Quantum Query Complexity with Matrix-Vector Products

Andrew M. Childs
Joint Center for Quantum Information and Computer Science, Department of Computer Science, and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA

Shih-Han Hung
Joint Center for Quantum Information and Computer Science, Department of Computer Science, and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA

Tongyang Li
Joint Center for Quantum Information and Computer Science, Department of Computer Science, and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA
Center for Theoretical Physics, MIT, Cambridge, MA, USA

Abstract

We study quantum algorithms that learn properties of a matrix using queries that return its action on an input vector. We show that for various problems, including computing the trace, determinant, or rank of a matrix or solving a linear system that it specifies, quantum computers do not provide an asymptotic speedup over classical computation. On the other hand, we show that for some problems, such as computing the parities of rows or columns or deciding if there are two identical rows or columns, quantum computers provide exponential speedup. We demonstrate this by showing equivalence between models that provide matrix-vector products, vector-matrix products, and vector-matrix-vector products, whereas the power of these models can vary significantly for classical computation.

2012 ACM Subject Classification Theory of computation → Quantum query complexity; Theory of computation → Design and analysis of algorithms

Keywords and phrases Quantum algorithms, quantum query complexity, matrix-vector products

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.55

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2102.11349`

Funding AMC and SHH acknowledge support from the Army Research Office (grant W911NF-20-1-0015); the Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams and Accelerated Research in Quantum Computing programs; and the National Science Foundation (grant CCF-1813814).
Tongyang Li: TL acknowledges support from ARO contract W911NF-17-1-0433, NSF grant PHY-1818914, and a Samsung Advanced Institute of Technology Global Research Partnership.

Acknowledgements We thank Robin Kothari for bringing our attention to work on classical algorithms in the matrix-vector and vector-matrix-vector query models, and for providing feedback on an initial version of this paper. We thank Max Simchowitz and Blake Woodworth for a discussion that clarified aspects of their paper [8], and Jialin Zhang for clarifications of her paper [21]. We also thank Ashley Montanaro for pointing out connections to his paper [18], and Joran van Apeldoorn and Sander Gribling for a discussion of their paper [4].

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 55; pp. 55:1–55:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Algorithms for linear algebra problems – for example, solving linear systems and determining basic properties of matrices such as rank, trace, determinant, eigenvalues, and eigenvectors – constitute a fundamental research area in applied mathematics and theoretical computer science. Such tasks have widespread applications in scientific computation, statistics, operations research, and many other related areas. Algorithmic linear algebra also provides a fundamental toolbox that can inspire the design of algorithms in general.

There are several possible models of access to a matrix, and linear-algebraic algorithms can depend significantly on how the input is represented (as discussed further below). One natural model is the *matrix-vector product* (Mv) oracle. For a matrix $M \in \mathbb{F}^{n\times m}$ in a given field $\mathbb{F}$, the Mv oracle takes $x \in \mathbb{F}^m$ as input and outputs $Mx \in \mathbb{F}^n$. Matrix-vector products arise, for example, as the elementary step of the power method (and the related Lanczos method) for computing the largest eigenvector of a matrix. Matrix-vector products also commonly appear in streaming algorithms, especially in the technique of sketching (see the survey [22] for more information).

Recent work has studied the classical complexity of various basic problems in the Mv model. Specifically, Sun, Woodruff, Yang, and Zhang [21] studied the complexities of various linear algebra, statistics, and graph problems using matrix-vector products, and Braverman, Hazan, Simchowitz, and Woodworth [8] proved tight bounds on maximum eigenvalue computation and linear regression in this model. Rashtchian, Woodruff, and Zhu [19] considered a generalization to the vector-matrix-vector product (vMv) oracle, which returns $x^\top My$ for given input vectors $x \in \mathbb{F}^n, y \in \mathbb{F}^m$, and studied the complexity of various linear algebra, statistics, and graph problems in this setting. Table 1 includes a partial summary of these results.

Quantum computers can solve certain problems much faster than classical computers, so it is natural to study quantum query complexity with matrix-vector products. Lee, Santha, and Zhang recently studied the quantum query complexity of graph problems with cut queries [17], which are closely related to matrix-vector queries. For a weighted graph $G = (V, w)$ where $|V| = n$ and w assigns a nonnegative integer weight to each edge, the input of a cut query is a subset $S \subseteq V$ and the output is $|w(S, V \setminus S)|$, the total weight of the edges between S and $V \setminus S$. This can be viewed as a version of the vMv model over $\mathbb{Z}$, with the extra assumptions that $x \in \{0,1\}^n, y \in \{0,1\}^m$ are both boolean and M is a symmetric matrix with nonnegative integer entries. Reference [17] gives quantum algorithms for determining all connected components of G with $O(\log^6 n)$ quantum cut queries, and for outputting a spanning forest of G with $O(\log^8 n)$ quantum cut queries. Both problems require $\Omega(n/\log n)$ classical cut queries, so the quantum algorithms provide exponential speedups.

In other recent work on structured queries for graph problems, Montanaro and Shao studied the problem of learning an unknown graph with "parity queries" [18]: for an unknown graph with adjacency matrix A, the parity oracle takes as input a string x that encodes a subset of the vertices, and returns $x^\top Ax \bmod 2$. This query model is the vMv model over $\mathbb{F}_2$ with the extra restriction that the left and right vectors are identical.

Van Apeldoorn and Gribling studied Simon's problem for linear functions over a prime field $\mathbb{F}_p$ [4]. In this problem, the oracle encodes a linear function $f \colon \mathbb{F}_p \to \mathbb{F}_p$, and the task is to determine if the function is one-to-one, or if there is a one-dimensional subspace $H \subset \mathbb{F}_p$ such that for every $x, x' \in \mathbb{F}_p^n$, $f(x) = f(x')$ if and only if $x - x' \in H$. Such a function can be represented by a square matrix over $\mathbb{F}_p$, and the problem is equivalent to determining whether that matrix is full rank or has nullity 1 using matrix-vector product queries.

Other past work has developed linear algebraic quantum algorithms using different input models. Quantum algorithms for high-dimensional linear algebra have been studied extensively since Harrow, Hassidim, and Lloyd introduced a method for generating a quantum state proportional to the solution of a large, sparse system of linear equations [14]. This algorithm assumes a quantum oracle that determines the locations and values of the nonzero entries of a matrix in any given row or column, and the ability to generate a quantum state that encodes the right-hand side of the linear system. Subsequent work has led to improved and generalized algorithms under similar assumptions. However, it is challenging to find practical applications that achieve speedup over classical computation [2,11]. Recent work by Apers and de Wolf [5] gives polynomial quantum speedup for producing an explicit classical description of the solution of a Laplacian linear system, assuming adjacency-list access to the underlying graph of the Laplacian. Note also that for various problems including determinant estimation, rank testing, linear regression, etc., there is a large separation between the classical query complexities under Mv and entrywise queries ($\tilde{\Theta}(n)$ [21] and $\Theta(n^2)$, respectively). These results show how the model of access to a matrix can significantly impact the complexity of solving linear-algebraic problems. A better understanding of the quantum matrix-vector oracle could therefore provide a useful tool for the design of future quantum algorithms.

Contributions. We conduct a systematic study of quantum query complexity with a matrix-vector oracle for a matrix $M \in \mathbb{F}_q^{m\times n}$, where $\mathbb{F}_q$ is a given finite field. Using this model, we provide results on the quantum query complexities of linear algebra and statistics problems.

First, we prove that various linear algebra problems, including

- computing the trace $\operatorname{tr}(M)$ of $M \in \mathbb{F}_q^{n\times n}$;
- computing the determinant $\det(M)$ of $M \in \mathbb{F}_q^{n\times n}$;
- solving the linear system $Ax = b$ for $A \in \mathbb{F}_q^{n\times n}$; and
- testing whether $\operatorname{rank}(M) = n$ or $\operatorname{rank}(M) \leq n/2$ for a matrix $M \in \mathbb{F}_q^{m\times n}$;

require $\Omega(n)$ quantum queries to the Mv oracle. Since $O(n)$ queries suffice to determine the entire matrix, even classically, these results show that no quantum speedup is possible. (As a side effect, we improve the $\Omega(n/\log n)$ classical lower bound for trace computation [21] to $\Omega(n)$.)

Our quantum lower bound for trace computation applies results of Copeland and Pommersheim [12] by viewing the problem as a special case of *coset identification*. Our lower bounds for other linear algebra problems are all proved by the polynomial method [1,6]. We show how to symmetrize the success probability to a univariate polynomial, and then give a lower bound on the polynomial degree using an observation of Koiran, Nesme, and Portier [16].

On the other hand, we determine the matrix-vector quantum query complexity of several statistics problems, including

- computing the row and column parities of $M \in \mathbb{F}_2^{m\times n}$;
- deciding if there exist two identical columns in $M \in \mathbb{F}_2^{m\times n}$; and
- deciding if there exist two identical rows in $M \in \mathbb{F}_2^{m\times n}$.

Specifically, we prove that their quantum query complexities with an Mv oracle are $O(1)$, $O(\log n)$, and $O(\log m)$, respectively. Compared to the classical bounds using either the Mv oracle [21] or the vMv oracle [19], our quantum algorithms achieve *exponential* quantum speedups.

Technically, these results build upon our observation that the quantum query complexities in the Mv model under left or right multiplication are *identical* (Theorem 5). In particular, one right Mv query can be simulated using one left Mv query, and vice versa. In contrast, classically there is a significant difference between matrix-vector (Mv) and vector-matrix (vM) queries – for example, computing the parity of rows over $\mathbb{F}_2$ only takes $O(1)$ Mv queries, but computing the parity of columns over $\mathbb{F}_2$ requires $\Theta(n)$ Mv queries. In contrast, for both problems a quantum computer can achieve the smaller query complexity by switching to the easier side.

Table 1 Comparison of classical and quantum query complexities with matrix-vector (Mv) and vector-matrix-vector (vMv) product oracles for an $m \times n$ matrix. For trace and linear regression, $m = n$. Known query complexities over $\mathbb{R}$ and $\mathbb{F}_q$ are included for completeness; results over different fields are incomparable in general.

Problem	Classical Mv	Classical vMv	Quantum (this paper)
Trace	$O(n), \Omega(n/\log n)$ for matrix with entries in $\{0, 1, \dots, n^3\}$ & queries with entries in $\{0, 1, \dots, n^C\}$, $C \in \mathbb{N}$ [21]; $\Theta(n)$ over $\mathbb{F}_q$ (Theorem 16)	$O(n), \Omega(n/\log n)$ for matrix with entries in $\{0, 1, \dots, n^3\}$ & queries with entries in $\{0, 1, \dots, n^C\}$, $C \in \mathbb{N}$ [19]; $\Theta(n)$ over $\mathbb{F}_q$ (Theorem 16)	$\Theta(n)$ over $\mathbb{F}_q$ (Theorem 16)
Linear regression	$\Theta(n)$ over $\mathbb{R}$ [8]; $\Theta(n)$ over $\mathbb{F}_q$ (Theorem 24)	$\Theta(n^2)$ over $\mathbb{F}_q$ (Corollary 25)	$\Theta(n)$ over $\mathbb{F}_q$ (Theorem 24)
Rank testing	$k+1$ to distinguish rank $\le k$ from $k' > k$ over $\mathbb{R}$ [21]; $\Theta(n)$ over $\mathbb{F}_q$ (Theorem 27)	$\Omega(k^2)$ to distinguish rank k from $k+1$ over $\mathbb{F}_q$ [19]; $\Omega(n^{2-O(\epsilon)})$ for non-adaptive $(1 \pm \epsilon)$-approximation over $\mathbb{R}$ [19]	$\Theta(\min\{m, n\})$ to distinguish rank $\min\{m, n\}$ from $\le \frac{1}{2}\min\{m, n\}$ over $\mathbb{F}_q$ (Theorem 27)
Two identical columns	$O(n/m)$, $m = \Omega(\log(n/\epsilon))$ over $\mathbb{F}_2$ [21]	$O(n \log n), \Omega(n)$ over $\mathbb{F}_2$ [19]	$O(\log n)$ over $\mathbb{F}_2$ (Corollary 8)
Two identical rows	$O(\log m)$ over $\mathbb{F}_2$ [21]	$O(n \log n), \Omega(n)$ over $\mathbb{F}_2$ [19]	$O(\log m)$ over $\mathbb{F}_2$ (Corollary 8)
Majority of columns	$\Omega(n/\log n)$ for binary matrices over $\mathbb{R}$ [21]	$\Theta(n^2)$ over $\mathbb{F}_2$ [19]	$O(1)$ for binary matrices over $\mathbb{R}$ (Corollary 10)
Majority of rows	$O(1)$ for binary matrices over $\mathbb{R}$ [21]	$\Theta(n^2)$ over $\mathbb{F}_2$ [19]	$O(1)$ for binary matrices over $\mathbb{R}$ (Corollary 10)
Parity of columns	$\Theta(n)$ over $\mathbb{F}_2$ [21]	$\Theta(n)$ over $\mathbb{F}_2$ (Lemma 7)	$O(1)$ over $\mathbb{F}_2$ (Corollary 6)
Parity of rows	$O(1)$ over $\mathbb{F}_2$ [21]	$\Theta(m)$ over $\mathbb{F}_2$ (Lemma 7)	$O(1)$ over $\mathbb{F}_2$ (Corollary 6)

Our results are summarized in Table 1, including some implications of our results for classical query complexity and a few additional results over $\mathbb{R}$. Note that there can be large gaps between the classical query complexities with Mv and vMv queries, but they are the same in the quantum setting due to an equivalence between quantum Mv and vMv queries (Theorem 11), which follows along similar lines to the equivalence between Mv and vM queries. The Mv–vMv equivalence is closely related to a similar equivalence shown in the work of Lee, Santha, and Zhang [17], as we discuss further in Section 3.2.

Open questions. Our paper leaves several natural open questions for future investigation:

- For linear algebra problems such as those we studied, can we also prove quantum query lower bounds for matrices over the real field $\mathbb{R}$? Our proofs rely on the polynomial method, and it is unclear how to adapt them to a setting with continuous input.
- Can we prove a quantum lower bound for the task of minimizing a quadratic form $f(x) = \frac{1}{2}x^\top Ax + b^\top x$, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$? Note that f is minimized at $x = -A^{-1}b$, and we can determine the vector b and implement Mv queries to the matrix A using fast quantum gradient computation [15], so this is closely related to the previous open question. Quadratic form minimization is a special case of optimizing a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ by quantum evaluation queries, where previous works [3, 10, 13] left a quadratic gap between the best known quantum upper and lower bounds of $\tilde{O}(n)$ and $\Omega(\sqrt{n})$, respectively.
- For the finite field case, can we identify other problems with quantum speedup over the classical matrix-vector oracle, or find advantage compared to other quantum oracles such as entrywise queries?

Organization. We review necessary background in Section 2. We prove the equivalence of quantum matrix-vector and vector-matrix-vector product oracles in Section 3. In Section 4, we prove tight quantum query complexity lower bounds on various linear algebra problems, including trace, determinant, linear systems, and rank.

2 Preliminaries

2.1 The quantum query model

Given a set X and an abelian group G, let $f\colon X \to G$ be a function. Access to f is provided by a black-box unitary operation $U_f\colon |x,y\rangle \mapsto |x, y + f(x)\rangle$ for all $x \in X$ and $y \in G$. We call an application of U_f a (standard) query.

For a finite abelian group G, the Fourier transform over G is

$$F_G \coloneqq \frac{1}{|G|^{1/2}} \sum_{x\in G} \sum_{y\in\hat{G}} \chi_y(x) |y\rangle\langle x|, \tag{1}$$

where $\hat{G}$ is a complete set of characters of G, and $\chi_y\colon G \to \mathbb{C}$ denotes the y^{th} character of G. Since $\hat{G} \cong G$, we label elements of $\hat{G}$ using elements of G. Note that χ_y is a group homomorphism, i.e., $\chi_y(x+z) = \chi_y(x)\chi_y(z)$. In addition, the characters satisfy the orthogonality condition

$$\frac{1}{|G|} \sum_{z\in G} \chi_y(z)^* \chi_w(z) = \delta_{yw}. \tag{2}$$

A phase query is defined as a standard query conjugated by the Fourier transform acting on the output register. In other words, for $x \in X$ and $y \in G$, a phase query acts as

$$\begin{aligned} |x,y\rangle &\xmapsto{\mathbb{1}\otimes F_G^\dagger} \frac{1}{|G|^{1/2}} \sum_{z\in G} \chi_y(z)^* |x,z\rangle \\ &\xmapsto{U_f} \frac{1}{|G|^{1/2}} \sum_{z\in G} \chi_y(z)^* |x, z + f(x)\rangle \\ &\xmapsto{\mathbb{1}\otimes F_G} \frac{1}{|G|} \sum_{z\in G} \chi_y(z)^* \chi_w(z + f(x)) |x,w\rangle = \chi_y(f(x))|x,y\rangle. \end{aligned} \tag{3}$$

The equality in (3) follows from the orthogonality condition in (2). Since one can simulate a phase query using a single standard query and vice versa, the query complexities of any problem are equal with these two models.

Over a finite field $\mathbb{F}_q$ for prime power $q = p^r$, the Fourier transform over $\mathbb{F}_q$ is the unitary transformation $|x\rangle \mapsto q^{-1/2} \sum_{y \in \mathbb{F}_q} e(xy)|y\rangle$, where the exponential function $e\colon \mathbb{F}_q \to \mathbb{C}$ is defined as $e(z) := e^{2\pi i \mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(z)/p}$ and the trace function $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}\colon \mathbb{F}_q \to \mathbb{F}_p$ is defined as $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(z) := z + z^p + z^{p^2} + \cdots + z^{p^{r-1}}$.

Over the field of real numbers, the quantum Fourier transform is

$$F_{\mathbb{R}} := \int_{\mathbb{R}} \mathrm{d}y \int_{\mathbb{R}} \mathrm{d}x \, e^{2\pi i y x} |y\rangle\langle x|. \tag{4}$$

The basis states $\{|x\rangle : x \in \mathbb{R}\}$ are normalized to the Dirac delta function, i.e., for $x, x' \in \mathbb{R}$, $\langle x'|x\rangle = \delta(x - x')$. Here the Dirac delta function δ satisfies $\int_{\mathbb{R}} \mathrm{d}x' \, \delta(x - x') f(x') = f(x)$ for any function f. Furthermore, we have $\int_{\mathbb{R}} \mathrm{d}y \, e^{2\pi i y(x-x')} = \delta(x - x')$. By direct calculation using these facts, $F_{\mathbb{R}}^{\dagger} F_{\mathbb{R}} = \int_{\mathbb{R}} \mathrm{d}x \, |x\rangle\langle x| = \mathbb{1}$.

While we can formally consider a model of query complexity over $\mathbb{R}$ with arbitrary precision, its practical instantiation requires discrete approximation. We can achieve precision ϵ by approximating real numbers with $s = O(\log(1/\epsilon))$ bits, and can then replace the continuous Fourier transform with the discrete Fourier transform over $\mathbb{Z}_{2^s}$. It is straightforward to show that a discretized phase query over $\mathbb{Z}_{2^s}$ can be implemented by Fourier transforming a standard query that maps discretized inputs to discretized function values.

2.2 The coset identification problem

Copeland and Pommersheim studied a kind of quantum query problem that they call the *coset identification problem* [12]. They define this problem in a generalized query model where the black box does not necessarily perform a standard or phase query, although their definition includes those cases. In the coset identification problem, we fix a finite group G and a subgroup $H \leq G$. The algorithm is given access to a unitary transformation $\pi(g)$, where π is a representation of G on vector space V. When π is given, the vector space V is called the representation space (or simply, the representation) of G [20, Chapter 1]. The goal is to determine which coset of H the unknown element $g \in G$ belongs to.

▶ **Definition 1** (Coset identification problem [12]). *A coset identification problem for a finite group G and subgroup $H \leq G$ is a 3-tuple (π, V, F) such that*

- *π is a unitary representation of G in the complex vector space V, and*
- *F is a function constant on left cosets of $H \leq G$ and distinct on distinct cosets, i.e., $F(g) = F(g')$ if and only if $g' = gh$ for some $h \in H$.*

Given a black box that performs the unitary transformation $\pi(g)$, the goal is to compute $F(g)$.

Copeland and Pommersheim show that the optimal success probability of a t-query algorithm for a coset identification problem can be calculated by taking, over all irreps Y of H, the maximum of the fraction of the induced representation $Y^{\uparrow}$ of G shared with $V^{\otimes t}$. Furthermore, the optimal algorithm can be non-adaptive. For a representation V, let $I(V)$ denote the set of irreducible characters of G appearing in V.

▶ **Theorem 2** (Optimal success probability of coset identification [12, Corollary 5.7]). *The optimal success probability of any t-query quantum algorithm $\mathcal{A}$ for the coset identification problem (π, V, F) for finite group G and subgroup $H \leq G$, under uniformly random inputs in G, is*

$$\Pr[\mathcal{A}^{\pi(g)} = F(g)] = \max_{Y} \frac{\dim Y^{\uparrow}_{V^{\otimes t}}}{\dim Y^{\uparrow}}, \tag{5}$$

where the probability is maximized over all irreducible representations Y of H, $Y^{\uparrow}$ is the induced representation of G, and A_B is the maximal subrepresentation of A such that $I(A_B) \subseteq I(B)$ for representations A, B.

The *oracle discrimination problem* is the special case of the coset identification problem where H is the trivial group, i.e., the function F is injective. In this case, $Y^{\uparrow} = \operatorname{span}\{|g\rangle : g \in G\}$.

▶ **Corollary 3** (Optimal success probability of oracle discrimination [12, Theorem 4.2])**.** *The optimal success probability of the oracle discrimination problem is*

$$\frac{1}{|G|} \sum_{i \in I(V^{\otimes t})} d_i^2, \tag{6}$$

where $I(V^{\otimes t})$ is the irrep content of $(\pi^{\otimes t}, V^{\otimes t})$ and d_i is the dimension of irrep $i \in I(V^{\otimes t})$.

We consider the complexity of standard queries in the matrix-vector model. In this model, oracle access to a matrix $M \in \mathbb{F}^{m\times n}$ for field $\mathbb{F}$ and positive integers m, n is the unitary operation $U(M)\colon |x, y\rangle \mapsto |x, y + Mx\rangle$. The map U is a representation of the additive group of matrices since it is a group homomorphism satisfying $U(M)U(N) = U(M + N)$ for all matrices M, N of the same dimensions. A phase query is also a unitary representation since it is a standard query conjugated by a fixed unitary matrix (the quantum Fourier transform).

2.3 The polynomial method

We will use the polynomial method to obtain quantum lower bounds. Here we state a version for non-boolean functions as used in [1].

▶ **Lemma 4.** *Let $\mathcal{A}$ be a t-query quantum algorithm given access to the input $x \in [m]^n$ for $m, n \in \mathbb{Z}$ through oracle $U_x\colon |i, j\rangle \mapsto |i, j + x_i\rangle$ for $i \in [n]$ and $j \in [m]$. The acceptance probability of $\mathcal{A}$ on input x is a degree-$(2t)$ polynomial in $x_1, \ldots, x_n$.*

3 Equivalence of matrix-vector and vector-matrix-vector products

In this section, we show that the matrix-vector and vector-matrix-vector models are equivalent, i.e., for any problem, the quantum query complexities in these models differ by at most a constant factor. Furthermore, we show that in the matrix-vector model, left matrix-vector products and right matrix-vector products are equivalent. This is in stark contrast to the classical case where these query complexities can differ significantly, as mentioned in Section 1 and discussed further below.

3.1 Left and right matrix-vector queries

We first show that left matrix-vector products and right matrix-vector products are equivalent.

▶ **Theorem 5.** *Quantum query complexities in the left and right matrix-vector models over a finite field are identical. In particular, one right* Mv *query can be simulated using one left* Mv *query, and vice versa.*

Proof. For input matrix $M \in \mathbb{F}_q^{n\times m}$, a matrix-vector (Mv) query applies the unitary transformation

$$U^{\mathsf{Mv}}(M)\colon |x, y\rangle \mapsto |x, y + Mx\rangle \tag{7}$$

for every $x \in \mathbb{F}_q^m$ and $y \in \mathbb{F}_q^n$. Conjugating by a quantum Fourier transform on the output register yields a phase query

$$\begin{aligned}
|x, y\rangle &\xmapsto{\mathbb{1} \otimes F^\dagger_{\mathbb{F}_q^n}} q^{-1/2} \sum_z e(-y^\top z)|x, z\rangle \\
&\xmapsto{U^{\mathsf{Mv}}(M)} q^{-1/2} \sum_z e(-y^\top z)|x, z + Mx\rangle \\
&\xmapsto{\mathbb{1} \otimes F_{\mathbb{F}_q^n}} q^{-1} \sum_{z,w} e(-y^\top z + w^\top (z + Mx))|x, w\rangle \\
&= \sum_w \delta[y = w] e(-y^\top z + w^\top (z + Mx))|x, w\rangle \\
&= e(y^\top Mx)|x, y\rangle. \qquad (8)
\end{aligned}$$

We denote this unitary transformation by $\widetilde{U^{\mathsf{Mv}}}(M)$.

Conjugating a phase query by a swap gate, we have

$$\begin{aligned}
|x, y\rangle &\xmapsto{\mathsf{SWAP}} |y, x\rangle \\
&\xmapsto{\widetilde{U^{\mathsf{Mv}}}(M)} e(x^\top My)|y, x\rangle \\
&\xmapsto{\mathsf{SWAP}} e(x^\top My)|x, y\rangle \\
&= e(y^\top M^\top x)|x, y\rangle. \qquad (9)
\end{aligned}$$

This yields $\widetilde{U^{\mathsf{Mv}}}(M^\top)$, which in turn gives $U^{\mathsf{Mv}}(M^\top)$ upon conjugation by an inverse quantum Fourier transform on the output register. Thus one can simulate the oracle $U^{\mathsf{Mv}}(M^\top)$ using one query to $U^{\mathsf{Mv}}(M)$, showing equivalence of the two models. ◀

In contrast to Theorem 5, Sun, Woodruff, Yang, and Zhang show that for the task of computing the row parities of an $m \times n$ matrix M over $\mathbb{F}_2$, the left query complexity is $\Omega(m)$, whereas the right query complexity is 1 [21]. Thus we have shown that computing column parities over $\mathbb{F}_2$ in the Mv model has quantum query complexity 1, significantly less than the classical query complexity of $\Omega(n)$.

▶ **Corollary 6.** *The query complexity of computing the row parities and the column parities of an $m \times n$ matrix over $\mathbb{F}_2$ is 1.*

Note that it is easy to understand the randomized query complexities of these problems in the vMv model.

▶ **Lemma 7.** *The randomized query complexities of computing the row parities and the column parities of an $m \times n$ matrix over $\mathbb{F}_2$ are $\Theta(m)$ and $\Theta(n)$, respectively.*

Proof. Each query reveals one bit of information, while the row parities convey m bits, giving a lower bound of $\Omega(m)$. An algorithm querying $(e_1, 1^n), \ldots, (e_m, 1^n)$ learns the row parities with probability 1, giving an upper bound of m. The query complexity of column parities follows immediately from the symmetry of the vMv oracle. ◀

The randomized query complexities of determining if there exist identical columns or identical rows are $\Theta(n/m)$ and $\Theta(\log m)$, respectively [21]. Theorem 5 implies that for identical columns, there is an exponential quantum speedup.

▶ **Corollary 8.** *The query complexities of deciding if there exist two identical columns and rows in a $m \times n$ matrix over $\mathbb{F}_2$ are $O(\log n)$ and $O(\log m)$, respectively.*

Proof. By Theorem 5, it suffices to give an algorithm for determining if there are two identical rows. To make the proof self-contained, we describe the algorithm of Sun, Woodruff, Yang, and Zhang [21, Section 4.2]. The algorithm makes q random queries $v_1, \dots, v_q$, the entries of which are sampled uniformly. The algorithm outputs 1 if and only if there exist two entries i, j such that $(Mv_k)_i = (Mv_k)_j$ for $k \in [q]$.

To analyze the performance, for any two identical rows $m_i^\top, m_j^\top$, $\Pr_v[m_i^\top v = m_j^\top v] = 1$. For $m_i \neq m_j$, $\Pr_v[m_i^\top v = m_j^\top v] \leq 1/2$. Therefore for a matrix that has two identical rows, the algorithm outputs 1 with probability 1. On the other hand, for a matrix that has no identical rows, the algorithm outputs 1 with probability

$$\begin{aligned}\Pr_{v_1,\dots,v_q}[\exists i, j \in [m],\ \forall \ell \in [q], m_i^\top v_\ell = m_j^\top v_\ell] &\leq \sum_{i,j\in[m], i\neq j} \Pr_{v_1,\dots,v_q}[\forall \ell \in [q], m_i^\top v_\ell = m_j^\top v_\ell] \\ &\leq \binom{m}{2} 2^{-q}. \qquad (10)\end{aligned}$$

Taking $q = 2\log m$, the probability is no more than $\frac{1}{2} - \frac{1}{2m}$. ◀

The equivalence of left and right queries also holds over the reals.

▶ **Theorem 9.** *Quantum query complexities in the left and the right matrix-vector models over $\mathbb{R}$ are identical. In particular, one right* Mv *query can be simulated using one left* Mv *query, and vice versa.*

Proof. The same idea as in the proof of Theorem 5 applies. First, a phase query can be simulated by conjugating a standard query by the quantum Fourier transform. This yields $\widetilde{U^{\mathsf{Mv}}}(M)$. Conjugating a phase query by a swap gate gives $\widetilde{U^{\mathsf{Mv}}}(M^\top)$ with the same calculation as in (9). This in turn yields $U^{\mathsf{Mv}}(M^\top)$ upon conjugating $\widetilde{U^{\mathsf{Mv}}}(M^\top)$ by an inverse quantum Fourier transform. ◀

Note that with finite precision, a phase query can be simulated using the quantum Fourier transform over an integer modulus (see Section 2.1 for details).

As an example, we determine the query complexity of the majority of rows or columns: given a binary matrix $M \in \{0,1\}^{m\times n}$, compute the majority of each row or column over $\mathbb{R}$, i.e., for each row or column, determine if there are more 1s than 0s.

▶ **Corollary 10.** *In the matrix-vector model, the query complexities of computing the majorities of rows and columns of an $m \times n$ matrix over $\mathbb{R}$ are* 1.

Proof. By Theorem 9, it suffices to show the query complexity of the majority of rows is 1. With a single query $(1, 1, \dots, 1)^\top$, the majority of each row is determined. ◀

This result is not significantly affected by considering computation with finite precision. The number of 1s in each row and each column is an integer in $[0, k]$ for $k = \max\{m, n\}$. Thus a truncation with $O(\log k)$ bits suffices to perform the computation with no error.

3.2 The vector-matrix-vector model

We now relate the power of the matrix-vector and vector-matrix-vector query models. In the vector-matrix-vector model, the algorithm is given access to M via $U^{\mathsf{vMv}}\colon |x, y, a\rangle \mapsto |x, y, a + y^\top Mx\rangle$. We can simulate one vMv query using two Mv queries and an ancilla space storing a matrix-vector product:

$$\begin{aligned}
|x,y,a\rangle &\xmapsto{U^{\mathsf{Mv}}(M)} |x,y,a\rangle|Mx\rangle \\
&\mapsto |x,y,a+y^\top Mx\rangle|Mx\rangle \\
&\xmapsto{U^{\mathsf{Mv}}(M)^\dagger} |x,y,a+y^\top Mx\rangle|0\rangle. \qquad (11)
\end{aligned}$$

On the other hand, an Mv phase query (defined previously in (8)) can be simulated using a vMv phase query by setting $a=1$:

$$|x,y,1\rangle \mapsto e(y^\top Mx)|x,y,1\rangle. \qquad (12)$$

Such a vMv phase query can be constructed using one application of U^{vMv}:

$$\begin{aligned}
|x,y,a\rangle &\xmapsto{\mathbb{1}\otimes\mathbb{1}\otimes F_{\mathbb{F}_q}^\dagger} \sum_b e(-ab)|x,y,b\rangle \\
&\xmapsto{U^{\mathsf{vMv}}(M)} \sum_b e(-ab)|x,y,b+y^\top Mx\rangle \\
&\xmapsto{\mathbb{1}\otimes\mathbb{1}\otimes F_{\mathbb{F}_q}} \sum_{bc} e(-ab+c(b+y^\top Mx))|x,y,c\rangle \\
&= e(ay^\top Mx)|x,y,a\rangle. \qquad (13)
\end{aligned}$$

Thus we have shown the following.

▶ **Theorem 11.** *Quantum query complexities in the matrix-vector and vector-matrix-vector models differ by at most a constant factor. In particular, one* vMv *query can be simulated using two* Mv *queries, and one* Mv *query can be simulated using one* vMv *query.*

This is again in stark contrast to the classical case, where the Mv model can be much more powerful than the vMv model. For example, for distinguishing a full-rank matrix from a rank-$(n-1)$ matrix, the randomized query complexity in the vMv model is $\Omega(n^2)$ [19], while the randomized query complexity in the Mv model is $O(n)$ [21].

Note that Lee, Santha, and Zhang [17] previously studied the equivalence between quantum Mv and vMv oracles. They focus on the special case where the matrix M is the adjacency matrix of a graph with nonnegative integer weights and the inputs $x \in \{0,1\}^n, y \in \{0,1\}^m$ are boolean. In that setting, they prove equivalence between the vMv oracle and the additive oracle $a\colon 2^{[n]} \to \mathbb{Z}$ that returns $a(S) = \sum_{(u,v)\in S^{(2)}} w(u,v)$ for $S \subseteq [n]$, where $S^{(2)}$ denotes the set of cardinality-2 subsets of S. They also study relationships with other oracles that encode specific information about graphs (cuts, disjoint cuts, etc.; see Section 4 of [17]). In contrast, our Theorem 5, Theorem 9, and Theorem 11 work for inputs and matrices in fields, and do not apply to other graph oracles. While these results are, strictly speaking, incomparable, they are closely related, both following from a generalization of the Bernstein-Vazirani algorithm [7].

4 Linear algebra over finite fields

We now consider the quantum query complexity of particular linear algebra problems in the matrix-vector query model. Specifically, we consider learning the trace (Section 4.1), computing the null space and determinant (Section 4.2), solving linear systems (Section 4.3), and estimating the rank (Section 4.4).

4.1 Trace

In this section, we show that the quantum query complexity of computing the trace of an $n \times n$ matrix over $\mathbb{F}_q$ is $\Theta(n)$. Since there is a trivial algorithm that computes the trace by learning the entire matrix using n queries, we focus on the lower bound.

Learning the trace can be regarded as a coset identification problem (defined in Section 2.2) in the group $G = \mathbb{F}_q^{n\times n}$ with subgroup $H = \{M \in \mathbb{F}_q^{n\times n} : \mathrm{tr}M = 0\} \cong \mathbb{F}_q^{n^2-1}$. The irreducible characters χ_Z of H are indexed by $Z \in \mathbb{Z}_m^{n\times n}$ with $Z_{nn} = 0$, and satisfy $\chi_Z(M) = e(\langle Z, M\rangle)$ where $\langle Z, M\rangle := \sum_{i,j=1}^{n} Z_{ij}M_{ij}$.

4.1.1 Learning the trace over $\mathbb{F}_2$

First we consider the case $q = 2$. Then the irreducible characters χ_Z of H for $Z \in \mathbb{Z}_m^{n\times n}$ (with $Z_{nn} = 0$) satisfy

$$\chi_Z(M) = (-1)^{\langle Z,M\rangle}. \tag{14}$$

For irredicible character Z, the induced representation can be decomposed into two irreducible characters of G:

$$\chi_{Z,0}(M) = (-1)^{\langle Z,M\rangle}; \qquad \chi_{Z,1}(M) = (-1)^{\langle Z,M\rangle+\mathrm{tr}M}. \tag{15}$$

It is easy to check that for $M \in G$, $\chi_{Z,0}(M + E^{(nn)}) = \chi_{Z,0}(M)$ and $\chi_{Z,1}(M + E^{(nn)}) = -\chi_{Z,1}(M)$, where $E^{(ij)}$ is an $n \times n$ matrix whose entries are zero except that $(E^{(ij)})_{ij} = 1$. We emphasize that in (15), $M \in G$ (rather than in H since we are now looking at the representations of the entire group), and $Z_{nn} = 0$.

On the other hand, recall that the phase query oracle is $U(M)\colon |x, y\rangle \mapsto (-1)^{y^\top Mx}|x, y\rangle$, which is a unitary representation of M with character $\xi(M) := \mathrm{tr}(U(M)) = \sum_{x,y\in\mathbb{F}_2^n}(-1)^{y^\top Mx}$. To determine the optimal success probability, we calculate the irrep content of $U^{\otimes t}$. The character of $U^{\otimes t}$ is ξ^t, satisfying

$$\begin{aligned}\mathrm{tr}(U^{\otimes t}(M)) &= \mathrm{tr}(U(M))^t = (\xi(M))^t \\ &= \left(\sum_{x,y\in\mathbb{F}_2^n}(-1)^{y^\top Mx}\right)^t = \sum_{x_1,\dots,x_t,y_1,\dots,y_t\in\mathbb{F}_2^n}(-1)^{\sum_i y_i Mx_i}.\end{aligned} \tag{16}$$

Thus it has non-zero Fourier coefficient at W if and only if $W \in R_t$, where R_t is the set of matrices of rank no more than t.

We now check containment of the irreps (15) in $U^{\otimes t}$. We find

$$m_{Z,0}^{(t)} = \langle \xi^t, \chi_{Z,0}\rangle > 0 \iff Z \in R_t, \qquad m_{Z,1}^{(t)} = \langle \xi^t, \chi_{Z,0}\rangle > 0 \iff Z + \mathbb{1}_n \in R_t. \tag{17}$$

By Theorem 2, to succeed with probability better than 1/2, we must choose a Z such that both $m_{Z,0}^{(t)} > 0$ and $m_{Z,1}^{(t)} > 0$. However, now we show this is impossible with $t < n/2$.

▶ **Lemma 12.** *The set $\{Z : m_{Z,0}^{(t)} > 0 \wedge m_{Z,1}^{(t)} > 0\}$ is empty for $t < n/2$.*

Proof. We show that the set is non-empty only if $t \geq n/2$. Suppose there exists Z such that $m_{Z,0} > 0$ and $m_{Z,1} > 0$. By (17), $Z \in R_t$ and $Z + \mathbb{1}_n \in R_t$. Since the ranks of Z and $Z + \mathbb{1}_n$ are no more than t, we conclude that the rank of $\mathbb{1}_n = Z + Z + \mathbb{1}_n$ is no more than $2t$. Therefore $t \geq n/2$. ◀

This implies an $n/2$ lower bound, formally stated as follows.

▶ **Lemma 13.** *For $t < n/2$, any t-query quantum algorithm computing the trace of an $n \times n$ matrix over $\mathbb{F}_2$ succeeds with probability at most $1/2$.*

Proof. By Theorem 2 and Lemma 12, the optimal success probability for a uniformly random matrix in $\mathbb{F}_2^{n\times n}$ is

$$\frac{1}{2}\max_{Z}\sum_{b=0}^{1}\delta[m_{Z,b} > 0] \leq \frac{1}{2} \tag{18}$$

for $t < n/2$. ◀

On the upper bound side, we present an $\lceil n/2 \rceil$-query quantum algorithm, showing that the above lower bound is achievable.

▶ **Lemma 14.** *In the matrix-vector query model, there exists an $\lceil n/2 \rceil$-query quantum algorithm that computes the trace of an $n \times n$ matrix over $\mathbb{F}_2$ with probability 1.*

Proof. First we pad the matrix with one extra zero row and one extra zero column if n is odd, and denote the padded matrix by M'. Let $\ell = \lceil n/2 \rceil$. It is clear that one query to $M' \in \mathbb{F}_2^{2\ell \times 2\ell}$ can be simulated using one query to M. By Theorem 2, it suffices to find an irreducible character such that both $m_{Z,0} > 0$ and $m_{Z,1} > 0$. Now consider

$$Z = \begin{bmatrix} \mathbb{1}_\ell & 0 \\ 0 & 0 \end{bmatrix} = \sum_{i=1}^{\ell} e_i e_i^\top, \qquad Z + \mathbb{1}_{2\ell} = \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{1}_\ell \end{bmatrix} = \sum_{i=\ell+1}^{2\ell} e_i e_i^\top. \tag{19}$$

The algorithm first prepares the state

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}|e_1,\ldots,e_\ell\rangle|e_1,\ldots,e_\ell\rangle + \frac{1}{\sqrt{2}}|e_{\ell+1},\ldots,e_{2\ell}\rangle|e_{\ell+1},\ldots,e_{2\ell}\rangle. \tag{20}$$

Making ℓ phase queries in parallel, we have

$$\begin{aligned} |\psi_M\rangle &= \widetilde{U^{\mathsf{Mv}}}(M')|\psi_0\rangle \\ &= \frac{1}{\sqrt{2}}(-1)^{\sum_{i=1}^{\ell} M'_{ii}}|e_1,\ldots,e_\ell\rangle|e_1,\ldots,e_\ell\rangle \\ &\quad + \frac{1}{\sqrt{2}}(-1)^{\sum_{i=\ell+1}^{2\ell} M'_{ii}}|e_{\ell+1},\ldots,e_{2\ell}\rangle|e_{\ell+1},\ldots,e_{2\ell}\rangle. \end{aligned} \tag{21}$$

Measuring in the basis $\{|\psi_0\rangle\langle\psi_0|, |\psi_1\rangle\langle\psi_1|\}$, where

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}|e_1,\ldots,e_\ell\rangle|e_1,\ldots,e_\ell\rangle - \frac{1}{\sqrt{2}}|e_{\ell+1},\ldots,e_{2\ell}\rangle|e_{\ell+1},\ldots,e_{2\ell}\rangle, \tag{22}$$

the algorithm outputs the trace with probability 1. ◀

The results of this section are summarized in the following theorem.

▶ **Theorem 15.** *In the matrix-vector query model, no quantum algorithm can compute the trace of an $n \times n$ matrix over $\mathbb{F}_2$ with probability better than 1/2 using fewer than $n/2$ queries, and there exists a quantum algorithm that succeeds with probability 1 using $\lceil n/2 \rceil$ queries.*

4.1.2 Learning the trace over $\mathbb{F}_q$

Now we prove a linear lower bound for the task of learning the trace over $\mathbb{F}_q$. The proof idea is the same as in the case $q = 2$, generalized to any finite field.

▶ **Theorem 16.** *In the matrix-vector query model over $\mathbb{F}_q$, computing the trace of an $n \times n$ matrix with probability more than $1/q$ requires at least $n/2$ queries.*

Proof. The induced representation of Z (defined in the second paragraph of Section 4.1) can be decomposed into q 1-dimensional irreps whose characters are

$$\chi_{Z,s}(M) = e(\langle Z, M\rangle + s \cdot \mathrm{tr} M) = e(\langle Z + s\mathbb{1}_n, M\rangle) \tag{23}$$

for $s \in \mathbb{F}_q$. Again, recall that a phase query oracle $U(M)\colon |x, y\rangle \mapsto e(y^\top Mx)|x, y\rangle$ is a unitary representation of M. The character of U is the trace $\xi(M) := \mathrm{tr}(U(M)) = \sum_{x,y\in\mathbb{F}_q^n} e(y^\top Mx)$. The optimal success probability is determined by the irrep content of $U^{\otimes t}$, and the character of $U^{\otimes t}$ is ξ^t, satisfying

$$\mathrm{tr}(U^{\otimes t}(M)) = \xi^t(M) = \sum_{x_1,\dots,x_t,y_1,\dots,y_t\in\mathbb{F}_q^n} e\left(\sum_{i=1}^{t} y_i^\top Mx_i\right). \tag{24}$$

Thus for every $s \in \mathbb{Z}_m$,

$$m_{Z,s}^{(t)} = \langle \xi^t, \chi_{Z,s}\rangle > 0 \iff Z + s \cdot \mathbb{1}_n \in R_t, \tag{25}$$

where R_t is the set of matrices of rank no more than t. Since $\mathbb{1}_n \notin R_{n-1}$, we conclude for $t < n/2$ the success probability is at most $1/q$, as claimed. ◀

4.2 Null space

In this section, we show a linear lower bound on the matrix-vector quantum query complexity of computing the rank of a matrix $M \in \mathbb{F}_q^{m\times n}$ for $m \geq n$. This is without loss of generality since for $m < n$, by Theorem 5, we can simulate oracle access to $M^\top$ using one query to M.

The rank problem is an instance of the hidden subgroup problem (HSP) over $\mathbb{F}_q^m$ since two vectors map to the same value if and only if their difference is in the null space. However, the lower bound for the abelian HSP [16] does not directly apply to this problem since the instance is more structured – specifically, the subgroup hiding function is a linear transformation.

We recall some standard facts from linear algebra over finite fields. For $\ell \geq m$, let $\binom{\ell}{m}_q := \frac{\prod_{i=0}^{m-1}(q^\ell - q^i)}{\prod_{i=0}^{m-1}(q^m - q^i)}$ denote a Gaussian binomial coefficient.

▶ **Lemma 17.** *The number of m-dimensional subspaces of an ℓ-dimensional space over $\mathbb{F}_q$ is $\binom{\ell}{m}_q$.*

▶ **Lemma 18.** *For integers $k \leq m \leq \ell$ and any k-dimensional space V over $\mathbb{F}_q$, the number of m-dimensional subspaces of an ℓ-dimensional space containing V is $\binom{\ell-k}{m-k}_q$.*

For proofs of these facts, see for example [9, Lemma 9.3.2].

Computing the rank. Now we consider the problem of computing the rank of a matrix $M \in \mathbb{F}_q^{m\times n}$ for $m \geq n$. A matrix M has rank r if and only if its null space is $(n - r)$-dimensional.

By Lemma 4, the success probability of a t-query algorithm is a degree-$2t$ polynomial in δ_{xy}. This polynomial P can be written as

$$P(\delta) = \sum_{S \subseteq \mathbb{F}_q^n \times \mathbb{F}_q^m} c_S \prod_{(x,y) \in S} \delta_{xy}, \tag{26}$$

with $c_S = 0$ for $|S| > \deg(P)$. For an input M, the assignments to these variables are $\delta_{xy} = \delta[Mx = y]$; we will sometimes write $\delta_{xy} = \delta_{xy}(M)$ to emphasize that δ is a function of M.

Now symmetrize by averaging over all matrices with nullity d, giving

$$\begin{aligned} Q(d) &\coloneqq \mathop{\mathbb{E}}_{M \sim Y_d}[P(\delta(M))] \\ &= \sum_{S \subseteq \mathbb{F}_q^n \times \mathbb{F}_q^m} c_S \mathop{\mathbb{E}}_{M \sim Y_d}\left[\prod_{(x,y) \in S} \delta_{xy}(M)\right] \\ &= \sum_{S \subseteq \mathbb{F}_q^n \times \mathbb{F}_q^m} c_S \Pr_{M \sim Y_d}[Mx = y \ \forall (x,y) \in S], \end{aligned} \tag{27}$$

where Y_d is the set of matrices of nullity d. Here M is drawn uniformly from Y_d. Since $0 \le P(\delta(M)) \le 1$, we have $0 \le Q(d) \le 1$. The following lemma states that we can approximate $Q(d)$ with a low-degree polynomial. Van Apeldoorn and Gribling previously showed the same statement in their proof of a lower bound for Simon's problem for linear functions [4, Lemma 3]. That problem can be viewed as a special case of our problem with $m = n$. We observe that essentially the same proof establishes this lemma for $m \ge n$.

▶ **Lemma 19.** *There exists a polynomial R of degree at most $2t$ such that for each $d \in [n]$, $R(q^d) = Q(d)$.*

We emphasize that we do not bound the degree of $Q(d)$ because we do not know how to represent it as a polynomial in d. Instead, the lower bound is established by showing (i) a lower bound on the degree of the polynomial R and (ii) that the degree of R is no more than $2t$.

Next, recall a lemma by Koiran, Nesme, and Portier [16, Lemma 5].

▶ **Lemma 20.** *Let $c > 0$ and $\xi > 1$ be constants and let f be a real polynomial with the following properties:*

1. *for any integer $0 \le i \le n$, $|f(\xi^i)| \le 1$;*
2. *for some real number $1 \le x_0 \le \xi$, $|f'(x_0)| \ge c$.*

Then $\deg f = \Omega(n)$.

Lemma 19 and Lemma 20 imply an $\Omega(\min\{m,n\})$ lower bound for distinguishing a matrix is full-rank or has nullity 1. The case $m = n$ was previously shown by van Apeldoorn and Gribling [4, Theorem 1]. We briefly explain the main ideas for completeness. By Lemma 19, for $d \in \{0, 1, \ldots, n-1\}$, $R(q^d) = Q(d)$ and $\deg(R) \le 2t$. For distinguishing a full-rank matrix (i.e., $d = 0$) from a rank $n-1$ matrix (i.e, $d = 1$), we set $R(1) \ge 1 - \epsilon$ and $R(q) \le \epsilon$. There exists $x_0 \in [1, q]$ such that $R'(x_0) \ge \frac{|R(q) - R(1)|}{q-1} \ge \frac{1-2\epsilon}{q-1}$. By Lemma 20, $t = \Omega(n)$ for $m \ge n$. For $m < n$, an $\Omega(m)$ lower bound follows from Theorem 5. Overall, this gives the following.

▶ **Theorem 21.** *The bounded-error matrix-vector quantum query complexity of deciding if an $m \times n$ matrix over $\mathbb{F}_q$ is full-rank is $\Omega(\min\{m,n\})$. In particular, $\Omega(\min\{m,n\})$ queries are needed to decide whether the matrix is full-rank or has nullity 1.*

There is a trivial algorithm that learns an entire $m \times n$ matrix using $\min\{m, n\}$ queries. Thus the query complexity of computing the rank is $\Theta(\min\{m, n\})$.

▶ **Corollary 22.** *The bounded-error query matrix-vector quantum complexity of computing the rank of an $m \times n$ matrix over $\mathbb{F}_q$ is $\Theta(\min\{m, n\})$.*

With the same argument, the quantum query complexity of computing the determinant of an $n \times n$ matrix over $\mathbb{F}_q$ is $\Theta(n)$. Moreover, the classical query complexity is $\Theta(n^2)$, implied by the $\Omega(n^2)$ lower bound for rank testing by Rashtchian, Woodruff, and Zhu [19, Theorem 3.3].

▶ **Corollary 23** (Determinant)**.** *The bounded-error classical and quantum query complexities of computing the determinant of an $n \times n$ matrix over $\mathbb{F}_q$ through matrix-vector products are $\Theta(n^2)$ and $\Theta(n)$, respectively.*

4.3 Solving linear systems

In this section, we consider the quantum query complexity of solving the linear system $Ax = b$ for $A \in \mathbb{F}_q^{n\times n}$ is $\Theta(n)$. Since there is an n-query algorithm learning the entire matrix using n matrix-vector queries, we focus on the lower bound.

Our proof is based on a randomized reduction from deciding whether a submatrix is full rank. For a square matrix A, let A^{ij} be the submatrix obtained by deleting the i^{th} row and the j^{th} column, and let A_{ij} denote the (i, j) element of A. The elements of A^{-1} can be computed as

$$(A^{-1})_{ij} = \frac{\det A^{ij}}{\det A}. \tag{28}$$

Given an invertible A, one can use a linear system solver to decide whether $(A^{-1})_{11}$ is non-zero, and thus decide if the minor A^{11} is full-rank.

In our reduction, to decide whether $M \in \mathbb{F}_q^{n\times n}$ is full-rank given access to matrix-vector products, we pad M with one extra random row and one extra random column, giving a matrix $A \in \mathbb{F}_q^{(n+1)\times(n+1)}$. We show that with sufficiently high probability, the padded matrix is full-rank. Thus, invoking a linear system solver with $b = e_1$, we learn whether $\det M = 0$. Thus the linear regression lower bound follows from Theorem 21.

▶ **Theorem 24.** *The bounded-error matrix-vector quantum query complexity of solving an $n \times n$ linear system is $\Omega(n)$.*

Proof. Assume toward contradiction that $\mathcal{A}$ is a t-query quantum algorithm for determining whether $(A^{-1})_{11}$ is non-zero for any invertible $A \in \mathbb{F}_q^{(n+1)\times(n+1)}$, succeeding with probability $p \geq 1/3$ with $t = o(n)$. We present a t-query algorithm for determining whether an $n \times n$ matrix is full-rank with probability $p(1 - 1/q)^2 \geq 1/12$.

Given access to $M \in \mathbb{F}_q^{n\times n}$, the algorithm first samples two random vectors $u, v \in \mathbb{F}_q^n$ and a random element $a \in \mathbb{F}_q$ to give the padded matrix

$$A = \begin{bmatrix} a & u^\top \\ v & M \end{bmatrix}. \tag{29}$$

The matrix-vector product $A(x_0, x^\top)^\top$ for $x_0 \in \mathbb{F}_q, x \in \mathbb{F}_q^n$ can be computed using one Mv query to Mx since

$$A \begin{bmatrix} x_0 \\ x \end{bmatrix} = \begin{bmatrix} a_0 + u^\top x \\ x_0 v + Mx \end{bmatrix}. \tag{30}$$

We show that with probability at least $(1-1/q)^2$, the matrix A is invertible (i.e., $\det A \neq 0$) given that $\operatorname{rank}(M) \geq n-1$. If M is invertible, the submatrix $B = (v, M)$ is full-rank. If $\operatorname{rank}(M) = n-1$, then without loss of generality, we consider the case that the first $n-1$ rows of M are linearly independent, and the last row is a linear combination of the first $n-1$ rows, since other cases can be handled accordingly by rearranging the rows. We let

$$M = \begin{bmatrix} M' \\ w^\top \end{bmatrix}. \tag{31}$$

for an $(n-1) \times n$ matrix M' and an $n \times 1$ vector w. Since $w^\top$ is a linear combination of the first $n-1$ rows, we write $w^\top = c^\top M'$ for an $(n-1) \times 1$ vector c. Since M' is full-rank, the vector c satisfying $w^\top = c^\top M'$ is unique. Now write the vector

$$v = \begin{bmatrix} z \\ b \end{bmatrix} \tag{32}$$

for an $(n-1) \times 1$ matrix z and $b \in \mathbb{F}_q$. The matrix B is not full rank if and only if the last row is a linear combination of the first $n-1$ rows, i.e., $c^\top z = b$, since the first $n-1$ rows of B are linearly independent. Since v is a random vector with each element chosen independently, we have

$$\Pr[B \text{ is not full-rank}] = \Pr_{z,b}[c^\top z = b] = 1/q. \tag{33}$$

Thus with probability at least $1 - 1/q$ the matrix B is full-rank.

Conditioned on B being full-rank, the matrix A is not full-rank if and only if the vector $(a, u^\top)$ is in the vector space spanned by the rows of B. The number of vectors in the vector space is $q^{(n-1)}$. Thus

$$\Pr_{a,u,v}[A \text{ is not full-rank} \mid B \text{ is full-rank}] = 1/q. \tag{34}$$

Therefore with probability at least $1 - 1/q$, A is invertible. Conditioned on successfully simulating Mv queries of an invertible A, the algorithm $\mathcal{A}$ determines whether $(A^{-1})_{11}$ is nonzero with probability p. Thus the algorithm succeeds with probability at least $p(1-1/q)^2 \geq 1/12$ using $t = o(n)$ queries to M. By Theorem 21 we have a contradiction. ◀

The same proof idea shows that a lower bound for rank testing implies a lower bound for linear regression in the vMv model. Rashtchian, Woodruff, and Zhu show that the query complexity of distinguishing rank-n matrices from rank-$(n-1)$ matrices over $\mathbb{F}_q$ is $\Omega(n^2)$ [19, Theorem 3.3].

▶ **Corollary 25.** *The bounded-error classical* vMv *query complexity of solving an $n \times n$ linear system over $\mathbb{F}_q$ is $\Omega(n^2)$.*

Proof. By the same idea as in the proof of Theorem 24, it suffices to show that one vMv query to the $(n+1) \times (n+1)$ matrix A in (29) can be simulated with one vMv query to the $n \times n$ matrix M. For any query x, y, we let $x = (x_0, x_1^\top)^\top$ and $y = (y_0, y_1^\top)^\top$ for $n \times 1$ matrices x_1, y_1. The product $y^\top A x$ can be computed using one vMv query to M since $y^\top A x = a y_0 x_0 + y_0 u^\top x_1 + y_1^\top v x_0 + y_1^\top M x_1$. Since no $o(n^2)$-query classical algorithm can distinguish rank-n matrices from rank-$(n-1)$ matrices [19, Theorem 3.3], the bounded-error query complexity of solving linear systems is $\Omega(n^2)$. ◀

4.4 Rank testing

In this section, we show a linear lower bound on distinguishing whether an $m \times n$ matrix M has $\operatorname{rank}(M) = n$ or $\operatorname{rank}(M) \le n/2$, where $m \ge n$. First we show the following lemma using ideas from [16].

▶ **Lemma 26.** *Let $\xi \ge 2$ and let n be an even integer. Then any polynomial f satisfying*

1. $0 \le f(\xi^i) \le 1$ *for* $i \in \{0, 1, \dots, n-1\}$ *and*

2. $f(1) \le 1/3$ *and* $f(\xi^i) \ge 2/3$ *for* $i \in \{n/2, n/2+1, \dots, n-1\}$

has $\deg(f) = \Omega(n)$*.*

Proof. Let $d = \deg(f)$. Toward contradiction, we assume $d = o(n)$. For intervals $S_i := [\xi^i, \xi^{i+1})$, since $\deg(f'), \deg(f'') = o(n)$, there exists an index $a \in \{9n/10, \dots, n-3, n-2\}$ such that none of the roots of f' and f'' has its real part in S_a. This implies that f' is monotonically increasing or decreasing in S_a, i.e., f is concave or convex. In each case, $f(\frac{\xi^a+\xi^{a+1}}{2}) \in [0,1]$. If f is convex in S_a,

$$\left| f'\left(\frac{\xi^a + \xi^{a+1}}{2}\right) \right| \le \frac{1}{\xi^{a+1} - \frac{\xi^{a+1}+\xi^a}{2}} = \frac{2}{\xi^{a+1} - \xi^a} \le \frac{2}{\xi^a} \le 2\xi^{-9n/10}. \tag{35}$$

If f is concave in S_a, reflecting about the x-axis gives the same bound.

By the second constraint, there exists $x_0 \in [1, \xi^{n/2}]$ such that

$$|f'(x_0)| \ge \frac{|f(\xi^{n/2}) - f(1)|}{\xi^{n/2} - 1} \ge \xi^{-n/2}/3. \tag{36}$$

Therefore

$$\left| \frac{f'(\frac{\xi^a+\xi^{a+1}}{2})}{f'(x_0)} \right| \le 6\xi^{-2n/5} \le \xi^{3-2n/5}. \tag{37}$$

On the other hand, since $\deg(f') = d-1$, denoting the roots $a_1, \dots, a_{d-1} \in \mathbb{C}$, we write

$$f'(x) = \lambda \prod_{i=1}^{d-1} (x - a_i). \tag{38}$$

Thus

$$\left| \frac{f'(\frac{\xi^a+\xi^{a+1}}{2})}{f'(x_0)} \right| = \prod_{i=1}^{d-1} \left| \frac{\frac{\xi^a+\xi^{a+1}}{2} - a_i}{x_0 - a_i} \right| = \prod_{i=1}^{d-1} |g(a_i)|, \tag{39}$$

where

$$g(x) = \frac{x - \frac{\xi^a+\xi^{a+1}}{2}}{x - x_0}. \tag{40}$$

Our goal is to show that for each i, $|g(a_i)| \ge \frac{1}{2\xi}$. Recall that for each i, $\Re(a_i) \notin S_a$. Also for real $x \notin S_a$, $x \ge x_0$, we have $|g(x)| \ge \frac{\xi-1}{2\xi} \ge \frac{1}{2\xi}$. For real roots, $|g(a_i)| \ge \frac{1}{2\xi}$. Now we consider the case where $a_i = \alpha + \beta i$ for $\beta \neq 0$, giving

$$|g(\alpha + \beta i)|^2 = \frac{(\alpha - \frac{\xi^a+\xi^{a+1}}{2})^2 + \beta^2}{(\alpha - x_0)^2 + \beta^2}. \tag{41}$$

If $(\alpha - \frac{\xi^a+\xi^{a+1}}{2})^2 \geq (\alpha - x_0)^2$, then $|g(\alpha+\beta i)| \geq 1$. Otherwise,

$$|g(\alpha+\beta i)| \geq \left| \frac{\alpha - \frac{\xi^a+\xi^{a+1}}{2}}{\alpha - x_0} \right| \geq \frac{1}{2\xi}. \tag{42}$$

We have shown that $|g(a_i)| \geq \frac{1}{2\xi}$ for every root a_i. Now we have

$$\left| \frac{f'(\frac{\xi^a+\xi^{a+1}}{2})}{f'(x_0)} \right| = \prod_{i=1}^{d-1} |g(a_i)| \geq (2\xi)^{-d+1} \geq \xi^{2-2d}. \tag{43}$$

Thus by (37), we have $\xi^{3-2n/5} \geq \xi^{2-2d}$ and conclude $d \geq n/5 - 1/2 = \Omega(n)$ – a contradiction. ◀

Lemma 19 and Lemma 26 imply the following theorem.

▶ **Theorem 27.** *The bounded-error matrix-vector quantum query complexity of determining whether a matrix $M \in \mathbb{F}_q^{m\times n}$ has $\operatorname{rank}(M) = n$ or $\operatorname{rank}(M) \leq n/2$ is $\Omega(n)$.*

References

1 Scott Aaronson. Quantum lower bound for the collision problem. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 635–642, 2002. arXiv:quant-ph/0111102.

2 Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291, 2015.

3 Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, 2020. arXiv:1809.00643.

4 Joran van Apeldoorn and Sander Gribling. Simon's problem for linear functions, 2018. arXiv:1810.12030.

5 Simon Apers and Ronald de Wolf. Quantum speedup for graph sparsification, cut approximation and laplacian solving. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science*, pages 637–648. IEEE, 2020. arXiv:1911.07306.

6 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. arXiv:quant-ph/9802049.

7 Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

8 Mark Braverman, Elad Hazan, Max Simchowitz, and Blake Woodworth. The gradient complexity of linear regression. In *Conference on Learning Theory*, pages 627–647, 2020. arXiv:1911.02212.

9 Andries E. Brouwer, Arjeh M. Cohen, and Arnold Neumaier. *Distance-Regular Graphs*. Springer-Verlag, 1989.

10 Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, 2020. arXiv:1809.01731.

11 Andrew M. Childs. Equation solving by simulation. *Nature Physics*, 5:861, 2009. doi:10.1038/nphys1473.

12 Daniel Copeland and Jamie Pommersheim. Quantum query complexity of symmetric oracle problems. *Quantum*, 5:403, 2021. arXiv:1812.09428.

13 Ankit Garg, Robin Kothari, Praneeth Netrapalli, and Suhail Sherif. No quantum speedup over gradient descent for non-smooth convex optimization. In *12th Innovations in Theoretical Computer Science Conference (to appear)*, 2021. arXiv:2010.01801.

14 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv:0811.3171.

15 Stephen P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical Review Letters*, 95(5):050501, 2005. `arXiv:quant-ph/0405146`.

16 Pascal Koiran, Vincent Nesme, and Natacha Portier. The quantum query complexity of the abelian hidden subgroup problem. *Theoretical Computer Science*, 380(1-2):115–126, 2007.

17 Troy Lee, Miklos Santha, and Shengyu Zhang. Quantum algorithms for graph problems with cut queries. In *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms*, pages 939–958. SIAM, 2021. `arXiv:2007.08285`.

18 Ashley Montanaro and Changpeng Shao. Quantum algorithms for learning graphs and beyond, 2020. `arXiv:2011.08611`.

19 Cyrus Rashtchian, David P. Woodruff, and Hanlin Zhu. Vector-matrix-vector queries for solving linear algebra, statistics, and graph problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. `arXiv:2006.14015`.

20 Jean-Pierre Serre. *Linear Representations of Finite Groups*, volume 42 of *Graduate Texts in Mathematics*. Springer, 1977.

21 Xiaoming Sun, David P. Woodruff, Guang Yang, and Jialin Zhang. Querying a matrix through matrix-vector products. In *46th International Colloquium on Automata, Languages, and Programming*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019. `arXiv:1906.05736`.

22 David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014. `arXiv:1411.4357`.