CS 59300 – Algorithms for Data Science

Classical and Quantum approaches

Lecture 17 (11/6)

Quantum linear algebra toolkits (II)

https://ruizhezhang.com/course_fall_2025.html

The slides are borrowed from Dong An's talk at IPAM

Quantum linear algebra toolbox

- Basic linear algebra operations
  - Input models for vectors and matrices
  - Matrix-vector multiplication
  - Matrix/vector addition: linear combination of unitaries (LCU)
  - Matrix multiplication
- Linear systems of equations

Quantum linear system problem (QLSP)

Classical version: given an N-by-N Hermitian matrix A, and a vector b, compute

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

- For non-Hermitian A, consider dilation: $\begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix} \begin{bmatrix} 0 \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$

Quantum version: given a block-encoding for A (assuming $\|A\| = 1$) and a state preparation oracle for $|b\rangle$, compute an ϵ-approximation of the quantum state:

$$\| |\tilde{x}\rangle - |x\rangle \| \leq \epsilon \qquad |x\rangle = \frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}$$

Parameters:

- Condition number: $\kappa \coloneqq \|A\|\|A^{-1}\| = \|A^{-1}\|$
- Accuracy ϵ

Harrow-Hassidim-Lloyd (HHL)

QLSP as an eigenvalue transformation problem:

- Consider the eigen-decomposition: $A = \sum_j \lambda_j |v_j\rangle\langle v_j|$
- $|b\rangle = \sum_j \beta_j |v_j\rangle$

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

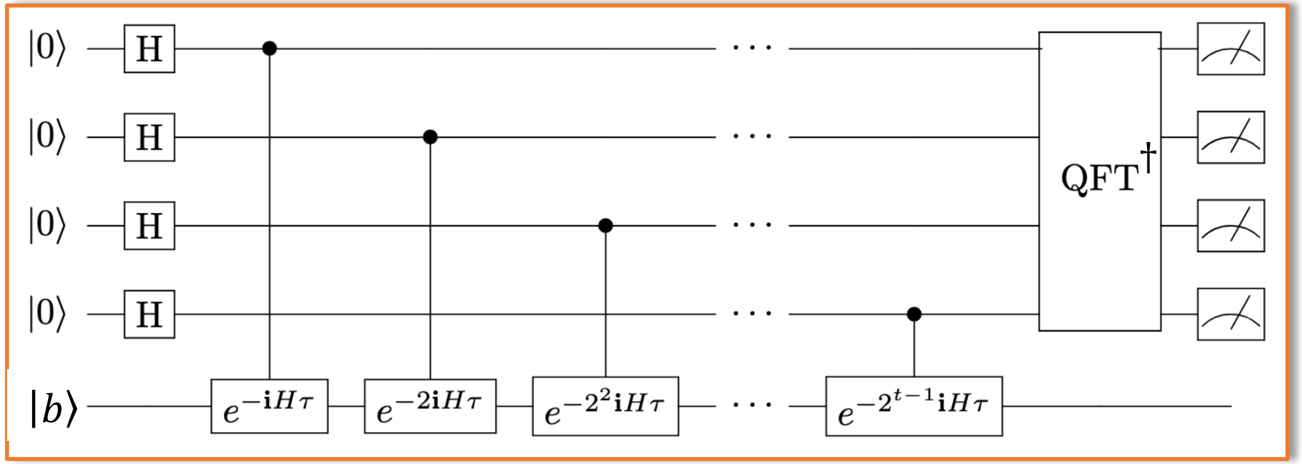
Need to do:

1. Store the information (binary encoding) of λ_j's in an ancilla register coherently $\longleftarrow$ QPE
2. Multiply the factor λ_j^{-1} to each eigenvector $|v_j\rangle$

Harrow-Hassidim-Lloyd (HHL)

Need to do:

1. Store the information (binary encoding) of λ_j's in an ancilla register coherently



$$U_{\mathrm{QPE}}|0\rangle|b\rangle = \sum_j \beta_j|\widetilde{\lambda_j}\rangle|v_j\rangle$$

Harrow-Hassidim-Lloyd (HHL)

Need to do:

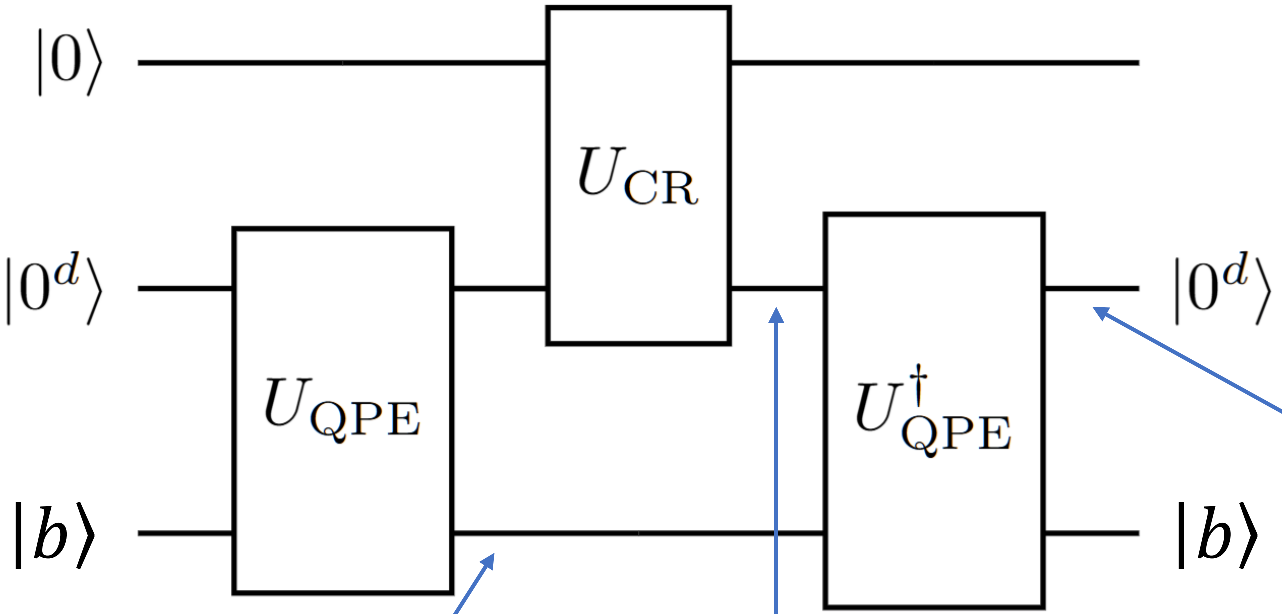
1. Store the information (binary encoding) of λ_j's in an ancilla register coherently
2. Multiply the factor λ_j^{-1} to each eigenvector $|v_j\rangle$

Controlled rotation:

$$U_{\mathrm{CR}}|0\rangle|\lambda_j\rangle = \left(\frac{C}{\lambda_j}|0\rangle + \sqrt{1-\frac{C^2}{\lambda_j^2}}|1\rangle\right)|\lambda_j\rangle$$

- See Sec. 4.3.2 of *QASC* if you want to know how to implement U_{CR}

Harrow-Hassidim-Lloyd (HHL)



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

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

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

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

Harrow-Hassidim-Lloyd (HHL)

Final state of HHL:

$$|0\rangle \sum_j C \frac{\beta_j}{\lambda_j} |v_j\rangle + |\perp\rangle = |0\rangle C A^{-1} |b\rangle + |\perp\rangle$$

- We need to choose C so that $\left|\frac{C}{\lambda_j}\right| \leq 1$, i.e., $C \leq |\lambda_0|$
- The success probability $= \| C A^{-1} |b\rangle \|^2 = C^2 \|x\|^2$, so we want C to be large

$C \sim 1/\kappa$

$$\Pr[\text{succ}] = \| C A^{-1} |b\rangle \|^2 = \kappa^{-2} \cdot \| A^{-1} |b\rangle \|^2 \geq \kappa^{-2} \cdot \frac{\| |b\rangle \|^2}{\|A\|^2} = \kappa^{-2}$$

- Number of repeats: $\mathcal{O}(\kappa^{-1})$
- What is the cost of QPE?

Harrow-Hassidim-Lloyd (HHL)

Due to the approximation errors from QPE, the actual final state is

$$|0\rangle \sum_j C \frac{\beta_j}{\widetilde{\lambda_j}} |v_j\rangle + |\perp\rangle \approx |0\rangle C A^{-1} |b\rangle + |\perp\rangle$$

where $\widetilde{\lambda_j} = \lambda_j (1 + e_j)$ with $|e_j| \leq \epsilon/4$

- For the unnormalized solution,

$$\|\tilde{x} - x\| = \left\| \sum_j \beta_j \left(\frac{1}{\widetilde{\lambda_j}} - \frac{1}{\lambda_j} \right) |v_j\rangle \right\| \leq \left\| \sum_j \frac{\beta_j}{\lambda_j} \left(\frac{-e_j}{1 + e_j} \right) |v_j\rangle \right\| = \mathcal{O}(\epsilon) \|x\|$$

- For the normalized solution state,

$$\| |\tilde{x}\rangle - |x\rangle \| = \left\| \frac{\tilde{x}}{\|\tilde{x}\|} - \frac{x}{\|x\|} \right\| \leq \left| 1 - \frac{\|\tilde{x}\|}{\|x\|} \right| + \frac{\|\tilde{x} - x\|}{\|x\|} \leq \epsilon$$

Harrow-Hassidim-Lloyd (HHL)

Due to the approximation errors from QPE, the actual final state is

$$|0\rangle \sum_j C \frac{\beta_j}{\widetilde{\lambda}_j} |v_j\rangle + |\perp\rangle \approx |0\rangle C A^{-1} |b\rangle + |\perp\rangle$$

where $\widetilde{\lambda}_j = \lambda_j (1 + e_j)$ with $|e_j| \leq \epsilon/4$

- Thus, QPE requires an addition error to within $\epsilon_{\mathrm{QPE}} = \mathcal{O}(\lambda_0 \epsilon) \leq \epsilon/\kappa$
- The cost of U_{QPE} is $\mathcal{O}(\kappa/\epsilon)$
- Overall complexity of HHL: $\mathcal{O}(\kappa^2/\epsilon)$
- Caveat: QPE requires Hamiltonian simulation $e^{-\mathbf{i}\tau A}$ and the $1/\epsilon$ in the complexity

QLSP via LCU (Fourier approach)

- Key identity:

$$\frac{1}{x} = \frac{\mathbf{i}}{\sqrt{2\pi}} \int_0^\infty \mathrm{d}y \underbrace{\int_{-\infty}^\infty z e^{-z^2/2} e^{-\mathbf{i}xyz} \mathrm{d}z}_{-\sqrt{2\pi}\mathbf{i}te^{-t^2/2}\big|_{t=xy}}$$

$$\forall \text{ odd } f(y) \text{ s.t. } \int_0^\infty f(y)\mathrm{d}y = 1, \qquad \int_0^\infty f(xy)\mathrm{d}y = \frac{1}{x}$$

QLSP via LCU (Fourier approach)

- Key identity:

$$\frac{1}{x} = \frac{\mathbf{i}}{\sqrt{2\pi}} \int_0^\infty \mathrm{d}y \int_{-\infty}^{\infty} z e^{-z^2/2} e^{-\mathbf{i}xyz} \mathrm{d}z$$

- It also holds for matrix:

$$\begin{aligned} A^{-1} &= \frac{\mathbf{i}}{\sqrt{2\pi}} \int_0^\infty \mathrm{d}y \int_{-\infty}^{\infty} z e^{-z^2/2} e^{-\mathbf{i}yzA} \mathrm{d}z \\ &\approx \frac{\mathbf{i}}{\sqrt{2\pi}} \int_0^Y \mathrm{d}y \int_{-Z}^{Z} z e^{-z^2/2} e^{-\mathbf{i}yzA} \mathrm{d}z \\ &\approx \frac{\mathbf{i}}{\sqrt{2\pi}} \sum_{j=0}^{J-1} \Delta_y \sum_{k=-K}^{K} \Delta_z z_k e^{-z_k^2/2} e^{-\mathbf{i}y_j z_k A} \end{aligned}$$

$$Y = \mathcal{O}\left(\kappa\sqrt{\log(\kappa/\epsilon)}\right),$$
$$Z = \mathcal{O}\left(\sqrt{\log(\kappa/\epsilon)}\right)$$

$$\text{LCU cost} \approx \|c\|_1 \approx \int_0^Y \mathrm{d}y \int_0^Z z e^{-z^2/2} \mathrm{d}z$$
$$= \mathcal{O}\left(\kappa\sqrt{\log(\kappa/\epsilon)}\right)$$

QLSP via LCU (Fourier approach)

- It also holds for matrix:

$$A^{-1} \approx \frac{\mathbf{i}}{\sqrt{2\pi}} \sum_{j=0}^{J-1} \Delta_y \sum_{k=-K}^{K} \Delta_z z_k e^{-z_k^2/2} e^{-\mathbf{i} y_j z_k A}$$

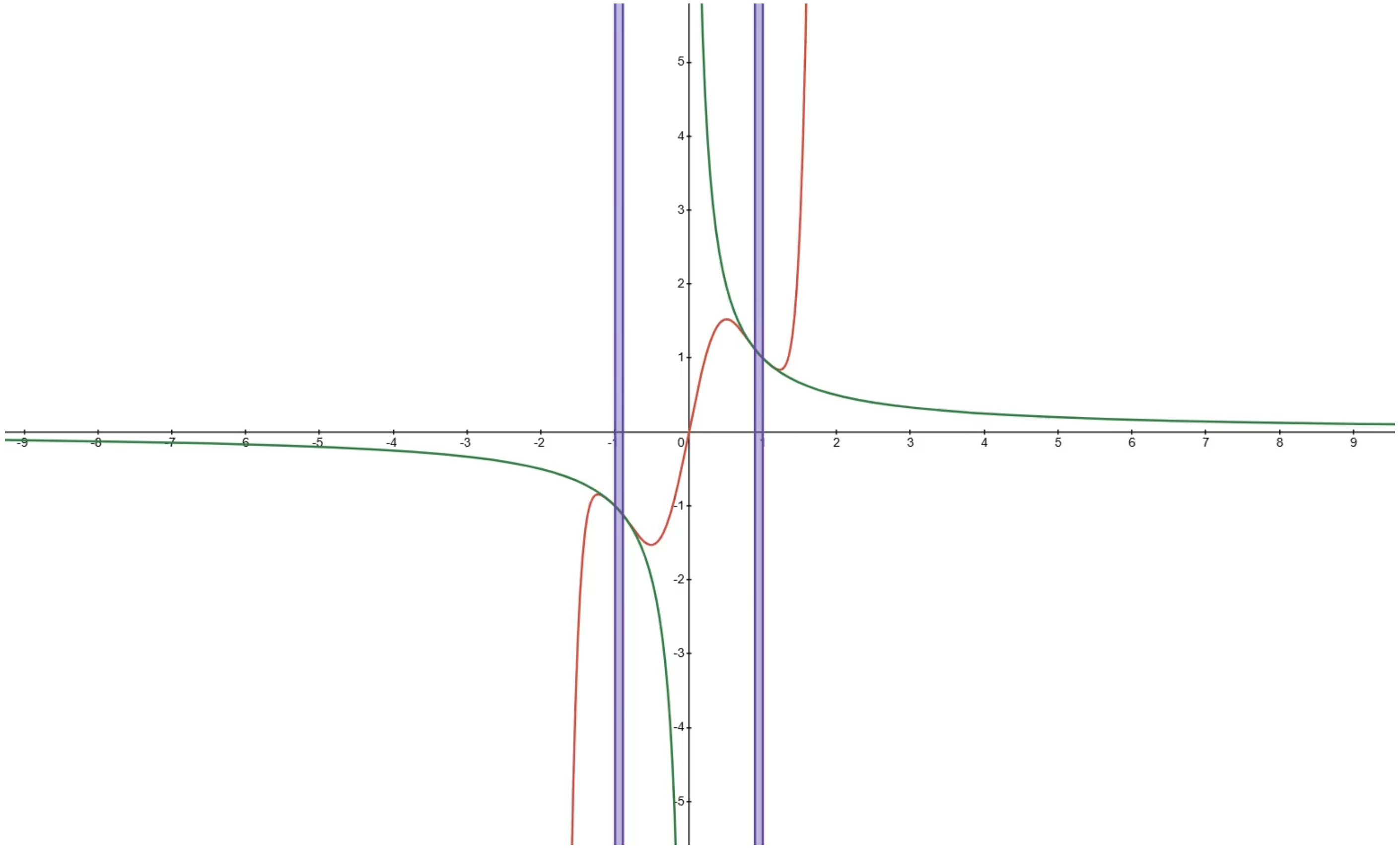
$$Y = \mathcal{O}\left(\kappa\sqrt{\log(\kappa/\epsilon)}\right),$$
$$Z = \mathcal{O}\left(\sqrt{\log(\kappa/\epsilon)}\right)$$

- LCU cost: $\mathcal{O}\left(\kappa\sqrt{\log(\kappa/\epsilon)}\right)$
- Hamiltonian simulation $e^{-\mathbf{i}AT}$ cost: $T\text{polylog}(1/\epsilon) = \kappa\text{polylog}(\kappa/\epsilon)$
- Overall complexity:

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

QLSP via LCU (Chebyshev approach)

$$\frac{1}{x} \approx \frac{1-(1-x^2)^d}{x} \qquad \forall x \in [-1,-\kappa] \cup [\kappa, 1] \qquad d \sim \kappa^2 \log(\kappa/\epsilon)$$

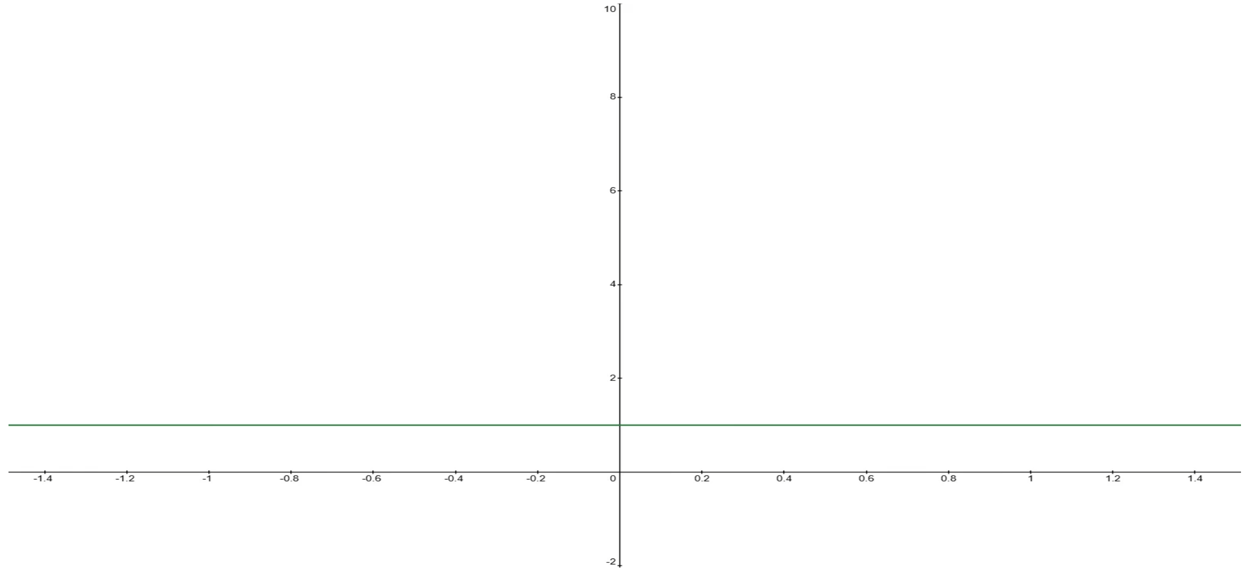


QLSP via LCU (Chebyshev approach)

$$\frac{1}{x} \approx \frac{1-(1-x^2)^d}{x} \qquad \forall x \in [-1,-\kappa] \cup [\kappa, 1] \qquad d \sim \kappa^2 \log(\kappa/\epsilon)$$

Chebyshev polynomials

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



QLSP via LCU (Chebyshev approach)

$$\frac{1}{x} \approx \frac{1-(1-x^2)^d}{x} \qquad \forall x \in [-1,-\kappa] \cup [\kappa, 1] \qquad d \sim \kappa^2 \log(\kappa/\epsilon)$$

Chebyshev polynomials

$$T_n(\cos\theta) = \cos(n\theta)$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \qquad T_0(x) = 1, T_1(x) = x$$

$$\frac{1-(1-x^2)^d}{x} = 4\sum_{j=0}^{d-1}(-1)^j\left(2^{-2d}\sum_{i=j+1}^{d}\binom{2d}{d+i}\right)T_{2j+1}(x)$$

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

QLSP via LCU (Chebyshev approach)

$$A^{-1} \approx 4 \sum_{j=0}^{\kappa \mathrm{polylog}(\kappa/\epsilon)} (-1)^j \left(2^{-2d} \sum_{i=j+1}^{d} \binom{2d}{d+i} \right) T_{2j+1}(A)$$

- The LCU cost $\approx \|c\|_1 = \mathcal{O}\left(\sqrt{d}\right) = \mathcal{O}\left(\kappa\sqrt{\log(\kappa/\epsilon)}\right)$
- $\left\|T_{2j+1}(A)\right\| \leq 1$ but it is non-unitary, so we need to construct a block-encoding for it
  - Suppose it has a cost $\mathcal{O}(j) \leq \kappa \mathrm{polylog}(\kappa/\epsilon)$
- Overall complexity:

$$\kappa\sqrt{\log(\kappa/\epsilon)} \times \kappa \mathrm{polylog}(\kappa/\epsilon) = \kappa^2 \mathrm{polylog}(\kappa/\epsilon)$$

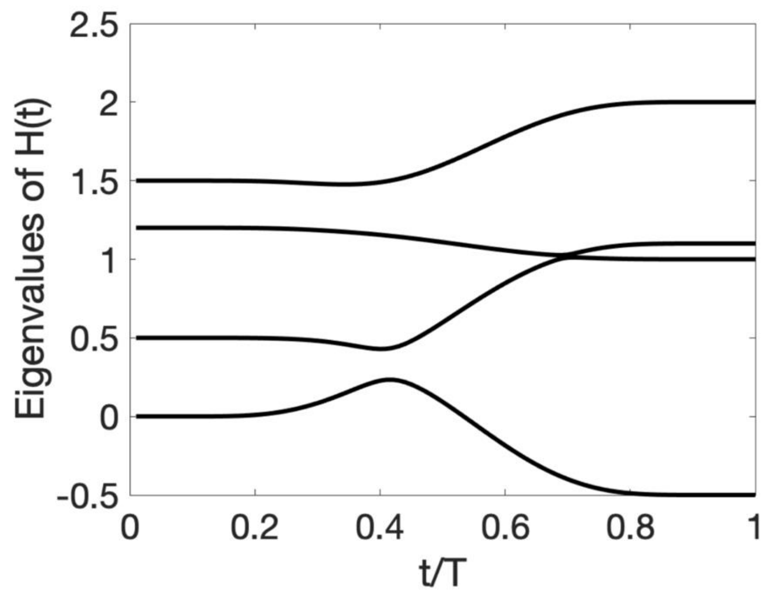
QLSP via Adiabatic quantum computation (AQC)

Schrödinger equation:

$$i\partial_t|\psi(t)\rangle = H(t/T)|\psi(t)\rangle, \qquad t \in [0,T]$$
$$H|\psi(0)\rangle = \lambda|\psi_0\rangle$$

Adiabatic evolution:

- Start from an eigenstate of the initial Hamiltonian $H(0)$
- $H(t/T)$ is changing slowly (i.e. T is large enough)
- Gap condition is satisfied
- The final state $|\psi(T)\rangle$ will approximate the eigenstate of $H(1)$



QLSP via Adiabatic quantum computation (AQC)

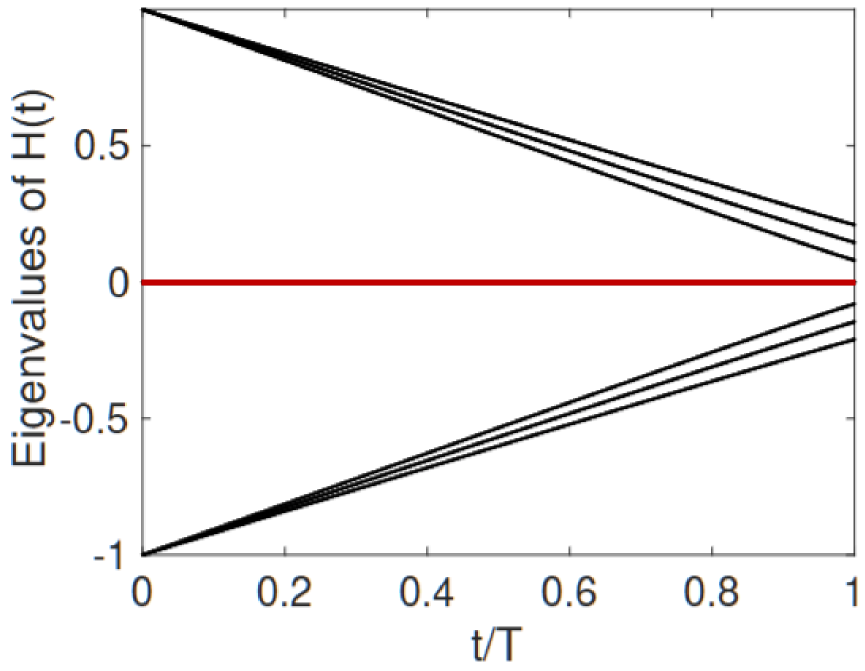
(Vanilla) AQC for QLSP:

$$H_0 = \begin{bmatrix} 0 & Q_b \\ Q_b & 0 \end{bmatrix}, \qquad H_1 = \begin{bmatrix} 0 & AQ_b \\ Q_b A & 0 \end{bmatrix}$$

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

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

- H_0 has two eigenstates with eigenvalue 0: $\begin{bmatrix} b \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ b \end{bmatrix}$
- H_1 has two eigenstates with eigenvalue 0: $\begin{bmatrix} x \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ b \end{bmatrix}$
- If we start from $\begin{bmatrix} b \\ 0 \end{bmatrix}$, it will be evolved to $\begin{bmatrix} x \\ 0 \end{bmatrix}$
- QLSP $\Longrightarrow$ (Time-dependent) Hamiltonian simulation



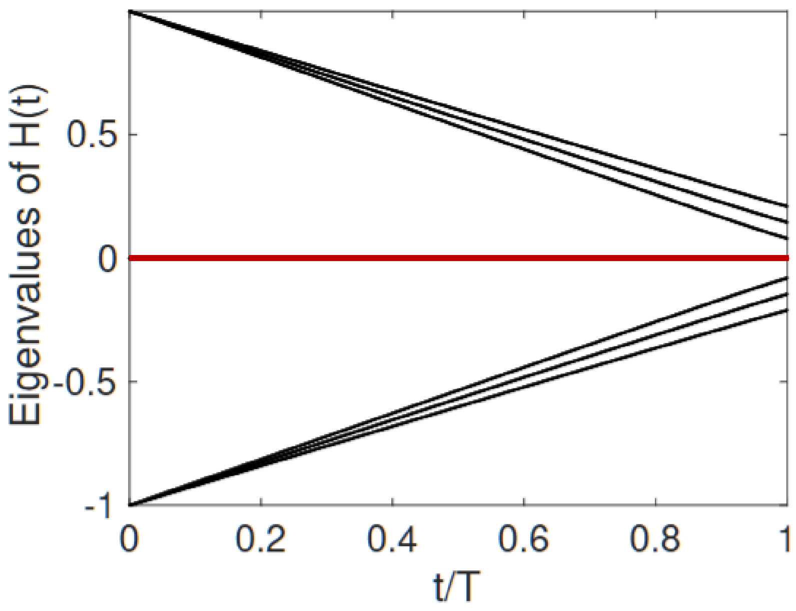
QLSP via Adiabatic quantum computation (AQC)

Quantum adiabatic theorem (Jansen-Ruskai-Seile '06).

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

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

- To bound the error by ϵ, $T \sim \Delta_*^{-3}\epsilon^{-1}$
- In QLSP, $\Delta_* \sim \kappa^{-1}$
- Thus, $T \sim \kappa^3\epsilon^{-1}$



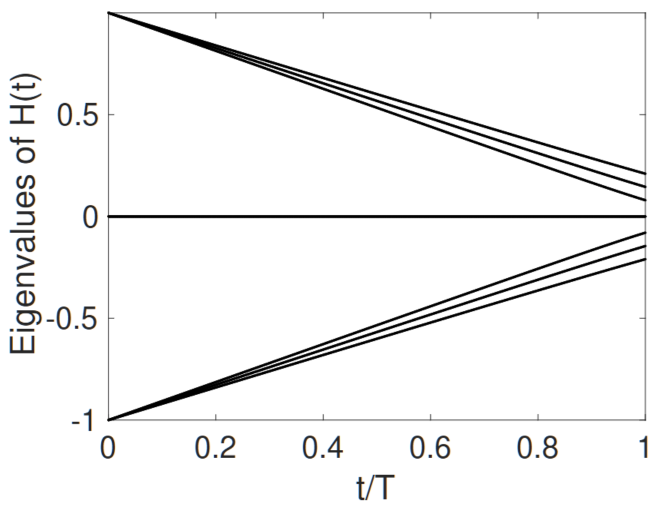
Time-optimal AQC for QLSP

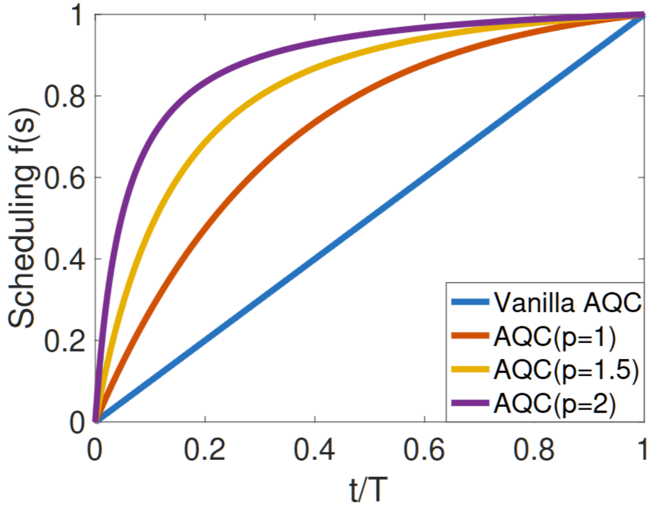
$$i\partial_t|\psi(t)\rangle = H(t/T)|\psi(t)\rangle, \qquad t \in [0, T]$$

- We can interpolate $H(s) = \big(1 - f(s)\big)H_0 + f(s)H_1$ with a clever design of $f(s)$ such that $\|H'(s)\|$ is small when the gap $\Delta(s)$ is small
- **An-Lin '19:**

AQC(p): $$\frac{\mathrm{d}f(s)}{ds} = c\Delta\big(f(s)\big)^p \quad \Longrightarrow \quad f(s) = \frac{\kappa}{\kappa - 1}\left(1 - \big(1 + s(\kappa^{p-1} - 1)\big)^{\frac{1}{1-p}}\right)$$

$$T = \mathcal{O}(\kappa/\epsilon)$$

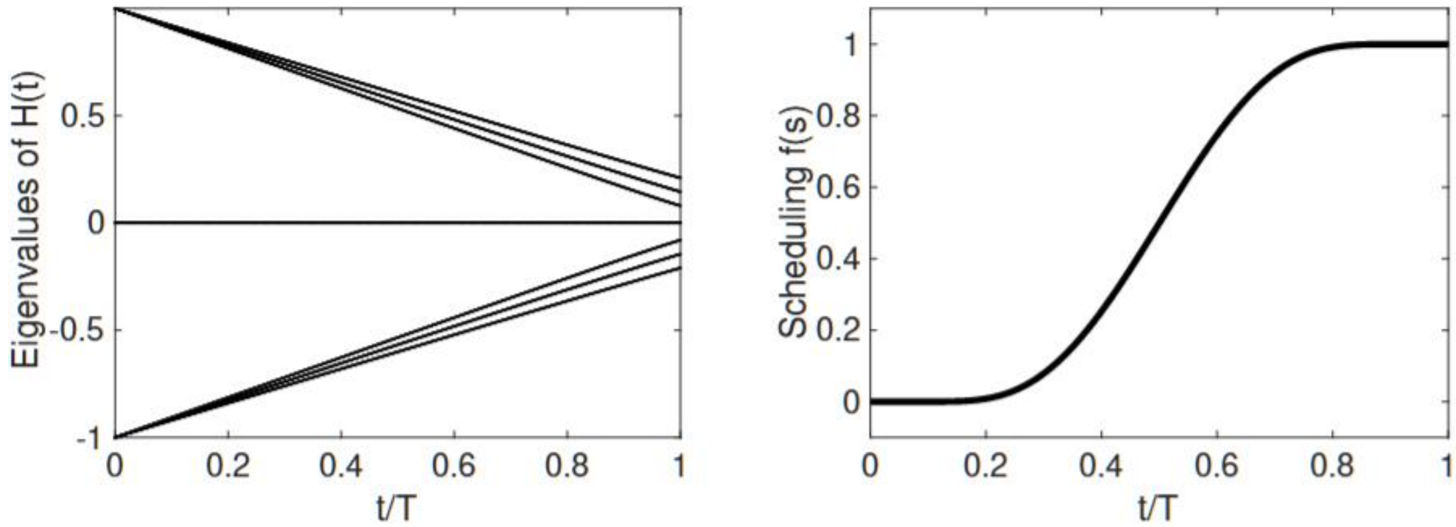




- Quantum Adiabatic Theorem gives a linear rate error decaying
- To improve the ϵ-dependence, we need higher-order convergence, i.e. error $\sim T^{-k}$
- This is true if $f(s)$ satisfies the boundary cancellation condition: all derivatives of $H\big(f(s)\big)$ vanishes at the boundary $s = 0$ and $s = 1$
- **An-Lin '19:**

AQC(exp): $$f(s) = c' \int_0^s \exp\left(-\frac{1}{x(1-x)}\right) \mathrm{d}x$$

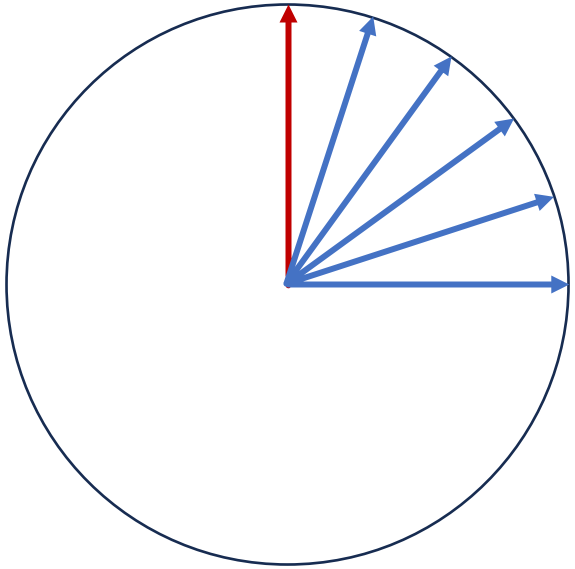
- $T = \kappa \mathrm{polylog}(\kappa/\epsilon)$
- Lower bound for QLSP:



Quantum Zeno effect (QZE)

Projective measurement:

- Let P be a projector "implemented" in quantum
- We can apply P to a quantum state $|\psi\rangle$:
  - w.p. $\|P|\psi\rangle\|^2$, $|\psi\rangle \mapsto \frac{P|\psi\rangle}{\|P|\psi\rangle\|}$
  - w.p. $1 - \|P|\psi\rangle\|^2$, $|\psi\rangle \mapsto \frac{P^\perp|\psi\rangle}{\|P^\perp|\psi\rangle\|}$

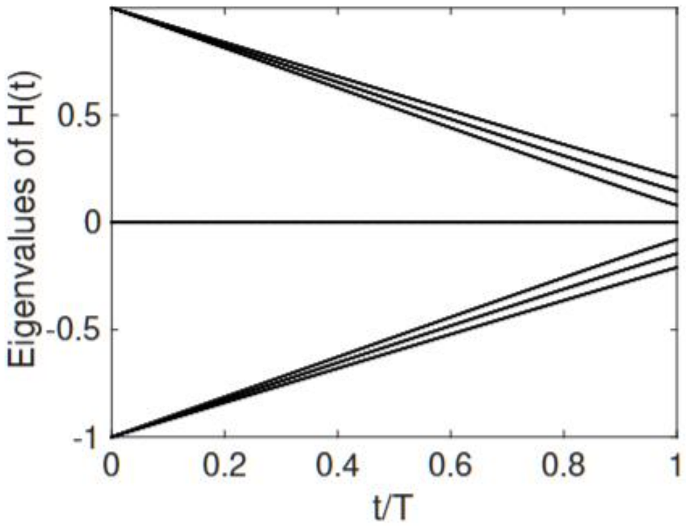


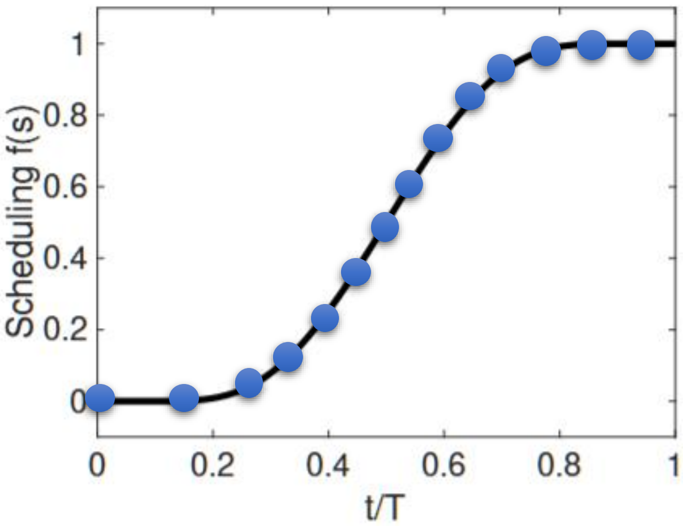
Quantum Zeno effect

- Suppose $|\psi_0\rangle, |\psi_1\rangle, \ldots, |\psi_M\rangle$ s.t. $|\langle\psi_{i-1}|\psi_i\rangle|^2 \geq 1 - \mathcal{O}(M^{-1})$
- Let $P_i \coloneqq |\psi_i\rangle\langle\psi_i|$ be the projector for each $i \in [M]$
- With $\mathcal{O}(1)$ success probability, we can convert $|\psi_0\rangle$ to $|\psi_M\rangle$ by applying projective measurements $P_1, P_2, \ldots, P_M$ to $|\psi_0\rangle$

QZE for adiabatic algorithms

Continuous AQC path:





- Let $|x(t)\rangle$ be the ground state of $H(t)$
- Choose $0 = t_0 < t_1 < t_2 < \cdots < t_M = 1$ such that $|\langle x(t-1)|x(t)\rangle|$ is sufficiently large
- Implement the projector $P_i \coloneqq |x(t_i)\rangle\langle x(t_i)|$ (i.e. the ground state projector) in block-encoding
- By QZE, we can convert $|x(0)\rangle$ to $|x(t_M)\rangle = |x\rangle$ without time-dependent Hamiltonian simulation

Quantum linear algebra toolbox

- Basic linear algebra operations
  - Input models for vectors and matrices
  - Matrix-vector multiplication
  - Matrix/vector addition: linear combination of unitaries (LCU)
  - Matrix multiplication
- Linear systems of equations
  - HHL: κ^2/ϵ
  - LCU (Fourier and Chebyshev): $\kappa^2 \log(1/\epsilon)$ (Can be improved to $\kappa \log(1/\epsilon)$ via VTAA)
  - AQC: $\kappa \log(1/\epsilon)$ (optimal)
  - Recent survey: *arXiv:2411.02522v3*

Application of QLSP: quantum SVM

Classification:

- **Input:** training data $\{(\mathbf{x}_i, y_i)\}_{i\in[N]}$, where $y_i \in \{\pm 1\}$; kernel function $K(\cdot,\cdot)$
- **Goal:** learn a classifier

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} w_i K(\mathbf{x}_i, \mathbf{x})\right)$$

Support vector machine (SVM)

- $K(\mathbf{x}, \mathbf{y}) \coloneqq \phi(\mathbf{x})^\top \phi(\mathbf{y})$, where $\phi(\cdot)$ is a given feature map

$$\begin{aligned} &\min_{\mathbf{w},b,e} \frac{1}{2}\mathbf{w}^\top \mathbf{w} + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \\ &\text{s.t.} \quad y_i \mathbf{w}^\top \phi(\mathbf{x}_i) = 1 - e_i, \qquad \forall i \in [N] \end{aligned}$$

$$\Longrightarrow$$

$$(K + \gamma^{-1} I)\boldsymbol{w} = \mathbf{y}$$

$$K_{ij} \coloneqq \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

QSVM

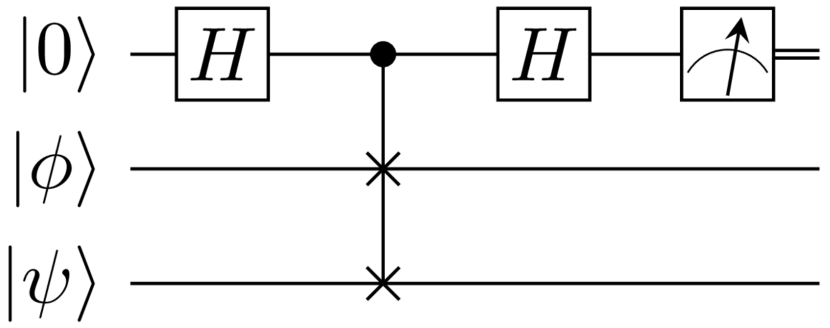
- Construct the block-encoding of the kernel matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$
- Construct the block-encoding of the matrix $H = K + \gamma^{-1} I$
- Apply the QLS solver to prepare the state $|\boldsymbol{w}\rangle = H^{-1}|\mathbf{y}\rangle$
- For each test data $\mathbf{x}$:
  - Compute $f(\mathbf{x}) = \sum_{i=1}^{N} w_i K(\mathbf{x}_i, \mathbf{x})$
  - Output $\mathrm{sgn}\big(f(\mathbf{x})\big)$

Complexity: $\tilde{O}(\kappa\alpha\ \mathrm{polylog}(NM)\log(\epsilon^{-1}))$

- α: block-encoding normalization factor
- M: dim of feature space; N: number of data

$$f(\mathbf{x}) = \langle \boldsymbol{w} | K(\cdot, \mathbf{x})\rangle$$

We can prepare $|K(\cdot, \mathbf{x})\rangle$ and use the swap test to estimate the inner product



Input model: matrices

Matrix as quantum gate (block-encoding):

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

$$A \approx \alpha(\langle 0^a| \otimes I)U_A(|0^a\rangle \otimes I)$$

Or equivalently,

$$U_A \approx \begin{bmatrix} A/\alpha & * \\ * & * \end{bmatrix}$$

- α is called the block-encoding factor and should satisfy $\alpha \geq \|A\|$
- U_A is called an (α, a, ϵ)-block-encoding of A
- Constructing U_A is generally hard, but easy in special cases such as unitaries, sparse matrices, special structured matrices (Gilyen et al. '18; Camps et al. '22)

Construct block-encodings

Block-encoding always exists:

- Let A be a general matrix with $\|A\| \leq 1$
- Consider the SVD $A = W\Sigma V^\dagger$, where Σ's diagonal entries are in $[0,1]$
- We have a $(1,1,0)$-block-encoding for A:

$$U_A \coloneqq \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma & \sqrt{I - \Sigma^2} \\ \sqrt{I - \Sigma^2} & -\Sigma \end{bmatrix} \begin{bmatrix} V^\dagger & 0 \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} A & W\sqrt{I - \Sigma^2} \\ \sqrt{I - \Sigma^2} V^\dagger & -\Sigma \end{bmatrix}$$

Construct block-encodings

Diagonal matrix:

- Let A be a diagonal matrix given by the quantum oracle:

$$O_A : |0\rangle|i\rangle \mapsto \left(A_{ii}|0\rangle + \sqrt{1 - |A_{ii}^2|}|1\rangle\right)|i\rangle$$

- $U_A \coloneqq O_A$ is a $(1,1,0)$-block-encoding for A:

$$\langle 0|\langle j|U_A|0\rangle|i\rangle = A_{ii}\delta_{ij} \quad \forall i, j \in [N]$$

Construct block-encodings

Sparse matrix:

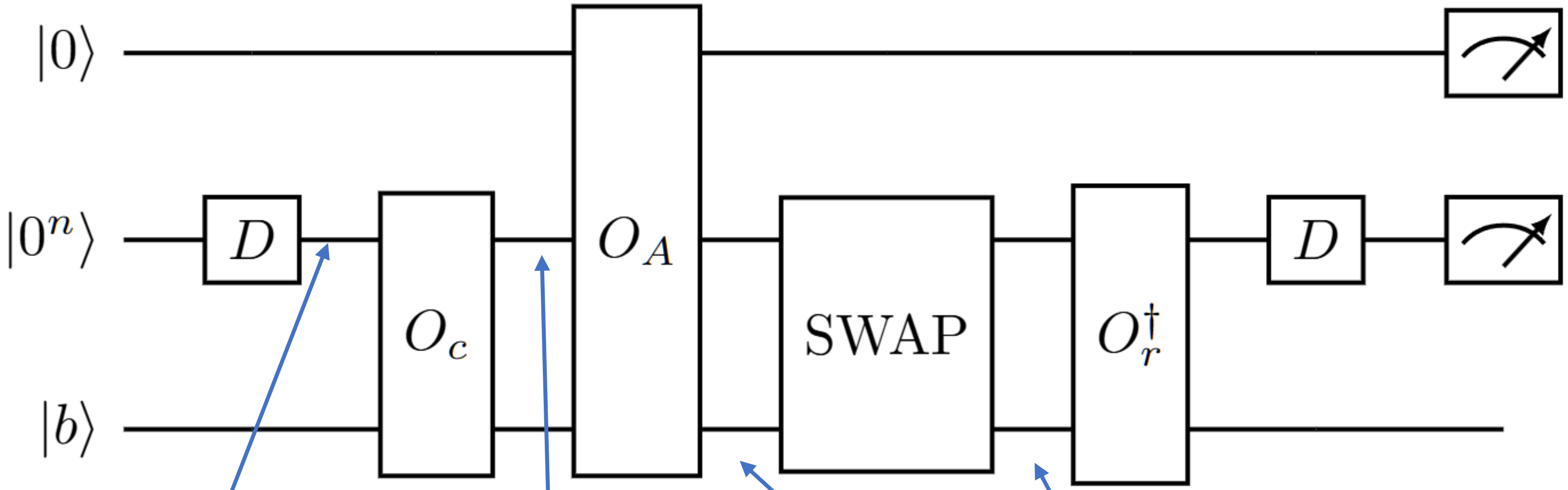
- Let A be a sparse matrix such that each row/column has $\leq S = 2^s$ nonzero entries
- Three input oracles:

$$O_r|l\rangle|i\rangle = |r(i,l)\rangle|i\rangle$$
$$O_c|r\rangle|j\rangle = |c(j,r)\rangle|j\rangle$$
$$O_A|0\rangle|i\rangle|j\rangle = \left(A_{ii}|0\rangle + \sqrt{1-|A_{ii}^2|}|1\rangle\right)|i\rangle|j\rangle$$

- $r(i,l)$ is the position of the l-th nonzero entry in the i-th row
- $c(j,r)$ is the position of the r-th nonzero entry in the j-th column

Construct block-encodings

Sparse matrix:



$$|0\rangle \frac{1}{\sqrt{S}} \sum_{r=0}^{S-1} |r\rangle\, |j\rangle$$

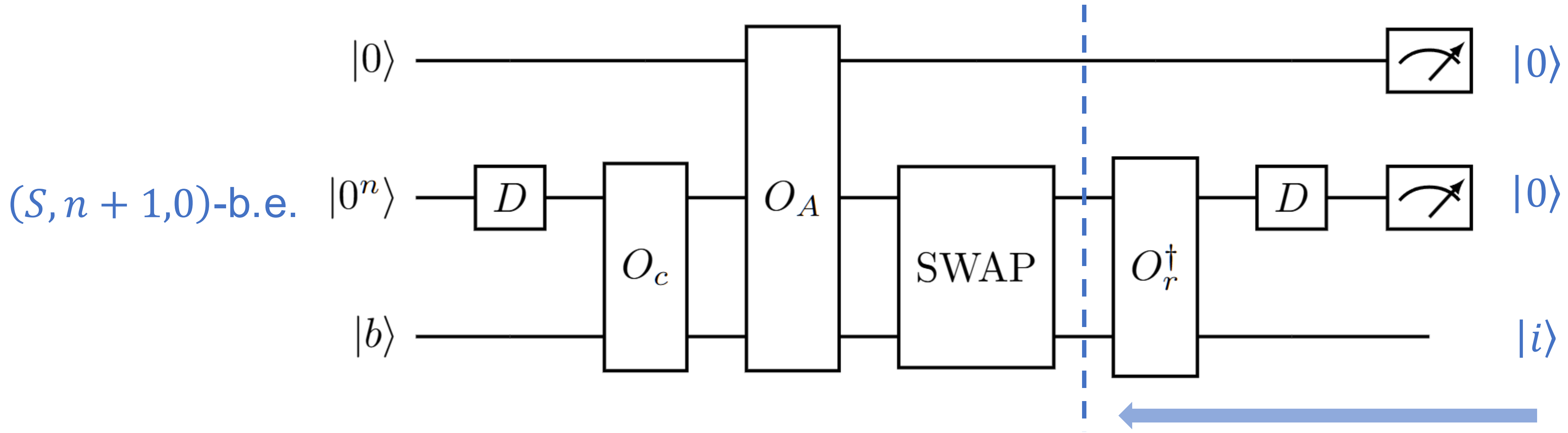
$$|0\rangle \frac{1}{\sqrt{S}} \sum_{r=0}^{S-1} |c(j,r)\rangle\, |j\rangle$$

$$\frac{1}{\sqrt{S}} \sum_{r=0}^{S-1} A_{c(j,r),j} |0\rangle |c(j,r)\rangle\, |j\rangle + |1\rangle |*\rangle$$

$$\frac{1}{\sqrt{S}} \sum_{r=0}^{S-1} A_{c(j,r),j} |0\rangle\, |j\rangle |c(j,r)\rangle + |1\rangle |*\rangle$$

Construct block-encodings

Sparse matrix:



$$\frac{1}{\sqrt{S}} \sum_{r=0}^{S-1} A_{c(j,r),j} |0\rangle\, |j\rangle |c(j,r)\rangle + |1\rangle |*\rangle \qquad \frac{1}{\sqrt{S}} \sum_{l=0}^{S-1} |0\rangle\, |r(i,l)\rangle |i\rangle$$

$$\langle 0|\langle 0|\langle i| U_A |0\rangle |0\rangle |j\rangle = \frac{1}{S} \sum_{l,r=0}^{S-1} A_{c(j,r),j}\, \delta_{i,c(j,r)} \delta_{j,r(i,l)} = \frac{1}{S} A_{ij}$$