-parity phases $\Rightarrow$ block encoding of $\sin(Ht)$.
4. LCU combination: $e^{-iHt} = \cos(Ht) - i \sin(Ht)$.

Cost. $O(\alpha t + \log(1/\varepsilon))$ queries to U_H.

Application: Hamiltonian simulation — the recipe

Setup. H Hermitian, $\|H\| \leq 1$, $(\alpha, a, 0)$-block encoding U_H.

Recipe.

1. Approximate $\cos(\alpha t x)$ and $\sin(\alpha t x)$ by Jacobi–Anger truncations of degree $d = O(\alpha t + \log(1/\varepsilon))$.
2. QSVT with even-parity phases $\Rightarrow$ block encoding of $\cos(Ht)$.
3. QSVT with odd-parity phases $\Rightarrow$ block encoding of $\sin(Ht)$.
4. LCU combination: $e^{-iHt} = \cos(Ht) - i \sin(Ht)$.

Cost. $O(\alpha t + \log(1/\varepsilon))$ queries to U_H.

Optimality. Matches the no-fast-forwarding lower bound: $\Omega(\alpha t + \log(1/\varepsilon)/\log\log(1/\varepsilon))$ (Berry–Ahokas–Cleve–Sanders '07). The cost is *linear* in time t and *logarithmic* in precision $1/\varepsilon$.

Application II: matrix inversion — the idea

Problem. Given A with eigenvalues $\lambda_k \in [\kappa^{-1}, 1]$ and a state $|b\rangle = \sum_k \beta_k |v_k\rangle$, prepare $|x\rangle \propto A^{-1}|b\rangle = \sum_k \frac{\beta_k}{\lambda_k} |v_k\rangle$.

Application II: matrix inversion — the idea

Problem. Given A with eigenvalues $\lambda_k \in [\kappa^{-1}, 1]$ and a state $|b\rangle = \sum_k \beta_k |v_k\rangle$, prepare $|x\rangle \propto A^{-1}|b\rangle = \sum_k \frac{\beta_k}{\lambda_k} |v_k\rangle$.

Key insight. Matrix inversion = applying $f(x) = 1/x$ to each eigenvalue. QSVT can apply any polynomial to eigenvalues — so approximate $1/x$ by a polynomial!

Application II: matrix inversion — the idea

Problem. Given A with eigenvalues $\lambda_k \in [\kappa^{-1}, 1]$ and a state $|b\rangle = \sum_k \beta_k |v_k\rangle$, prepare $|x\rangle \propto A^{-1}|b\rangle = \sum_k \frac{\beta_k}{\lambda_k}|v_k\rangle$.

Key insight. Matrix inversion = applying $f(x) = 1/x$ to each eigenvalue. QSVT can apply any polynomial to eigenvalues — so approximate $1/x$ by a polynomial!

Why $P(x) \approx 1/(2\kappa x)$ and not $1/x$? QSVT requires $|P(x)| \leq 1$ on $[-1, 1]$. But $1/x$ blows up near $x = 0$. The smallest eigenvalue is κ^{-1}, so $\max_{x \in [\kappa^{-1}, 1]} 1/x = \kappa$. We rescale: $P(x) \approx 1/(2\kappa x)$ has $|P| \leq 1/2 < 1$ on $[\kappa^{-1}, 1]$.

Application II: matrix inversion — the idea

Problem. Given A with eigenvalues $\lambda_k \in [\kappa^{-1}, 1]$ and a state $|b\rangle = \sum_k \beta_k |v_k\rangle$, prepare $|x\rangle \propto A^{-1}|b\rangle = \sum_k \frac{\beta_k}{\lambda_k}|v_k\rangle$.

Key insight. Matrix inversion = applying $f(x) = 1/x$ to each eigenvalue. QSVT can apply any polynomial to eigenvalues — so approximate $1/x$ by a polynomial!

Why $P(x) \approx 1/(2\kappa x)$ and not $1/x$? QSVT requires $|P(x)| \leq 1$ on $[-1, 1]$. But $1/x$ blows up near $x = 0$. The smallest eigenvalue is κ^{-1}, so $\max_{x \in [\kappa^{-1}, 1]} 1/x = \kappa$. We rescale: $P(x) \approx 1/(2\kappa x)$ has $|P| \leq 1/2 < 1$ on $[\kappa^{-1}, 1]$.

What QSVT gives us. Apply P^{SV} to the block-encoded A:

$$A = \sum_k \lambda_k |v_k\rangle\langle v_k| \quad \xrightarrow{\text{QSVT}} \quad P^{\mathrm{SV}}(A) = \sum_k \frac{1}{2\kappa\lambda_k} |v_k\rangle\langle v_k| = \frac{1}{2\kappa} A^{-1}.$$

Application II: matrix inversion — the recipe

Setup. $\|A\| \leq 1$, $(1, a, 0)$-block encoding of A.

Application II: matrix inversion — the recipe

Setup. $\|A\| \leq 1$, $(1, a, 0)$-block encoding of A.

Recipe.

1. Choose P of degree $O(\kappa \log(\kappa/\varepsilon))$ with $P(x) \approx 1/(2\kappa x)$ on $[\kappa^{-1}, 1]$.

Application II: matrix inversion — the recipe

Setup. $\|A\| \leq 1$, $(1, a, 0)$-block encoding of A.

Recipe.

1. Choose P of degree $O(\kappa \log(\kappa/\varepsilon))$ with $P(x) \approx 1/(2\kappa x)$ on $[\kappa^{-1}, 1]$.
2. QSVT $\Rightarrow$ block encoding of $\frac{1}{2\kappa} A^{-1}$.

Application II: matrix inversion — the recipe

Setup. $\|A\| \leq 1$, $(1, a, 0)$-block encoding of A.

Recipe.

1. Choose P of degree $O(\kappa \log(\kappa/\varepsilon))$ with $P(x) \approx 1/(2\kappa x)$ on $[\kappa^{-1}, 1]$.
2. QSVT $\Rightarrow$ block encoding of $\frac{1}{2\kappa} A^{-1}$.
3. Apply to $|b\rangle$: the encoded block gives $\frac{1}{2\kappa} A^{-1} |b\rangle$, but only with probability $\sim 1/\kappa^2$ (from the $1/(2\kappa)$ scaling).

Application II: matrix inversion — the recipe

Setup. $\|A\| \leq 1$, $(1, a, 0)$-block encoding of A.

Recipe.

1. Choose P of degree $O(\kappa \log(\kappa/\varepsilon))$ with $P(x) \approx 1/(2\kappa x)$ on $[\kappa^{-1}, 1]$.
2. QSVT $\Rightarrow$ block encoding of $\frac{1}{2\kappa} A^{-1}$.
3. Apply to $|b\rangle$: the encoded block gives $\frac{1}{2\kappa} A^{-1} |b\rangle$, but only with probability $\sim 1/\kappa^2$ (from the $1/(2\kappa)$ scaling).
4. Boost success probability with $O(\kappa)$ rounds of fixed-point amplitude amplification.

Application II: matrix inversion — the recipe

Setup. $\|A\| \leq 1$, $(1, a, 0)$-block encoding of A.

Recipe.

1. Choose P of degree $O(\kappa \log(\kappa/\varepsilon))$ with $P(x) \approx 1/(2\kappa x)$ on $[\kappa^{-1}, 1]$.
2. QSVT $\Rightarrow$ block encoding of $\frac{1}{2\kappa} A^{-1}$.
3. Apply to $|b\rangle$: the encoded block gives $\frac{1}{2\kappa} A^{-1} |b\rangle$, but only with probability $\sim 1/\kappa^2$ (from the $1/(2\kappa)$ scaling).
4. Boost success probability with $O(\kappa)$ rounds of fixed-point amplitude amplification.

Total cost. $\tilde{O}(\kappa \log(1/\varepsilon))$ queries to A, $O(\kappa)$ to $|b\rangle$.

Application II: matrix inversion — the recipe

Setup. $\|A\| \leq 1$, $(1, a, 0)$-block encoding of A.

Recipe.

1. Choose P of degree $O(\kappa \log(\kappa/\varepsilon))$ with $P(x) \approx 1/(2\kappa x)$ on $[\kappa^{-1}, 1]$.
2. QSVT $\Rightarrow$ block encoding of $\frac{1}{2\kappa} A^{-1}$.
3. Apply to $|b\rangle$: the encoded block gives $\frac{1}{2\kappa} A^{-1} |b\rangle$, but only with probability $\sim 1/\kappa^2$ (from the $1/(2\kappa)$ scaling).
4. Boost success probability with $O(\kappa)$ rounds of fixed-point amplitude amplification.

Total cost. $\tilde{O}(\kappa \log(1/\varepsilon))$ queries to A, $O(\kappa)$ to $|b\rangle$.

Caveats.

- State prep for $|b\rangle$ can dominate the cost.
- Read-out: you get $|x\rangle \propto A^{-1} |b\rangle$ as a quantum state. Extracting a single entry costs $O(1)$ measurements, but reading all N entries costs $O(N)$ — no exponential speedup there.
- Dequantisation: for low-rank A, classical algorithms match this in poly(κ) (Tang '19).

Why QSVT is the good framework

Unification table.

Algorithm	Target function	Degree	Prior framework
Amplitude amplification	$\mathrm{sgn}(x)$ (threshold at $\sqrt{a}$)	$O(1/\sqrt{a})$	Grover '96
Hamiltonian simulation	$e^{-itx} = \cos(tx) - i\sin(tx)$	$O(\alpha t + \log \frac{1}{\varepsilon})$	Product formulae
Matrix inversion	$1/x$ on $[\kappa^{-1}, 1]$	$O(\kappa \log \frac{\kappa}{\varepsilon})$	HHL '09
Quantum walk search	$\mathrm{sgn}(x)$ (threshold at $\sqrt{\delta}$)	$O(1/\sqrt{\delta})$	Szegedy '04

Optimality. Each application matches its known query lower bound (BBBV; BACS '07; HHL hardness).

If your task is "apply $f(A)$ to $|\psi\rangle$" and f admits a low-degree polynomial approximation, you have a quantum algorithm.

Take-aways

- **Block encoding** is the right input model for non-unitary operators.
- **QSP** realises any admissible polynomial of a ± 1-bounded scalar via d alternating rotations of a single qubit.
- **QSVT** lifts QSP to act on all singular values of a block-encoded matrix simultaneously. The mechanism is the 2D invariant-subspace decomposition ("qubitization").
- All of Grover, HHL, Hamiltonian simulation, and amplitude estimation become a *choice of polynomial*.
- Cost is dictated by approximation theory
- The resulting algorithms are typically query-optimal. (See next lecture)

Further reading

- Gilyén, Su, Low, Wiebe. *Quantum singular value transformation and beyond.* STOC '19.
- Low, Chuang. *Hamiltonian simulation by qubitization.* Quantum '19.
- Low, Chuang. *Hamiltonian simulation by uniform spectral amplification.* '17.
- Martyn, Rossi, Tan, Chuang. *Grand unification of quantum algorithms.* PRX Quantum '21.
- Lin. *Lecture notes on quantum algorithms for scientific computation.* 2022.
- Tang, Tian. *A CS guide to the quantum singular value transformation.* SOSA '24.
- Dong, Meng, Whaley, Lin. *Efficient phase-factor evaluation in quantum signal processing.* PRA '21.
- Haah. *Product decomposition of periodic functions in quantum signal processing.* Quantum '19.

Questions?