The Quantum Query Complexity of the Determinant

Sebastian Dörn
Inst. für Theoretische Informatik
Universität Ulm
89069 Ulm, Germany

Thomas Thierauf
Fak. Elektronik und Informatik
HTW Aalen
73430 Aalen, Germany

{sebastian.doern,thomas.thierauf}@uni-ulm.de

Abstract

In this paper we give tight quantum query complexity bounds of some important linear algebra problems. We prove $\Theta(n^2)$ quantum query bounds for verify the determinant, rank, matrix inverse and the matrix power problem.

1 Introduction

The computation of the quantum query complexity for special problems is a fast growing area in quantum computing. For example, quantum query algorithms have been presented for several problems from computer science (see e.g. [BHT98, BDHHMSW01, Amb04]), graph theory (see e.g. [DHHM04, BDFLS04, MSS05, AS06, Doe07a, Doe07b]) and algebra (see e.g. [MN05, BS06, DT07, DT08]). In most cases the quantum query complexity is better than the classical query complexity. Therefore quantum algorithms have the potential to demonstrate that for some problems, quantum computation is more efficient than classical computation.

In our paper we show that for some important problems from linear algebra quantum computing is not better than classical computation. We prove $\Omega(n^2)$ lower bounds of the quantum query complexity for the verification of the determinant, rank, matrix inverse and the matrix power problem. Since $O(n^2)$ is a trivial upper bound for the (quantum) query complexity of these problems, our bounds are tight.

There are several reasons for studying the query complexity of such linear algebra problems. On the one hand side, these are fundamental and basic problems which have many applications in computer science. On the other hand, our work shows that there are several problems for which quantum computation is not faster than classical one. Furthermore there is some hope that our reduction can be used to improve the quantum query lower bound for other problems. For example, from the $\Omega(n^2)$ quantum query lower bound for the determinant it is tempting to conjecture the same bound for the perfect matching problem. However, the known quantum lower bound for the perfect matching problem is only $\Omega(n^{1.5})$ (see [Zha04]).

In Sections 3, we consider the matrix power problem. Given two $n \times n$ integer matrices A and B and an integer m, we have to decide whether the m'th power of A is equal the matrix B. In Section 4 we look at the decision problem, whether the inverse of a matrix A is equal a matrix B. Furthermore we are interested, whether a given matrix is singular.

2 Preliminaries

2.1 Quantum Query Model

In the query model, the input $x_1, \ldots, x_N$ is contained in a black box or oracle and can be accessed by queries to the black box. As a query we give i as input to the black box and the black box outputs x_i. The goal is to compute a Boolean function $f : \{0,1\}^N \rightarrow \{0,1\}$ on the input bits $x = (x_1, \ldots, x_N)$ minimizing the number of queries. The classical version of this model is known as decision tree.

The quantum query model was explicitly introduced by Beals et al. [BBCMW01]. In this model we pay for accessing the oracle, but unlike the classical case, we use the power of quantum parallelism to make queries in superposition. The state of the computation is represented by $|i, b, z\rangle$, where i is the query register, b is the answer register, and z is the working register.

A quantum computation with T queries is a sequence of unitary transformations

$$U_0 \rightarrow O_x \rightarrow U_1 \rightarrow O_x \rightarrow \ldots \rightarrow U_{T-1} \rightarrow O_x \rightarrow U_T,$$

where each U_j is a unitary transformation that does not depend on the input x, and O_x are query (oracle) transformations. The oracle transformation O_x can be defined as $O_x : |i, b, z\rangle \rightarrow |i, b \oplus x_i, z\rangle$.

The computation consists of the following three steps:

1. Go into the initial state $|0\rangle$.

2. Apply the transformation $U_T O_x \cdots O_x U_0$.

3. Measure the final state.

The result of the computation is the rightmost bit of the state obtained by the measurement.

The quantum computation determines f with bounded error, if for every x, the probability that the result of the computation equals $f(x_1, \ldots, x_N)$ is at least $1 - \epsilon$, for some fixed $\epsilon < 1/2$. In the query model of computation each query adds one to the query complexity of an algorithm, but all other computations are free. The time complexity of the algorithm is usually measured in terms of the total circuit size for the unitary operations U_i. All quantum algorithms in this paper are bounded error.

The quantum query complexity of black box computation has become a great interest in quantum computing. The black box model provides a simple and abstract framework for the construction of quantum algorithms. All quantum algorithms can be formulated in the black box model, we can determine the speed up against classical algorithm, and we can prove lower bounds for the quantum query complexity.

2.2 A tool for quantum query lower bounds

We use the following special case of the adversary method of Ambainis [Amb02] to prove lower bounds for the quantum query complexity.

Theorem 2.1 [Amb02] *Let $D \subseteq \{0,1\}^n$ a decision problem. Let furthermore $A, B \subseteq \{0,1\}^n$ such that $A \subseteq D$ and $B \subseteq \overline{D}$.*

Let m and m' be numbers such that

1. *for every $(x_1, \ldots, x_n) \in A$ there are $\geq m$ values $i \in \{1, \ldots, n\}$ such that $(x_1, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots, x_n) \in B$,*

2. *for every $(x_1, \ldots, x_n) \in B$ there are $\geq m'$ values $i \in \{1, \ldots, n\}$ such that $(x_1, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots, x_n) \in A$.*

Then every bounded-error quantum algorithm that decides D has quantum query complexity $\Omega(\sqrt{m \cdot m'})$.

3 Matrix Power

In this section we determine the quantum query complexity of the *matrix power* resp. *matrix power element problem.* In the matrix power problem we have given two $n \times n$ matrices A and B and an integer m. Decide whether $A^m = B$. In the matrix power element problem we have given a $n \times n$ matrix A and integers i, j, a, m. Decide if $(A^m)_{i,j} = a$. We show tight bounds on the quantum query complexity of these problems, namely $\Theta(n^2)$. For this task, we define the following problem for n variables $x_1, \dots, x_n \in \{0, 1\}$ and $0 \leq a \leq n$,

$$\textsc{Sum}_n = \{\, (x_1, \dots, x_n, a) \mid \sum_{i=1}^{n} x_i = a \,\}.$$

Lemma 3.1 *The quantum query complexity of* $\textsc{Sum}_n$ *is* $\Theta(n)$*.*

Proof. We apply Theorem 2.1 to the restriction of $\textsc{Sum}_n$ to $a = \lfloor n/2 \rfloor$. That is, we consider $D = \{\, (x_1, \dots, x_n) \in \{0, 1\}^n \mid \sum_{i=1}^{n} x_i = \lfloor n/2 \rfloor \,\}$.

Define $A = D$ and $B = \{\, (x_1, \dots, x_n) \in \{0, 1\}^n \mid \sum_{i=1}^{n} x_i = \lfloor n/2 \rfloor + 1 \,\}$. For any sequence $x \in A$, if we change any of the $m = \lfloor n/2 \rfloor$ zeros of x to one, the resulting sequence will be in B. Conversely, for any sequence $x' \in B$, if we change any of the $m' = \lfloor n/2 \rfloor + 1$ ones of x' to zero, the resulting sequence will be in A. Therefore the quantum query complexity of D, and hence of $\textsc{Sum}_n$, is $\Omega(\sqrt{m \cdot m'}) = \Omega(n)$. □

We show in the following how to reduce $\textsc{Sum}_{n^2}$ to matrix power. In fact, the reduction maps to matrix power of 3. The lower bounds for the determinant and the matrix inverse that we show in the next section crucially depend on the hardness of matrix powers already for constant powers.

Theorem 3.2 *The quantum query complexity of the matrix power and the matrix power element problem is* $\Theta(n^2)$*. This already holds for powers of* 3*.*

Proof. Given $x_1, \dots, x_{n^2} \in \{0, 1\}$ and $0 \leq a \leq n^2$ as input for $\textsc{Sum}_{n^2}$, we construct a directed graph G as follows. G has $2n + 2$ nodes. With nodes $1, \dots, 2n$ we construct a bipartite graph with nodes $1, \dots, n$ on the left side and nodes $n+1, \dots, 2n$ on the right side. For the edges, we consider the variables x_k. Index k can be uniquely written as $k = (i - 1)n + j$, for $1 \leq i, j \leq n$. Edge $(i, n + j)$ is present in G iff $x_k = 1$. For the remaining two nodes, let $s = 2n + 1$ and $t = 2n + 2$. Add edges from s to all the nodes $1, \dots, n$ and edges from all nodes $n + 1, \dots, 2n$ to t. This completes the construction of graph G.

Observe that all paths from s to t in G have length 3 and each such path uniquely corresponds to a variable x_k with value 1. Moreover, there are no further paths of length 3 in G. Let A be the adjacency matrix of G. We conclude that the entry (s, t) of A^3 is the number of paths from s to t

in G, and all other entries are 0. Define matrix B with (s,t) entry a and 0 elsewhere. Then we have

$$\textsc{Sum}_{n^2}(x_1,\ldots,x_{n^2},a)=1 \Longleftrightarrow (A^3)_{s,t}=a \Longleftrightarrow A^3=B.$$

□

4 Determinant and Inverse

We consider the *determinant* and *inverse problem*. In the determinant problem we have given a $n \times n$ matrix A, decide whether $\det(A)=0$. In the inverse problem we have given a regular $n \times n$ matrix A and integers i,j,a. One has to decide whether $(A^{-1})_{i,j}=a$.

Theorem 4.1 *The quantum query complexity of the determinant and the inverse problem is* $\Theta(n^2)$.

Proof. We slightly modify the standard reduction from matrix power element to the determinant. Let $A=(a_{i,j})$ be a $n \times n$ matrix and a be given. By Theorem 3.2, we may assume that A is a 0-1-matrix and we consider the problem whether $(A^3)_{1,n}=a$. We will construct a matrix B such that $\left(A^3\right)_{1,n}=\det(B)$.

Interpret A as representing a directed bipartite graph on $2n$ nodes. That is, the nodes are arranged in two columns of n nodes each. In both columns, nodes are numbered from 1 to n. If $a_{i,j}=1$ then we put an edge from node i in the first column to node j in the second column.

Now, take 3 copies of this graph, put them in a sequence and identify each second column of nodes with the first column of the next graph in the sequence. We have 4 columns of n nodes each so far. Now we add a 5-th column of n nodes as well and connect it by horizontal edges with the 4-th column.

Call the resulting graph G'. Graph G' has $N=5n$ nodes, and the entry at position $(1,n)$ in A^3 is the number of paths in G' from node 1 in the first column to node n in the last column. Call these two nodes s and t, respectively.

Next, we add an edge from t to s and put self-loops at all nodes except s and t. Call the resulting graph G and let B be the adjacency matrix of G. From combinatorial matrix theory we know that the determinant of B is the signed sum of cycle covers of G. Any cycle cover of G consists of one cycle of length 5 which goes from s to t via the columns in G' and then back to s.

The remaining cycles of the cover are self-loops. Therefore each cycle cover corresponds to one path from s to t in G'. The sign of the cycle cover is $(-1)^{N+k}$, where k is the number of cycles in the cover. We have $k = N - 4$. Therefore the sign is 1. We conclude that $\det(B) = \left(A^3\right)_{1,n}$.

Note that the size of B is linear in the size of A. Therefore the lower bound for matrix powering carries over to the determinant.

In order to get a reduction to the matrix inverse problem, we modify graph G from above and add self-loops to nodes s and t. Let H be the resulting graph and $C = (c_{i,j})$ be the adjacency matrix of H. The identity permutation is an additional cycle cover of H compared to G. Therefore we have $\det(C) = \left(A^3\right)_{1,n} + 1$. If C has no inverse, then we have $\det(C) = 0$ and consequently $\left(A^3\right)_{1,n} = -1$. In the following, assume that C has an inverse.

Note that $c_{i,i} = 1$ since all nodes have self-loops. Furthermore, with the convention $s = 1$ and $t = N$ we have $c_{N,1} = 1$ because of the edge from t to s. Note that except for position $(N, 1)$ matrix C is an upper triangular matrix. We consider the Laplace expansion of $\det(C)$. Let $C_{i,j}$ denote the matrix obtained from C by deleting row i and column j. We consider the expansion for the first column:

$$\begin{aligned}\det(C) &= c_{1,1}\det(C_{1,1}) + (-1)^{N+1}\det(C_{N,1})\\ &= 1 + (-1)^{N+1}\det(C_{N,1}),\end{aligned}$$

because $\det(C_{1,1}) = 1$. Let c be the entry at position $(N, 1)$ in C^{-1}. By Cramer's rule we have $c = \det(C_{N,1})/\det(C)$. Hence we get

$$\det(C) = 1 + (-1)^{N+1}c \cdot \det(C).$$

We replace $\det(C)$ by $\left(A^3\right)_{1,n} + 1$ and obtain

$$\left(A^3\right)_{1,n}\left((-1)^{N+1} - c\right) = c.$$

It follows that $((-1)^{N+1} - c)$ is non-zero and we have

$$\left(A^3\right)_{1,n} = \frac{c}{(-1)^{N+1} - c}.$$

The size of C is linear in the size of A. Therefore the lower bound for matrix powering carries over to the inverse. □

In the *rank problem* we have given an $n \times n$ matrix A and integer k. One has to decide whether $\operatorname{rank}(A) = k$. For $k = n$ we have $\operatorname{rank}(A) = n \iff \det(A) \neq 0$. Therefore the quantum query complexity of the rank problem follows from the determinant problem.

Corollary 4.2 *The quantum query complexity of the rank problem is* $\Theta(n^2)$.

Open problems

The current quantum query lower bound for the perfect matching problem is $\Omega(n^{1.5})$ (see [Zha04]). Because the determinant is related to the perfect matching problem, we conjecture that the $\Omega(n^2)$ lower bound holds there as well.

Acknowledgments

We thank Thanh Minh Hoang for helpful discussions and the referees of the paper for valuable hints that simplified some computations.

References

[Amb02] A. Ambainis, *Quantum Lower Bounds by Quantum Arguments*, Journal of Computer and System Sciences 64: pages 750-767, 2002.

[Amb04] A. Ambainis, *Quantum walk algorithm for element distinctness*, Proceedings of FOCS'04: pages 22-31, 2004.

[AS06] A. Ambainis, R. Špalek, *Quantum Algorithms for Matching and Network Flows*, Proceedings of STACS'06: pages 172-183, 2006.

[BBCMW01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf, *Quantum lower bounds by polynomials*, Journal of ACM 48: pages 778-797, 2001.

[BDFLS04] A. Berzina, A. Dubrovsky, R. Freivalds, L. Lace, O. Scegulnaja, *Quantum Query Complexity for Some Graph Problems*, Proceedings of SOFSEM'04: pages 140-150, 2004.

[BDHHMSW01] H. Buhrman, C. Dürr, M Heiligman, P. Høyer, F. Magniez, M. Santha, R. de Wolf, *Quantum Algorithms for Element Distinctness*, Proceedings of CCC'01: pages 131-137, 2001.

[BHT98] G. Brassard, P. Høyer, A. Tapp, *Quantum cryptanalysis of hash and claw-free functions*, Proceedings of LATIN'98: pages 163-169, 1998.

[BS06] H. Buhrman, R. Špalek, *Quantum Verification of Matrix Products*, Proceedings of SODA'06: pages 880-889, 2006.

[DHHM04] C. Dürr, M. Heiligman, P. Høyer, M. Mhalla, *Quantum query complexity of some graph problems*, Proceedings of ICALP'04: pages 481-493, 2004.

[Doe07a] S. Dörn, *Quantum Complexity Bounds of Independent Set Problems*, Proceedings of SOFSEM'07 (SRF): pages 25-36, 2007.

[Doe07b] S. Dörn, *Quantum Algorithms for Graph Traversals and Related Problems*, Proceedings of CIE'07: pages 123-131, 2007.

[DT07] S. Dörn, T. Thierauf, *The Quantum Query Complexity of Algebraic Properties*, Proceedings of FCT'07: pages 250-260, 2007.

[DT08] S. Dörn, T. Thierauf, *The Quantum Complexity of Group Testing*, Proceedings of SOFSEM'08: pages 506-518, 2008.

[Gro96] L. Grover, *A fast mechanical algorithm for database search*, Proceedings of STOC'96: pages 212-219, 1996.

[MN05] F. Magniez, A. Nayak, *Quantum complexity of testing group commutativity*, Proceedings of ICALP'05: pages 1312-1324, 2005.

[MSS05] F. Magniez, M. Santha, M. Szegedy, *Quantum Algorithms for the Triangle Problem*, Proceedings of SODA'05: pages 1109-1117, 2005.

[Zha04] S. Zhang, *On the power of Ambainis's lower bounds*, Proceedings of ICALP'04, Lecture Notes in Computer Science 3142: pages 1238-1250, 2004.