Quantum lower bounds for convex optimization and real matrix-vector query problems

arXiv:2609.05679

Andrew Childs
University of Maryland

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

UMIACS
University of Maryland Institute for Advanced Computer Studies

JOINT CENTER FOR
QUANTUM INFORMATION
AND COMPUTER SCIENCE

The power of quantum computers

Using carefully designed interference between different computational paths, quantum computers can solve some problems dramatically faster than classical computers.

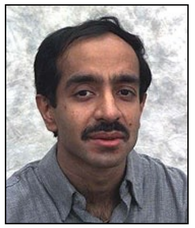
Some problems admit exponential quantum speedup.

Period finding, factoring, discrete log, quantum simulation, quantum linear algebra, Jones polynomial approximation, counting points on curves, graph connectivity with cut queries, …



Other problems admit polynomial quantum speedup.

Unstructured search, formula evaluation, collision finding, network flows, finding subgraphs, minor-closed graph properties, group commutativity, convex optimization, string problems, …



What problems can be solved significantly faster by quantum computers?

What are the limitations of quantum algorithms?

Quantum computation

States

$$|\psi\rangle = \sum_{x\in\{0,1\}^m} \psi_x|x\rangle \in \mathbb{C}^{2^m}$$

(m qubits)

amplitudes $\psi_x \in \mathbb{C}$

normalization: $\sum_x |\psi_x|^2 = 1$

Transformations

$$|\psi\rangle \mapsto U|\psi\rangle$$

unitary: $U^\dagger U = I$

built from 1 & 2-qubit gates

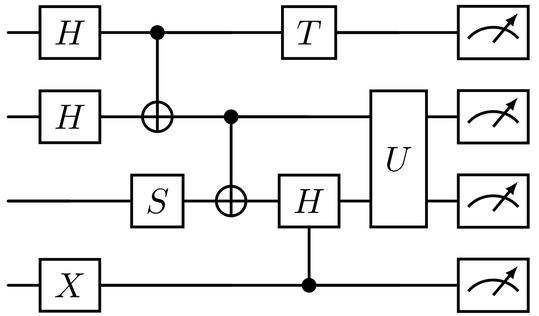
Measurements

outcome x occurs with probability $|\psi_x|^2$

Amplitudes are sums over computational paths:

$$\langle x_T|\, U_T \cdots U_2 U_1\, |x_0\rangle = \sum_{x_1,\ldots,x_{T-1}} \prod_{j=1}^{T} \langle x_j|U_j|x_{j-1}\rangle$$

Paths can interfere constructively or destructively.
Probabilities are quadratic in the amplitudes.



cost = number of gates

Query complexity

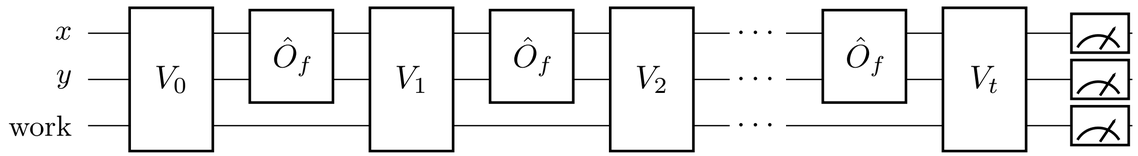
Query complexity: the input is provided by a black box. How many queries are needed to compute some property of the input?

- Models an expensive subroutine whose uses we want to minimize.
- Enables provable lower bounds.

Quantum queries act on superpositions of inputs:

$$O_f|x, y\rangle = |x,\, y + f(x)\rangle \quad \text{or}$$
$$\hat{O}_f|x, y\rangle = e^{2\pi i\, y\, f(x)}|x, y\rangle$$

The phase version is O_f conjugated by a Fourier transform on the second register.



Example: the OR of N bits. Classically, $\Theta(N)$ queries; quantumly, $\Theta(\sqrt{N})$ [Grover 96; BBBV 97]

Polynomial method [Beals, Buhrman, Cleve, Mosca, de Wolf 98]: after t queries, the acceptance probability is a polynomial of degree $\leq 2t$ in the input bits. Computing OR needs degree $\Omega(\sqrt{N})$, so $Q(\mathrm{OR}) = \Omega(\sqrt{N})$.

Query complexity of convex optimization

Minimize a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ over a convex body $K \subset \mathbb{R}^n$.

Given: an evaluation oracle $x \mapsto f(x)$, a membership oracle $x \mapsto \mathbf{1}_{x \in K}$, and a point in K.

We study the dependence on the dimension n.

Queries	**Classical**	**Quantum**
evaluation + membership	$\tilde{O}(n^2)$ [LSV 18] $\tilde{\Omega}(n^2)$ [ZZQL 26]	$\tilde{O}(n)$ [AGGW 18; CCLW 18] $\tilde{\Omega}(\sqrt{n})$ [AGGW 18; CCLW 18]
+ gradient oracle	$\tilde{O}(n)$	no known speedup

Gap: up to a quartic speedup?

With complexity measured in the precision ϵ (dimension-independent algorithms): $\Theta(1/\epsilon^2)$ both classically and quantumly — no speedup [Garg, Kothari, Netrapalli, Sherif 21].

Without a known point in K, an $\Omega(n)$ quantum lower bound is known [van Apeldoorn, Gilyén, Gribling, de Wolf 18].

Gradient estimation

Classically, computing $\nabla f(x)$ takes $n+1$ evaluations (finite differences).
Even for linear f, n evaluations are necessary.

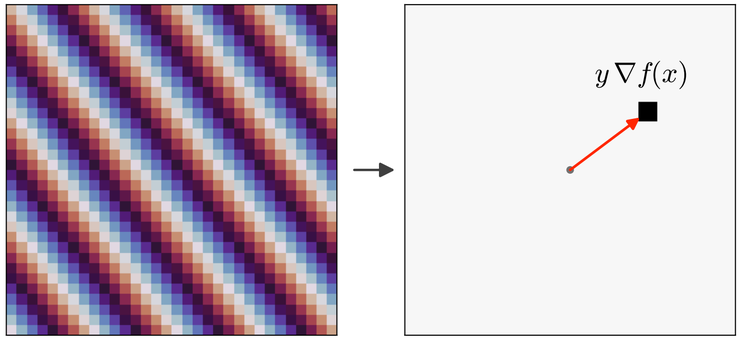
Quantum gradient estimation [Jordan 05]: superpose over a small grid of points near x and make one phase query. Since $f(x+\delta) \approx f(x) + \nabla f(x) \cdot \delta$,

$$\sum_\delta |x+\delta, y\rangle \overset{\hat{O}_f}{\longmapsto} \sum_\delta e^{2\pi i\, y f(x+\delta)} |x+\delta, y\rangle \approx e^{2\pi i\, y f(x)} \sum_\delta e^{2\pi i\, y\, \nabla f(x)\cdot\delta} |x+\delta, y\rangle$$

Up to a shift by the known value x, the first register is the Fourier transform of $|y\,\nabla f(x)\rangle$.

Convex optimization (roughly): $\tilde{O}(n)$ (sub)gradient steps suffice. Each step costs $\tilde{O}(n)$ evaluations classically but $\tilde{O}(1)$ quantumly, so $\tilde{O}(n^2) \to \tilde{O}(n)$ [AGGW 18; CCLW 18].

Given a gradient oracle, this advantage disappears.



Can a quantum algorithm use substantially fewer than n queries?

Quadratic minimization is matrix inversion

Simplest case: $f_{A,b}(x) = \frac{1}{2}\, x^\mathsf{T} A x + b^\mathsf{T} x$ with A positive definite. Best known quantum algorithm still uses $\tilde{O}(n)$ queries.

$\nabla f_{A,b}(x) = Ax + b$, so the minimizer is $x^* = -A^{-1}b$.

Gradient estimation turns evaluation queries into **matrix-vector (MV) queries**: $Ax = \nabla f(x) - b$ and $b = \nabla f(0)$, using $\mathrm{poly}(\log n)$ queries.

Conversely, given b, one matrix-vector query to A can evaluate $f_{A,b}$.

⇒ Minimizing $f_{A,b}$ with evaluation queries has the same cost as computing $A^{-1}b$ with MV queries (up to log factors).

Matrix-vector query: $U_A|x, y\rangle = |x,\, y + Ax\rangle$, $\hat{U}_A|x, y\rangle = e^{2\pi i\, y^\mathsf{T} A x}|x, y\rangle$

Real vectors are discretized with a precision the algorithm chooses; all lower bounds hold at every precision.

Clearly n queries suffice: query $e_1, \ldots, e_n$. **Can querying superpositions of vectors help?**

Prior work:

- Classical matrix-vector query complexity is well studied [Sun, Woodruff, Yang, Zhang 19; Braverman, Hazan, Simchowitz, Woodworth 20].
- Quantum: no speedup for trace, determinant, rank, or linear systems over finite fields [Childs, Hung, Li 21]. Most of these lower bounds use the polynomial method, which does not apply to real inputs.

Results

Quantum query lower bounds for real $n \times n$ matrices with matrix-vector queries:

Task	**Lower bound**
Trace (constant additive error)	$\Omega(n/\log n)$
Sign of the determinant	$\Omega(n)$
Magnitude of the determinant (n-dependent additive error)	$\Omega(n/\log n)$
First column of A^{-1} (accuracy $1/\mathrm{poly}(n)$, condition number $\mathrm{poly}(n)$, even for A positive definite)	$\Omega(n/\log n)$

All of these are $O(n)$, so the bounds are tight up to log factors. This answers open questions of [Childs, Hung, Li 21].

Corollary. Quantum convex optimization with evaluation queries requires $\tilde{\Omega}(n)$ queries, even for a quadratic f over a fixed unit ball with $\epsilon = 1/\mathrm{poly}(n)$.

- Therefore the $\tilde{O}(n)$ algorithms are optimal up to log factors: only quadratic speedup.
- With gradient queries, the bound is still $\tilde{\Omega}(n)$: at most poly(log) speedup.

Technique: the **determinantal witness method**.

Concurrent independent work [Augustino, Chakrabarti, Fontana, Herman, Kim, Ozgul, Voronova 26] gives a near-linear lower bound for convex optimization using related techniques.

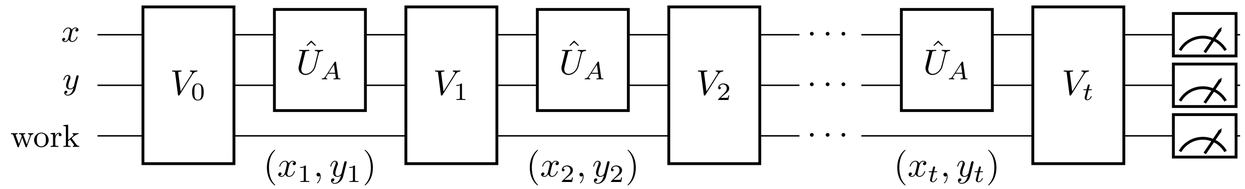
Queries and Fourier rank

A phase query applies a Fourier mode of A with a **rank-one** frequency:

$$\hat{U}_A|x,y\rangle = e^{2\pi i\, y^\mathsf{T} A x}|x,y\rangle = e^{2\pi i\langle A,\, yx^\mathsf{T}\rangle}|x,y\rangle, \qquad \langle A, M\rangle := \operatorname{tr}(A^\mathsf{T} M)$$

Expand a t-query algorithm $V_t\hat{U}_A V_{t-1}\cdots V_1\hat{U}_A V_0$ as a sum over paths. A path through query inputs $(x_1,y_1),\ldots,(x_t,y_t)$ picks up the phase

$$e^{2\pi i\sum_{j=1}^t y_j^\mathsf{T} A x_j} = e^{2\pi i\langle A,M\rangle}, \qquad M = \sum_{j=1}^t y_j x_j^\mathsf{T}$$



Amplitudes contain only frequencies of rank at most t. Probabilities contain only frequencies of rank at most $2t$.

Lemma. For any t-query algorithm, the probability of any outcome has the form

$$p(A) = \int_{R_{2t}} e^{2\pi i\langle A,M\rangle}\,\mathrm{d}\mu(M), \qquad R_r := \{M\in\mathbb{R}^{n\times n} : \operatorname{rank} M \le r\},$$

where μ is a complex measure that does not depend on A.

Compare the polynomial method: degree $\le 2t \quad\longrightarrow\quad$ **Fourier rank** $\le 2t$.

Determinantal witnesses

Goal: find a function β on $n \times n$ matrices whose Fourier transform $\hat{\beta}(M)$ vanishes whenever $\operatorname{rank} M < n$. Then any outcome probability $p(X)$ for an algorithm making $t < n/2$ queries satisfies

$$\int p(X)\,\beta(X)\,\mathrm{d}X = \int_{R_{2t}} \hat{\beta}(M)\,\mathrm{d}\mu(M) = 0.$$

Fourier transform: $\hat{\beta}(M) = \int e^{2\pi i \langle X, M\rangle} \beta(X)\,\mathrm{d}X$

The Fourier transform turns derivatives into multiplication:

$$\widehat{\frac{\partial}{\partial X_{ij}} b(X)}\,(M) = -2\pi i\, M_{ij}\, \hat{b}(M)$$

The **Cayley operator** is the determinant of the matrix of partial derivatives:

$$\det(\partial_X) = \det \begin{pmatrix} \frac{\partial}{\partial X_{11}} & \cdots & \frac{\partial}{\partial X_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial X_{n1}} & \cdots & \frac{\partial}{\partial X_{nn}} \end{pmatrix}$$

For any smooth, rapidly decaying function b, the **determinantal witness** $\beta = \det(\partial_X)\, b$ has

$$\hat{\beta}(M) = (-2\pi i)^n \det(M)\, \hat{b}(M)$$

which vanishes whenever $\operatorname{rank} M < n$.

So $\int p(X)\,\beta(X)\,\mathrm{d}X = 0$ for **every** algorithm making fewer than $n/2$ queries, for any b.

Sign of the determinant: the witness

Task: decide whether $\det A > 0$ or $\det A < 0$.

Choose b to be the Gaussian density on matrices (independent standard normal entries):

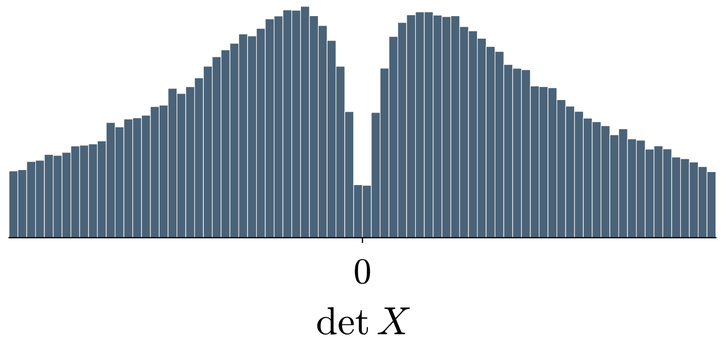
$$\gamma(X) \propto e^{-\frac{1}{2}\langle X, X\rangle}$$

Since $\partial\gamma/\partial X_{ij} = -X_{ij}\,\gamma$, each derivative pulls down a matrix entry, and the Cayley operator reassembles the determinant:

$$\beta(X) = (-1)^n \det(\partial_X)\,\gamma(X) = \det(X)\,\gamma(X)$$

Hard distribution: Probability density proportional to $|\beta(X)|$

Negating a row of X flips the sign of $\det X$ but leaves the distribution unchanged, so each sign has probability $\frac{1}{2}$.

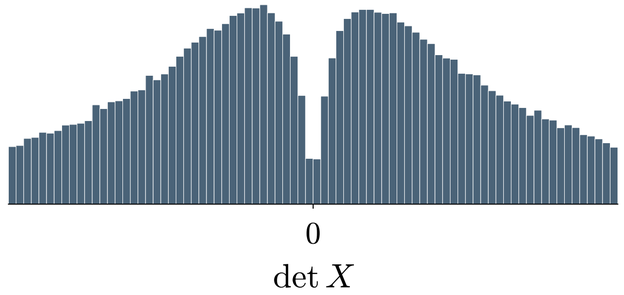


Sign of the determinant: the lower bound

Let $p(X)$ be the probability that the algorithm answers “positive” on input X. With fewer than $n/2$ queries,

$$0 = \int p(X)\, \det(X)\, \gamma(X)\, \mathrm{d}X \;\propto\; \mathbb{E}\big[\, p(X)\, \operatorname{sgn}(\det X) \,\big]$$

Expectations and probabilities are over the hard distribution.



⇒ The algorithm answers “positive” equally often on both halves of the distribution:

$$\Pr[\text{“positive”} \mid \det X > 0] = \Pr[\text{“positive”} \mid \det X < 0]$$

So its average success probability is **exactly** $\frac{1}{2}$: no better than guessing!

Robustness: matrices with huge entries or tiny $|\det X|$ are rare under the hard distribution. Excluding them gives a well-posed problem on which the algorithm still often fails.

Theorem. Deciding the sign of $\det A$ requires $\Omega(n)$ quantum matrix-vector queries, even with the promise that $|\det A|$ is not too small and the entries are bounded.

The determinantal witness method

The same reasoning works for other problems. Suppose a witness β has $\hat{\beta} = 0$ on matrices of rank at most $2t$, and it correlates with the correct answer $D(X) \in \{0, 1\}$. Then every t-query algorithm errs.

Lemma (informal). Any such algorithm fails on some input with probability at least

$$\frac{\left|\int D(X)\,\beta(X)\,\mathrm{d}X\right|}{\int |\beta(X)|\,\mathrm{d}X} \;-\; (\text{fraction of the weight of } \beta \text{ outside the promise})$$

Proof idea: $\int p\,\beta\,\mathrm{d}X = 0$, so $\int D\,\beta\,\mathrm{d}X = \int (D - p)\,\beta\,\mathrm{d}X$, which is small if the algorithm rarely errs.
(For the sign of the determinant, the ratio is $\frac{1}{2}$.)

Recipe: choose b so that $\beta = \det(\partial_X)\,b$ correlates with the answer.

- Even a correlation of $1/\mathrm{poly}(n)$ is enough. Repeating an algorithm $O(\log n)$ times pushes its error below that, so $\Omega(n/\log n)$ queries are needed.

	Polynomial method	**Determinantal witness method**
Input	$x \in \{0,1\}^N$	$A \in \mathbb{R}^{n \times n}$
t queries give	polynomial of degree $\leq 2t$	Fourier transform supported on rank $\leq 2t$
Certificate	dual polynomial [Sherstov 08]	dual witness

Summary and open problems

Queries	Classical	Quantum
evaluation + membership	$\tilde{O}(n^2)$	$\tilde{O}(n)$, $\tilde{\Omega}(n)$
+ gradient oracle	$\tilde{O}(n)$	$\tilde{O}(n)$, $\tilde{\Omega}(n)$

Summary

- Quantum computers can speed up convex optimization at most quadratically with evaluation queries, and only by poly(log) factors given gradients.
- Nearly linear lower bounds for trace, determinant and inversion with real matrix-vector queries (at most logarithmic speedup).
- A new technique, the **determinantal witness method**, based on Fourier rank and the Cayley operator.

Open problems

- Remove the log factors.
- Rank testing over the reals (solved over finite fields [Childs, Hung, Li 21]); other matrix-vector problems.
- Can these techniques be adapted to prove lower bounds outside the matrix-vector model?
- Precision: hardness of optimization only shown for $\epsilon = O(n^{-49})$. What is the joint scaling with n and ϵ?

The AI workflow

These results were generated primarily through conversations with Claude Opus 4.8 and 5, which provided the majority of the technical approach.

Prompt:

Consider the quantum query complexity of matrix inversion in a matrix-vector query model. In this model, there is an unknown n by n matrix A (which is promised to be invertible), and we are given access to an oracle that, for a specified n-component real vector x, outputs the product Ax. The goal is to compute the first column of the inverse of A (exactly) using as few quantum queries as possible. For a quantum computer, this oracle acts in the standard reversible way: a basis state $|x,y\rangle$ for n-component real vectors x,y is mapped to $|x,y+Ax\rangle$, and the oracle acts linearly on superpositions.

The model is subtle because the input and output involve real numbers. Suppose that they can be represented with arbitrarily high precision, and the goal is to compute the output to arbitrarily high precision, with bounded probability of error. If the model needs clarification, ask.

The query complexity of this problem is at most n since querying A on a basis fully determines it. Can you prove a better upper bound? What is the best lower bound you can prove?

This has been answered for finite fields, where the quantum query complexity is Theta(n) (see the paper mv.tex in the folder), but we want to solve it over the reals.

Claude: Long response that conjectures a linear lower bound, with a concrete approach that anticipates the dual witness method (but without the determinantal construction).

Prompt:

Can you propose a concrete approach to step (ii), showing that a spectrum supported on at most t rank-one frequencies cannot produce the right answer unless t = Omega(n)?

Claude: Proposes the determinantal construction using the Cayley operator and describes what would be needed to derive a lower bound from it.

The AI workflow (continued)

Subsequent prompts led to developing the trace lower bound (in retrospect, not the easiest case).

Attempts to prove the inversion lower bound quickly identified the appropriate witness but got stuck on the computation to establish the bound.

The determinant sign and magnitude lower bounds were established as intermediate steps.

The inversion bound was reduced to a conjecture about properties of random matrices.

Claude was prompted to write a program to check the random matrix conjecture numerically. In the process of optimizing the program, it made analytical observations that enabled a proof.

The process up to this point took three days. It then took about three weeks to understand and check all the details of the argument, write it up from scratch (addressing some technical issues that arose along the way), and check the correctness of the manuscript with assistance from GPT and Claude.

The impact of AI on science

Clearly, AI has had and will continue to have a huge impact on all areas of science, including math, theoretical computer science, and theoretical physics.

Pro:

- We can automate many routine tasks.
- We are answering questions that we had no idea how to approach before.

Con:

- We may miss out on the experience and enjoyment of discovering results ourselves.
- We may learn that a result is true but without understanding why (e.g., with formally verified but hard-to-follow proofs).
- There is a lot of uncertainty about how research will change as AI becomes more capable.

Optimistic view: AI is a powerful tool that will allow us to pose and solve harder problems. We should understand its capabilities and use it carefully to learn as much as we can.