Quantum Advantages for Functions of Matrices

Changpeng Shao

Academy of Mathematics and Systems Science, Chinese Academy of Sciences

East China Normal University
2026.5.25

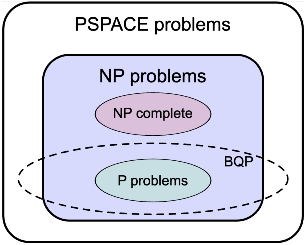
Background

Background

- Quantum computers can solve some problems much faster than classical computers, e.g.,
  - Shor's algorithm for integer factorization (exponential speedup)
  - Grover's algorithm for unstructured search (quadratic speedup)

- When exploring quantum advantages, we are usually interested in BQP complete problems. Those problems demonstrate the full power of quantum computers.
  - Hamiltonian simulation e^{iAt}
  - Matrix inversion A^{-1}
  - Matrix powers A^d
  - Matrix exponential e^{At} [SIAM Review 2003]

- Those problems can be summarized as computing $f(A)$, and more precisely computing $f(A)\mathbf{b}$ for a quantum computer.



Background

Assume A is sparse and Hermitian for convenience. Then the complexity of computing those matrix functions on a quantum computer is:

Hamiltonian simulation e^{iAt}	$\Theta(t)$
Matrix inversion A^{-1}	$\Theta(\kappa)$
Matrix powers A^d	$\Theta(\sqrt{d})$
Matrix exponential e^{At}	$\Theta(\sqrt{t})$

Here, κ is the condition number of A.

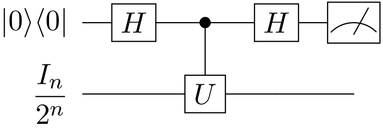
Key point: the complexity is polylog in dimension.

To understand better quantum advantages, we need to know classical hardness, usually complexity lower bounds.

The DQC1 (One clean qubit model) Model

The above is about the standard quantum circuit model, the BQP model. We can also consider a weak quantum model, the DQC1 model.

- A simple quantum computing model introduced by Knill and Laflamme in 1998, motivated by the study of nuclear magnetic resonance quantum computers.



- The probability of measuring 0 is

$$\frac{1}{2} + \frac{1}{2}\frac{\mathrm{Re}(\mathrm{Tr}[U])}{2^n}$$

The real part can be estimated up to ε by repeating the circuit $O(1/\varepsilon^2)$ times.

The DQC1 (One clean qubit model) Model

- In DQC1 model, we can estimate $\frac{\mathrm{Tr}[U]}{2^n}$ efficiently.
- It is believed to be weaker than BQP model, but it can still solve some problems that are hard for classical computers.

Knill & Laflamme (1998)	Estimating $\frac{1}{2^n}\mathrm{Tr}(U)$ is DQC1-complete
Shepherd (2006)	DQC1 $=$ DQCk, $k = O(\log n)$
Shor & Jordan (2008)	Estimating Jones polynomial is DQC1-complete
Brandão (2008)	Estimating $\frac{1}{2^n}\mathrm{Tr}(e^{-\beta A})$ is DQC1-complete
Morimae, Fujii & Fitzsimons (2014)	Classically hard to simulate DQC1
Edenhofer, Hasegawa & Le Gall (2025)	Estimating $\frac{1}{2^n}\mathrm{Tr}(A^k)$, $\frac{1}{2^n}\mathrm{Tr}(A^{-1})$ is DQC1-complete
Shao & Wang (2025)	Estimating $\frac{1}{2^n}\mathrm{Tr}(\log A)$ is DQC1-complete

- The above results assume A is log-local.

Normalized trace (spectral sum) estimation

Many DQC1-complete problems can be written as the following problem.

Problem 1 (Normalized trace (spectral sum) estimation)

Let $f : [-1, 1] \to [-1, 1]$ be continuous, A be a Hermitian matrix of dimension N, and the goal is to estimate the normalized trace

$$\frac{1}{N} \mathit{Tr}[f(A)].$$

Compared to BQP, DQC1 is far less well studied.

Classical algorithm for spectral sum (Avron, Toledo 2011):

$$\frac{1}{N}\mathrm{Tr}[f(A)] = \mathbb{E}[f(A)_{ii}] \approx \frac{1}{k}\sum_{i=1}^{k}\langle v_i|f(A)|v_i\rangle$$

for some random vectors $v_1, \ldots, v_k$.

Brief summary

In summary, for the task of functions of matrices

- BQP model is good at computing $|f(A)\mathbf{b}\rangle$, or $\langle \mathbf{a}|f(A)|\mathbf{b}\rangle$.
- DQC1 model is good at computing $\frac{1}{N}\mathrm{Tr}[f(A)]$.

This talk: These problems are shown to be BQP/DQC1-complete, thereby capturing the full computational power of quantum computers.

Problem statements and main results

Problem statement in the BQP model

Problem 2 (Approximate an entry of $f(A)$)

Let $f(x) : [-1,1] \to [-1,1]$ be a continuous function, let A be sparse and Hermitian with $\|A\| \le 1$. Given two indices i, j and accuracy ε, compute $\langle i|f(A)|j\rangle \pm \varepsilon$.

For a sparse matrix $A = (A_{i,j})$, we are given 2 (commonly used) oracles:

$$(i,j) \longrightarrow \boxed{\mathcal{O}_1} \longrightarrow p_{i,j}$$

$$(i,j) \longrightarrow \boxed{\mathcal{O}_2} \longrightarrow A_{i,j}$$

where p_{ij} is the index of the j-th nonzero entry in the i-th row.

In the quantum case, we assume the oracles can be used in superposition, e.g.,
$\sum \alpha_{i,j}|i,j\rangle|0\rangle \mapsto \sum \alpha_{i,j}|i,j\rangle|A_{i,j}\rangle$

The query complexity is the minimal number of calls to the oracles to solve the problem.

An example: matrix powers A^m

Classical algorithms:

- Assume A is s-sparse, then by definition

$$(A^m)_{i,j} = \sum_{k_1}\sum_{k_2}\cdots\sum_{k_{m-1}} A_{i,k_1}A_{k_1,k_2}\cdots A_{k_{m-1},j}$$

  The complexity is $O(s^{m-1})$.

- For x^m, there is an approximation polynomial of degree $\Theta(\sqrt{m})$, so can be improved to $s^{\widetilde{O}(\sqrt{m})}$ [Sachdeva & Vishnoi, '14].
- About trace, $\frac{1}{N}\mathrm{Tr}[A^m] = \mathbb{E}[(A^m)_{ii}]$ and $O(1/\varepsilon^2)$ samples are enough.
- Our result (lower bound): $\widetilde{\Omega}((s/2)^{(\sqrt{m}-1)/6})$.

Quantum algorithms:

- Upper bound by QSVT: $O(s\sqrt{m}/\varepsilon)$.
- Our result (lower bound): $\Omega(\sqrt{m})$.

General case (our results)

Assume f is continuous, A is s-sparse and Hermitian, then computing $f(A)_{i,j} \pm \varepsilon$ costs

	Quantum algorithm	Classical algorithm
Upper bound	$O(sd/\varepsilon)$	$O(s^{d-1})$
Lower bound	$\Omega(d)$	$\widetilde{\Omega}((s/2)^{(d-1)/6})$

where $d = \widetilde{\deg}_\varepsilon(f)$ is the approximate degree:

$$\begin{aligned} \widetilde{\deg}_\varepsilon(f) \;&=\; \min\{d : |f(x) - g(x)| \leq \varepsilon, \forall x \in [-1,1], \\ &\qquad g(x) \text{ is a polynomial of degree } d\}. \end{aligned}$$

The quantum lower bound is similar to the polynomial method for Boolean functions [Beals, Buhrman, Cleve, Mosca, de Wolf, '98].

Approximate degree: some examples

Recall:

Hamiltonian simulation e^{iAt}	$\Theta(t)$
Matrix inversion A^{-1}	$\Theta(\kappa)$
Matrix powers A^d	$\Theta(\sqrt{d})$
Matrix exponential e^{At}	$\Theta(\sqrt{t})$

We indeed have

- $\widetilde{\deg}(e^{ixt}) = \Theta(t)$ over $x \in [-1,1]$
- $\widetilde{\deg}((\kappa x)^{-1}) = \Theta(\kappa)$ over $x \in [-1, 1/\kappa] \cup [1/\kappa, 1]$
- $\widetilde{\deg}(x^d) = \Theta(\sqrt{d})$ over $x \in [-1,1]$
- $\widetilde{\deg}(e^{(x-1)t}) = \Theta(\sqrt{t})$ over $x \in [-1,1]$

BQP-completeness

Problem 3 (Entry estimation problem: decision version)

Let $f(x) : [-1, 1] \to [-1, 1]$ *be a continuous function, let* A *be an* $N \times N$ *sparse Hermitian matrix such that* $\|A\| \leq 1$*. Let* $\varepsilon \in (0, 1)$ *be the precision and* i, j *be two indices. Assume that one of the following holds:*

- *YES case: if* $f(A)_{ij} \geq \varepsilon$*, or*
- *NO case: if* $f(A)_{ij} \leq -\varepsilon$*.*

Decide which is the case.

Theorem 1 (BQP-completeness)

Assume that $\widetilde{\deg}_\varepsilon(f) = \Omega(\mathrm{polylog}(N))$*. Then the "entry estimation problem" is PromiseBQP-complete.*

Previous, only some special cases are known: $f(x) = e^{ixt}, x^{-1}, x^d, e^{xt}$.

The DQC1 case

Under "certain conditions", we have the following results.

Theorem 2 (Query complexity lower bound for DQC1)

Assume that A is s-sparse and Hermitian, then computing $\frac{1}{N}\mathit{Tr}[f(A)] \pm \varepsilon$ with $\varepsilon = \Omega(1)$ costs

	Quantum	*Classical*
Upper bound	$s \cdot \widetilde{\deg}_\varepsilon(f)$	$s^{\widetilde{\deg}_\varepsilon(f)}$
Lower bound	$\widetilde{\deg}_\varepsilon(f)$	$(s/2)^{\widetilde{\deg}_\varepsilon(f)/9}$

Theorem 3 (DQC1-completeness)

Assume that A is log-local and $\widetilde{\deg}_\varepsilon(f) = \Omega(\operatorname{poly}\log N)$, then estimating $\frac{1}{N}\mathit{Tr}[f(A)]$ is DQC1-complete.

Some technical ideas in the proofs

Key theorem in the proofs

Theorem (Key theorem)

Let $f : [-1,1] \to [-1,1]$ be continuous with odd part $f_{\mathrm{odd}}(x) = \frac{f(x)-f(-x)}{2}$, then there is a symmetric tridiagonal matrix

$$A = \begin{pmatrix} 0 & b_1 & & & \\ b_1 & 0 & b_2 & & \\ & b_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & b_{n-1} & 0 \end{pmatrix}_{n\times n}$$

satisfying $b_i \neq 0$, $\|A\| \leq 1$ and $f(A)_{1,n} \geq \varepsilon$, where $n = \widetilde{\deg}_\varepsilon(f_{\mathrm{odd}}) + O(1)$.

Proof. linear semi-infinite programming + dual polynomial method + properties of tridiagonal matrices. ∎

Lower bound's proof of quantum algorithms

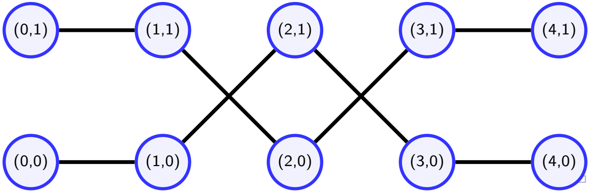
Main idea is inspired by the no fast-forwarding theorem [Berry, Ahokas, Cleve, Sanders, Comm. Math. Phys. '07].

Parity problem: Given $x_1, \ldots, x_n \in \{0,1\}$, compute $x_1 \oplus \cdots \oplus x_n$, the quantum query complexity is $\Theta(n)$.

We construct a weighted graph G:

- **vertices:** (i,b), where $i \in \{0,1,\ldots,n\}, b \in \{0,1\}$
- **edges:** an edge between $(i-1,b)$ and $(i, b \oplus x_i)$
- **weights:** chosen carefully

For example, $(x_1,x_2,x_3,x_4) = (0,1,1,0)$, then G is



Lower bound's proof of quantum algorithms

Essentially, G consists of two paths

$$(0,0) - (1,x_1) - (2,x_1 \oplus x_2) - \cdots - (n, x_1 \oplus \cdots \oplus x_n)$$
$$(0,1) - (1,1 \oplus x_1) - (2,1 \oplus x_1 \oplus x_2) - \cdots - (n, 1 \oplus x_1 \oplus \cdots \oplus x_n)$$

Let A be the adjacency matrix of G (essentially two symmetric tridiagonal matrices).

- **Case 1:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 0$, then $\langle 0,0|f(A)|n,1\rangle = 0$
- **Case 2:** if $x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1$, then we hope to find appropriate weights such that $\langle 0,0|f(A)|n,1\rangle \geq \varepsilon$ for some ε. The weights are determined by the key theorem.

As a result, if we can estimate $\langle 0,0|f(A)|n,1\rangle$, we can then solve the parity problem, which is known to be hard. It is very important to ensure that $n = \widetilde{\deg}(f) + O(1)$ in the key theorem.

Lower bound's proof of classical algorithms

Key tools: the Forrelation problem + Feynman's clock construction.

Forrelation problem (Aaronson & Ambainis, 2015):

Given $g_1, g_2 : \{0,1\}^n \to \{\pm 1\}$, let $D_i = \text{diag}(g_i(x) : x \in \{0,1\}^n)$, $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, define

$$\begin{aligned} \Phi(g_1, g_2) &:= \langle 0^n | H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n} | 0^n \rangle \\ &= \frac{1}{2^{3n/2}} \sum_{x,y \in \{0,1\}^n} (-1)^{x \cdot y} g_1(x) g_2(y). \end{aligned}$$

The goal is to compute $\Phi(g_1, g_2) \pm 1/3$

For this problem, the classical query complexity is lower bounded by $\Omega(\sqrt{2^n}/n)$, while the quantum query complexity is 1. The problem is also BQP-complete.

Feynman's clock construction

Let $U = U_m \cdots U_2 U_1$ be a unitary operator, define

$$A = \sum_{t=1}^{m} b_t \big(|t\rangle\langle t-1| \otimes U_t + |t-1\rangle\langle t| \otimes U_t^\dagger \big) = \begin{pmatrix} 0 & b_1 U_1^\dagger & & \\ b_1 U_1 & 0 & b_2 U_2^\dagger & \\ & b_2 U_2 & \ddots & \ddots \\ & & \ddots & \ddots \end{pmatrix}$$

Let $|\psi_t\rangle := |t\rangle \otimes U_t \cdots U_1 |0\rangle$, then

$$A|\psi_t\rangle = b_{t-1}|\psi_{t-1}\rangle + b_{t+1}|\psi_{t+1}\rangle$$

In subspace $\{|\psi_t\rangle : t = 0, 1, \ldots, m\}$, A is a symmetric tridiagonal matrix.

Rmk. Assume $b_i = 1/2$ for all i, if you can simulate e^{iAt} for $t = \Theta(m)$ on a classical computer, then you can simulate U. Namely, $e^{iAt}|0\rangle \otimes |0\rangle \approx |m\rangle \otimes U|0\rangle$.

Lower bound's proof of classical algorithms

In the Forrelation problem, we have $U = H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n}$. To ensure A is sparse in the clock construction, we decompose

$$H^{\otimes n} = (H \otimes I \otimes \cdots \otimes I)(I \otimes H \otimes \cdots \otimes I) \cdots (I \otimes I \otimes \cdots \otimes H)$$

Now $m = 3n + 3$,

$$\begin{aligned} |\psi_0\rangle &= |0\rangle \otimes |0\rangle \\ |\psi_m\rangle &= |m\rangle \otimes H^{\otimes n} D_1 H^{\otimes n} D_2 H^{\otimes n} |0\rangle \end{aligned}$$

Let $|\phi_m\rangle = |m\rangle \otimes |0\rangle$, then

$$\begin{array}{ccccc} \langle \phi_m | f(A) | \psi_0 \rangle & = & \langle \psi_m | f(A) | \psi_0 \rangle & \cdot & \Phi(g_1, g_2) \\ \downarrow & & \downarrow & & \downarrow \\ \text{an entry of } f(A) & & \text{Easy} & & \text{Hard} \\ \downarrow & & & & \\ \text{Hard} & & & & \end{array}$$

The above ideas generalize to DQC1 naturally

- An n-qubit unitary $U = U_m \cdots U_2 U_1$, $m = \mathrm{poly}(n)$.
- Feynman's circuit-to-Hamiltonian construction: $(|m\rangle := |0\rangle)$

$$A = \sum_{t=1}^{m} a_t |t\rangle\langle t| \otimes I_n + \sum_{t=1}^{m} b_t \big(|t\rangle\langle t-1| \otimes U_t + |t-1\rangle\langle t| \otimes U_t^\dagger \big),$$

- A key observation: for any eigenpair $(e^{i\theta}, |\phi_\theta\rangle)$ of U, the restriction of A to the invariant subspace spanned by

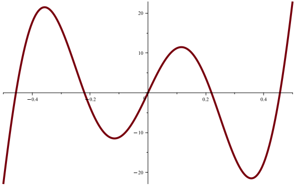
$$\Big\{ |0\rangle \otimes |\phi_\theta\rangle, \quad |1\rangle \otimes U_1|\phi_\theta\rangle, \quad \ldots, \quad |m-1\rangle \otimes U_{m-1} \cdots U_2 U_1 |\phi_\theta\rangle \Big\}$$

is a periodic Jacobi matrix of the form

$$A_\theta := \begin{pmatrix} a_1 & b_1 & & & b_m e^{i\theta} \\ b_1 & a_2 & b_2 & & \\ & b_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{m-1} \\ b_m e^{-i\theta} & & & b_{m-1} & a_m \end{pmatrix}.$$

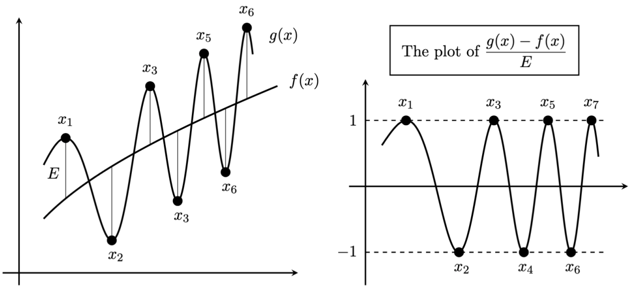
Technical overview

- Thus
$$\mathrm{Tr}[f(A)] = \sum_{\theta:\, e^{i\theta} \text{ is an eigenvalue of } U} \mathrm{Tr}[f(A_\theta)].$$
- Recall $\mathrm{Re}(\mathrm{Tr}[U]) = \sum_\theta \cos\theta$
- About periodic Jacobi matrix,
$$\det(xI - A_\theta) = 2b_1 \cdots b_m \big(\Delta(x) - \cos\theta\big),$$
where $\Delta(x)$ is independent of θ,
and $\mathrm{Tr}[\Delta(A_\theta)] = m\cos\theta$.



- A natural, but incorrect, idea: if $\Delta(x) \approx f(x)$, then $\mathrm{Tr}[f(A_\theta)] \approx m\cos\theta$.

Technical overview



- (crucial) Using Chebyshev equioscillation theorem to fix it.
- Roughly, $\Delta(x) \approx \dfrac{f(x)-g(x)}{E}$, from which we can reconstruct A.
- Then we obtain a connection between $\frac{1}{m2^n}\mathrm{Tr}[f(A)]$ and $\frac{1}{2^n}\mathrm{Re}(\mathrm{Tr}[U])$.
- $\Rightarrow$DQC1-completeness.

k-forrelation problem in the DQC1 model

- Let $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, and $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \{1, -1\}^{2^n}$ with oracles $O_{\mathbf{x}_1}, \ldots, O_{\mathbf{x}_k}$ to query them.
- Let $U = O_{\mathbf{x}_1} H^{\otimes n} O_{\mathbf{x}_2} H^{\otimes n} \cdots H^{\otimes n} O_{\mathbf{x}_k} H^{\otimes n}$ and

$$\texttt{Trace}_k(\mathbf{x}) := \frac{1}{2^n}\mathrm{Tr}[U].$$

- **Open problem:** The classical query complexity of estimating $\texttt{Trace}_k(\mathbf{x})$ is lower bounded by

$$\widetilde{\Omega}(2^{n(1-2/k)}).$$

- Using the connection established above, one can prove a lower bound of

$$\widetilde{\Omega}((s/2)^{\widetilde{\deg}_\varepsilon(f)/9}).$$

Final remarks

- Our results hold when $E_d/E_{d-1} < 1/2$, where $d = \widetilde{\deg}_\varepsilon(f)$ and
$$E_d = \min_{\deg(g)\le d} \max_{-1\le x\le 1} |f(x) - g(x)|.$$
- Many functions satisfy this condition.
- But we believe this condition can be removed.
- In addition, our lower bounds are not tight.
- References: arXiv:2311.06999, arXiv:2604.01519.

Thank you!