Quantum Information Processing (2019) 18:99
https://doi.org/10.1007/s11128-019-2208-7

Fourier 1-norm and quantum speed-up

Sebastián Alberto Grillo[1] · Franklin de Lima Marquezino[2]

Received: 27 October 2018 / Accepted: 4 February 2019 / Published online: 20 February 2019

Abstract
Understanding quantum speed-up over classical computing is fundamental for the development of efficient quantum algorithms. In this paper, we study such problem within the framework of the quantum query model, which represents the probability of output $x \in \{0, 1\}^n$ as a function $\pi(x)$. We present a classical simulation for output probabilities π, whose error depends on the Fourier 1-norm of π. Such dependence implies upper-bounds for the quotient between the number of queries applied by an optimal classical algorithm and our quantum algorithm, respectively. These upper-bounds show a strong relation between Fourier 1-norm and quantum parallelism. We show applications to query complexity.

Keywords Quantum query · Randomized query · Simulation

1 Introduction

A primary motivation in quantum computing is obtaining algorithms that solve problems much faster than the best classical counterparts. The quantum and classical decision tree models allow us to prove the existence of quantum speed-up in relation to classical query for several problems [6,17,27]. Query problems can be formulated as computing Boolean functions from inputs in $\{0, 1\}^n$, with complexity being defined as the number of queries to the input, ignoring other computations [12]. This implies an important simplification of the analysis in comparison with problems formulated by Turing machines, where separations between complexity classes are usually much harder to prove [2]. Several quantum algorithms can be formulated within query

✉ Franklin de Lima Marquezino
franklin@cos.ufrj.br

Sebastián Alberto Grillo
sgrillo@uaa.edu.py

[1] Universidad Autónoma de Asunción, Jejuí 667, Asunción, Brazil

[2] Universidade Federal do Rio de Janeiro, Avenida Horacio Macedo, 2030, COPPE/H319, Rio de Janeiro, Brazil

models,[1] thus this formalism is powerful enough for analyzing important algorithms, such as search algorithms [1] or even non-query algorithms as Shor's algorithm [3].

A complete understanding of quantum speed-up implies determining where and how it occurs. Thus, we can study such question from two distinct approaches: determining which *functions* or which *algorithms* allow a gap between quantum and classical computing. The first approach is intensively used in quantum query complexity, where effort is mainly invested in obtaining bounds for complexity measures and checking their tightness [1,12]. The second approach is commonly implemented by identifying which quantum features are hard to simulate within classical sources [5]. One of the earliest attempts to explain quantum advantage is the discussion of quantum parallelism in quantum algorithms [16].

A well-studied quantum feature is quantum entanglement [20], which has been identified as a necessary condition for quantum speed-up in pure-state algorithms [21]. At the same time, the study of quantum entanglement depends on whether pure or mixed quantum states are allowed [14] and the measure defined for such entanglement [21,29]. As an example of a widely applied entanglement measure, we can consider the size of partitions that describe product states in the quantum algorithm. If the size of the subsets in those partitions is upper-bounded by a constant through all the steps of the quantum algorithm, then it has an efficient classical simulation [21]. In addition, we can analyze the entanglement in a quantum state by measuring the Schmidt rank, where a polynomial upper-bound for this measure implies a polynomial classical simulation [29]. Using a model previously defined by Knill and Laflamme [22], different conditions for quantum speed-up were also identified. Such conditions are formulated on quantum correlations that are analyzed by a measure known as *quantum discord* [13]. A recent proposal comes from *no-go* theorems, identifying contextuality [11,23] as a necessary condition for quantum speed-up—this condition presents an inequality violation in contrast to the other conditions based on measures [19]. The identification of necessary conditions for quantum advantages is an important issue for theoretical purposes and for the design of better quantum algorithms, specially if the conditions can be monitored in our design. Summarizing, a general goal in this line of research is to obtain sufficient and necessary conditions for quantum speed-up.

The present work offers a new perspective about speed-up produced by quantum algorithms in the quantum query model (QQM), which is the quantum generalization for decision tree models. First, we consider that the probability of obtaining a given output is a linear combination of orthonormal functions, where such set of functions is denoted as *Fourier basis* [15]. This approach is usually referred to as *analysis of Boolean functions* and has several results in quantum query and computer science [24–26,28]. Using such representation of the output probability, we define a classical simulation of the quantum algorithm. The idea of our simulation is implementing minor simulations for parity functions from the Fourier basis, where each simulated function appears in the Fourier decomposition of the output probability. Similarly to related works in the context of quantum entanglement or quantum discord, in this paper we follow a strategy known as *dequantization* [5], which consists in analyzing

[1] For details and up-to-date examples, the reader may refer to S. Jordan, *Quantum Algorithm Zoo*, https://math.nist.gov/quantum/zoo/.

how hard is the simulation of some algorithm in relation to a given measure. We prove that the error in our simulation depends on the L_1 norm defined over the Fourier basis, where such norm is computed for the output probability. Thereby, a necessary property for a hard classical simulation of a given quantum algorithm is having a large Fourier 1-norm for its output probability functions. This necessary condition is formalized as an upper-bound for the quotient between the number of queries of an optimal classical algorithm and of the simulated quantum algorithm, respectively. Notice that a well-designed algorithm in the QQM setting should maximize such quotient.

The state of any algorithm in the QQM can be described as a sum of vectors whose phase change depends on the input. The phase of each of these vectors may depend on different values from input, which shows quantum parallelism in action [18]. We show that the minimum size and the number of such vectors limit the value of the Fourier 1-norm, which allows alternative necessary conditions for quantum speed-up. The Fourier 1-norm is maximized by the homogeneity on the size of the vectors, which implies that simulating such balanced probabilities can be expensive by classical means. Therefore, our results give more formalism to the notion of quantum parallelism. Finally, we show applications of our results on (1) upper-bounds for randomized query, (2) lower-bounds for exact quantum query, and (3) polynomial simulation by randomized query.

We interpret the Fourier 1-norm as a measure for the quantity of Walsh functions that are "compressed" into the output probability function. The Walsh functions can be seen as individual computations in quantum query parallelism. Our intuition is that Fourier 1-norm is the amount of quantum parallelism on the output, and that quantum parallelism has potentially a quadratic influence on the relation between classical and quantum query complexities. Another reason for preferring the Fourier 1-norm over other measures is the possibility of controlling this parameter on a family of bound degree polynomials. We believe this can be an alternative strategy for the design of quantum algorithms with speed-up over the classical counterparts.

This work is structured as follows. In Sect. 2, we introduce preliminary formulations and theorems about the QQM. In Sect. 3, we describe a classical simulation of quantum algorithms. In Sect. 4, we present the upper-bounds from our simulation. In Sect. 5, we present alternative applications of our results. In Sect. 6, we present our conclusion.

2 Preliminary notions

The QQM [9] describes algorithms computing functions whose domain is a subset of $\{0, 1\}^n$. We describe the states and operations within such model, over a Hilbert space $\mathcal{H}$ with basis states $|i\rangle\,|j\rangle$, where $i \in \{0, 1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$, for an arbitrary m. The query operator is defined as $O_x\,|i\rangle\,|j\rangle = (-1)^{x_i}\,|i\rangle\,|j\rangle$, where $x \equiv x_0x_1\cdots x_n$ is the input, and $x_0 \equiv 0$. The final state of the algorithm over input x is defined as $\left|\Psi_x^f\right\rangle = U_tO_xU_{t-1}\cdots O_xU_0\,|\Psi\rangle$, where $\{U_i\}$ is a set of unitary operators over $\mathcal{H}$ and $|\Psi\rangle$ is a fixed state in $\mathcal{H}$. The number of queries or steps is defined as the times that O_x occurs in the algorithm.

Definition 1 An indexed set of pairwise orthogonal projectors $\{P_z : z \in T\}$ is called a complete set of orthogonal projectors (CSOP) if it satisfies

$$\sum_{z\in T} P_z = I_{\mathcal{H}}, \tag{1}$$

taking $I_{\mathcal{H}}$ as the identity operator for $\mathcal{H}$.

Given a CSOP defined for the algorithm, the probability of obtaining the output $z \in T$ is $\pi_z(x) = \left\| P_z \left| \Psi_x^f \right\rangle \right\|^2$. We say that an algorithm computes a function $f : D \to T$ within error ε if $\pi_{f(x)}(x) \geq 1 - \varepsilon$ for all input $x \in D \subset \{0,1\}^n$.

2.1 An alternative formulation for the QQM

In this section, we introduce notation from a previous work [18]. We define a product of unitary operators $\widetilde{U}_n = U_n U_{n-1} \ldots U_0$. We denote a CSOP $\left\{\bar{P}_k : 0 \leq k \leq n\right\}$, where the range of each $\bar{P}_i$ is composed by vectors of the form $|i\rangle\,|\psi\rangle \in \mathcal{H}$, for $i \in \{0, 1, .., n\}$ and any state $|\psi\rangle$. We also introduce the notation $\widetilde{P}_i^j = \widetilde{U}_j^\dagger \bar{P}_i \widetilde{U}_j$. Notice that for any fixed j we have that $\left\{\widetilde{P}_k^j : 0 \leq k \leq n\right\}$ is also a CSOP. The following definition introduces an alternative representation for quantum query algorithms on the QQM.

Definition 2 Consider a set $\mathbb{Z}_{n+1} = \{0, 1, \ldots, n\}$. An indexed set of vectors $\left\{|\Psi(k)\rangle \in \mathcal{H} : k \in \mathbb{Z}_{n+1}^{t+1}\right\}$ is associated with a quantum query algorithm if we have that

$$|\Psi(a)\rangle = \widetilde{P}_{a_t}^t \ldots \widetilde{P}_{a_1}^1 \widetilde{P}_{a_0}^0 |\Psi\rangle, \tag{2}$$

for all $a \in \mathbb{Z}_{n+1}^{t+1}$.

In Lemma 1, we show that vectors associated with some algorithm represent the final state as phase flips [18]. In Sect. 4, we analyze the relation between minimum norm (or cardinality) of such vectors, and the computational gap between classical and quantum query.

Lemma 1 *If the indexed set of vectors* $\left\{|\Psi(k)\rangle \in H_A : k \in \mathbb{Z}_{n+1}^{t+1}\right\}$ *is associated with a quantum algorithm then*

$$\widetilde{U}_t^\dagger O_x U_t \ldots U_1 O_x U_0 |\Psi\rangle = \sum_{k_t=0}^{n} \cdots \sum_{k_0=0}^{n} (-1)^{\sum_{i=0}^{t} x_{k_i}} |\Psi(k_0, \ldots, k_t)\rangle. \tag{3}$$

Proof Following Ref. [18], we give a proof by induction on t. For $t = 0$, we have that Eq. (3) holds,

$$\widetilde{U}_0^\dagger O_x U_0 |\Psi\rangle = U_0^\dagger O_x U_0 |\Psi\rangle \tag{4}$$

$$= \sum_{k_0=0}^{n} (-1)^{x_{k_0}} |\Psi(k_0)\rangle. \tag{5}$$

For the second part of the induction, we shall notice that the equation

$$O_x |\Psi\rangle = \sum_{i \in \{k : x_k = 0\}} \bar{P}_i |\Psi\rangle - \sum_{i \in \{k : x_k = 1\}} \bar{P}_i |\Psi\rangle \tag{6}$$

implies the equation

$$\widetilde{U}_j^\dagger O_x \widetilde{U}_j |\Psi\rangle = \sum_{i \in \{k : x_k = 0\}} \widetilde{U}_j^\dagger \bar{P}_i \widetilde{U}_j |\Psi\rangle - \sum_{i \in \{k : x_k = 1\}} \widetilde{U}_j^\dagger \bar{P}_i \widetilde{U}_j |\Psi\rangle . \tag{7}$$

Suppose that Eq. (3) holds for some t, then applying Eq. (7) we obtain

$$\begin{aligned}
&\widetilde{U}_{t+1}^\dagger O_x U_t \ldots U_1 O_x U_0 |\Psi\rangle \\
&= \sum_{k_t=0}^{n} \ldots \sum_{k_0=0}^{n} (-1)^{\sum_{i=0}^{t} x_{k_i}} \sum_{k_{t+1}=0}^{n} (-1)^{x_{k_{t+1}}} \widetilde{P}_{k_{t+1}}^{t+1} |\Psi (k_0, \ldots, k_t)\rangle \\
&= \sum_{k_{t+1}=0}^{n} \ldots \sum_{k_0=0}^{n} (-1)^{\sum_{i=0}^{t+1} x_{k_i}} \widetilde{P}_{k_{t+1}}^{t+1} |\Psi (k_0, \ldots, k_t)\rangle \\
&= \sum_{k_{t+1}=0}^{n} \ldots \sum_{k_0=0}^{n} (-1)^{\sum_{i=0}^{t+1} x_{k_i}} |\Psi (k_0, \ldots, k_{t+1})\rangle .
\end{aligned}$$

□

The previous theorem shows that a quantum state depends on several components whose phases change independently on input x. Notice that the phase $(-1)^{\sum_{i=0}^{t} x_{k_i}}$ of each component $|\Psi (k_0, \ldots, k_t)\rangle$ is a Walsh function. Then, each of the components depends on t values from input, which at first sight is not impressive, considering that deterministic classical algorithms compute any function that depends on t values using t queries. However, all components together depend on the size n of input. Thus, we have the possibility of computing on n variables using just t queries, which gives us another intuition about the computational speed-up by quantum means. Therefore, this formulation presents quantum parallelism more explicitly than a sequence of unitary operators.

3 A classical simulation for quantum query algorithms and polynomials

In this section, we introduce our simulation of quantum query algorithms by classical algorithms. However, our simulation can also be extended to polynomials. This simulation is defined over the output probability $\pi_z(x)$ of the quantum algorithm.

We consider the Fourier basis for the vector space of all functions $f : \{0, 1\}^n \to \mathbb{R}$ [15] given by the functions

$$\chi_b : \{0, 1\}^n \to \{1, -1\} ,$$

such that $\chi_b(x) = (-1)^{b\cdot x}$ for $b \in \{0, 1\}^n$ and $b \cdot x = \sum_i b_i x_i$. This family contains a constant function that we denote as $\chi_0 = 1$. Therefore, any function $f : \{0, 1\}^n \to \mathbb{R}$ can be represented as a linear combination

$$f = \sum_{b\in\{0,1\}^n} \alpha_b \chi_b, \tag{8}$$

and we denote the Fourier 1-norm of f as

$$L(f) = \sum_{b\in\{0,1\}^n} |\alpha_b|. \tag{9}$$

Another measure is the degree of f, which is defined as

$$\deg(f) = \max_{|b|} \{b : \alpha_b \neq 0\}, \tag{10}$$

where $|b|$ denotes the number of ones in b.

Figure 1 presents the intuition behind our simulation. At the right, we have $\pi_1(x)$, the probability of obtaining output 1 by a quantum query algorithm on input x. Such function is decomposed into a linear combination of functions χ_b, following Eq. (8). The sub-simulations imply emulating each function χ_b by a classical algorithm that outputs 1 with probability $\widehat{\pi}_1^b(x)$, where: (1) $\chi_b(x) = 1$ implies that $\widehat{\pi}_1^b(x) = 1$; and, (2) $\chi_b(x) = -1$ implies that $\widehat{\pi}_1^b(x) = 0$. Notice that each $\widehat{\pi}_1^b(x)$ is a probability and cannot have negative values as functions χ_b. The composition step is assigning appropriate probabilities to each output $\widehat{\pi}_1^b(x)$, such that the sum produces an output probability whose shape resembles $\pi_1(x)$. As each $\widehat{\pi}_1^b(x)$ is similar but different in relation to χ_b, this procedure accumulates an important error. The proof of the following theorem shows the details.

Theorem 1 *Let $\mathcal{A}$ be a quantum algorithm that computes $f : S \to \{0, 1\}$ for $S \subset \{0, 1\}^n$, within error ε and t queries. Then, there is a classical algorithm which computes f within error*

$$\tilde{\varepsilon} = \frac{\varepsilon + L(\pi_1)}{1 + 2L(\pi_1)}$$

and $2t$ queries.

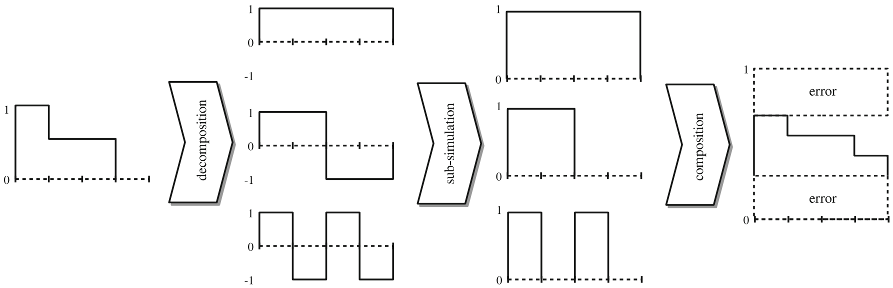

Fig. 1 The simulation produces a contracted version of the original output probability. The new output probability can be represented as a linear transformation applied over the original output probability

Proof If a quantum algorithm applies t queries, then $\deg(\pi_z) \leq 2t$ for every output z [10]. Let $\mathcal{D}(b)$ be the deterministic classic algorithm which outputs

$$\widehat{\pi}_1^b(x) = \frac{1}{2} + \operatorname{sgn}(\alpha_b)\left(\frac{\chi_b(x)}{2}\right), \tag{11}$$

for input x, where sgn is the sign function and $|b| \leq 2t$. We consider a randomized algorithm $\mathcal{R}$ which simply selects either: (1) an algorithm $\mathcal{D}(b)$, with probability $\frac{2|\alpha_b|}{1+2L(\pi_1)}$; or, (2) an algorithm that outputs 0 for any x, with probability $\frac{1}{1+2L(\pi_1)}$. Notice that algorithm $\mathcal{R}$ is the composition of sub-simulations, as we represent in Fig. 1. Since we denote by $\widehat{\pi}_1(x)$ the probability of obtaining output 1 given x with $\mathcal{R}$, by Eq. (11) we have

$$\widehat{\pi}_1(x) = \frac{\sum\limits_b 2|\alpha_b|\,\widehat{\pi}_1^b(x)}{1+2L(\pi_1)} \tag{12}$$

$$= \frac{\sum\limits_b |\alpha_b| + \sum\limits_b \alpha_b\chi_b(x)}{1+2L(\pi_1)}. \tag{13}$$

The algorithm $\mathcal{R}$ applies no more than $2t$ queries, since $\mathcal{D}(b)$ applies no more than $2t$ queries for each $|b| \leq 2t$.

Now, we must prove an upper-bound for the error in the simulation. We divide such proof in two cases, when $f(x) = 1$ and $f(x) = 0$. If $f(x) = 1$, then $\varepsilon \geq 1 - \pi_1(x) = 1 - \sum\limits_b \alpha_b\chi_b(x)$. This implies that

$$1 - \widehat{\pi}_1(x) = 1 - \frac{\left(L(\pi_1) + \sum\limits_b \alpha_b\chi_b(x)\right)}{1+2L(\pi_1)} \tag{14}$$

$$= \frac{1 + L(\pi_1) - \sum\limits_b \alpha_b\chi_b(x)}{1+2L(\pi_1)} \tag{15}$$

$$\leq \tilde{\varepsilon}. \tag{16}$$

Analogously, if $f(x) = 0$, then $\varepsilon \geq \pi_1(x) = \sum\limits_b \alpha_b\chi_b(x)$ and this implies that

$$\widehat{\pi}_1(x) \leq \frac{\varepsilon + L(\pi_1)}{1+2L(\pi_1)} = \tilde{\varepsilon}. \tag{17}$$

□

We described a classical simulation that imitates the output probability of a given quantum algorithm, but within a big error. Thus, the next theorem just gives a reduction in such error using probabilistic amplification.

Theorem 2 *Let $\mathcal{A}$ be a quantum algorithm that computes $f : S \to \{0, 1\}$ for $S \subset \{0, 1\}^n$, with error ε and t queries. Then, there is a classical algorithm which computes*

f within error $\exp\left(-\frac{j}{2(1-\tilde{\varepsilon})}\left(\frac{1}{2}-\tilde{\varepsilon}\right)^2\right)$*, where* $\tilde{\varepsilon}=\frac{\varepsilon+L(\pi_1)}{1+2L(\pi_1)}$ *and using* $2jt$ *queries for* $j\in\mathbb{N}$.

Proof We use a corollary of Chernoff bound [8]. For j, p, β such that $0\leq p\leq 1$, $0\leq\beta\leq 1$ and $0\leq j$, we have

$$\sum_{i=0}^{m}\binom{j}{i}p^i(1-p)^{j-i}\leq\exp\left(-\beta^2 jp/2\right), \tag{18}$$

where $m=\lfloor(1-\beta)\,jp\rfloor$.

We define an algorithm $\widehat{\mathcal{R}}$ using the classical algorithm $\mathcal{R}$ within error $\tilde{\varepsilon}$ from Theorem 1. Algorithm $\widehat{\mathcal{R}}$ consists in applying probability amplification on $\mathcal{R}$, that is, executing algorithm $\mathcal{R}$ j times and then selecting the most frequent result. Define X as the random variable that represents the number of correct answers. Taking $\beta=1-\frac{1}{2(1-\tilde{\varepsilon})}$ and $p=(1-\tilde{\varepsilon})$ in Eq. (18), then the error in $\widehat{\mathcal{R}}$ is upper-bounded by

$$\mathbb{P}\left[X\leq\left\lfloor\frac{j}{2}\right\rfloor\right]\leq\exp\left(-\frac{j}{2(1-\tilde{\varepsilon})}\left(\frac{1}{2}-\tilde{\varepsilon}\right)^2\right). \tag{19}$$

□

3.1 Polynomial simulation

The same technique can be applied for simulating a polynomial $p(x)$ approximating a function, instead of simulating the output probabilities of a given quantum algorithm. In this sense, Theorems 1 and 2 can be generalized. In order to formulate the corresponding theorems, we consider the usual notion of polynomial approximation:

Definition 3 A polynomial $p:\mathbb{R}^n\to\mathbb{R}$ ε-approximates a function $f:S\to\{0,1\}$ for $S\subset\{0,1\}^n$, if $|p(x)-f(x)|\leq\varepsilon$ for all $x\in\{0,1\}^n$.

Theorem 3 *Let* $p:\mathbb{R}^n\to\mathbb{R}$ *be a polynomial that* ε*-approximates* $f:S\to\{0,1\}$ *for* $S\subset\{0,1\}^n$*. If* p *has a degree equal or less than* $2t$*, then there is a classical algorithm which computes* f *within error*

$$\tilde{\varepsilon}=\frac{\varepsilon+L(p)}{1+2L(p)}$$

and $2t$ *queries.*

Proof The proof is similar to the proof of Theorem 1. □

We similarly introduce the corresponding reduction error theorem.

Theorem 4 *Let* p *be a polynomial that* ε*-approximates* $f:S\to\{0,1\}$ *for* $S\subset\{0,1\}^n$*. If* p *has a degree equal or less than* $2t$*, then there is a classical algorithm which computes* f *within error* $\exp\left(-\frac{j}{2(1-\tilde{\varepsilon})}\left(\frac{1}{2}-\tilde{\varepsilon}\right)^2\right)$*, where* $\tilde{\varepsilon}=\frac{\varepsilon+L(p)}{1+2L(p)}$ *and using* $2jt$ *queries for* $j\in\mathbb{N}$.

Proof The proof is similar to the proof of Theorem 2, but reducing error in Theorem 3 instead Theorem 1. □

4 Upper-bounds for quantum speed-up

In this section, we describe conditions which can slow down our simulation. Quantum speed-up only occurs when no classical simulation is efficient enough, thus any condition that makes difficult any classical simulation is a necessary condition for this computational gain. In this sense, we measure the quantum speed-up for a given quantum algorithm by the quotient $\frac{R}{t}$, where (1) such quantum algorithm applies t queries, and (2) an optimal classical algorithm executes the same computational task in R queries. This quotient can be interpreted as how much faster is a quantum algorithm in relation to the best classical algorithm.

The following theorem, which upper-bounds quantum speed-up using Fourier 1-norm, is the core of our results. It basically shows how high values for Fourier 1-norm are related to the speed quotient that we denoted.

Theorem 5 *Consider $D \subset \{0,1\}^n$ and a function $f : D \to \{0,1\}$ that is computed within error $\varepsilon > 0$ and t queries, by a quantum query algorithm. If we define*

$$F_\varepsilon(l) = \left\lceil \frac{-16 \ln(\varepsilon)(1+l)(1+l-\varepsilon)}{(1-2\varepsilon)^2} \right\rceil, \tag{20}$$

then

$$\frac{R_\varepsilon(f)}{t} \leq F_\varepsilon(L(\pi_1)), \tag{21}$$

where (1) $R_\varepsilon(f)$ denotes the minimum number of queries that are necessary for computing f within error ε by a randomized decision tree (See [12] *for a detailed definition.) and (2) $\pi_1(x)$ is the probability of the quantum algorithm returning output 1 for a given input x.*

Proof Suppose that we simulate the quantum algorithm using the randomized algorithm of Theorem 2 and promising an error that does not exceed ε for f. Thereby, from Eq. (19), we have

$$\varepsilon = \exp\left(-\frac{j}{2(1-\tilde{\varepsilon})}\left(\frac{1}{2}-\tilde{\varepsilon}\right)^2\right). \tag{22}$$

As $\frac{R_\varepsilon(f)}{t} \leq \lceil 2j \rceil$, if we obtain j from Eq. (22) we have Eq. (21). □

Last theorem has consequences in exact quantum complexity, as we find in the following corollary.

Corollary 1 *Consider a total function $f : D \to \{0,1\}$, then*

$$\frac{R_\varepsilon(f)}{Q_E(f)} \leq F_\varepsilon(L(f)), \tag{23}$$

where $Q_E(f)$ denotes the number of queries applied by an exact quantum query algorithm computing f.

Proof If a quantum query algorithm is exact, optimal and computes a total function then $t = Q_E(f)$ and $\pi_1 = f$. □

Theorem 5 can also be formulated for approximate polynomials, as follows.

Theorem 6 *Consider $D \subset \{0,1\}^n$ and a function $f : D \to \{0,1\}$ that is ε-approximated by a polynomial $p : \mathbb{R}^n \to \mathbb{R}$. If $\deg(p) \leq 2t$, then*

$$\frac{R_\varepsilon(f)}{2t} \leq F_\varepsilon(L(p)). \tag{24}$$

Proof Similar proof as for Theorem 5, but applying Theorem 4. □

We may expect from Fourier 1-norm that low values must imply problems that are easily simulated by classical means. Theorem 5 guarantees that low values of the Fourier 1-norm in relation to t imply such efficient classical simulation. Notice that Fourier 1-norm is defined on the output probability. Then, an explicit expression for the Fourier 1-norm as a function of the algorithm itself may be useful. Let k, h be vectors in $\mathbb{Z}^t_{n+1}$ and $|b| \leq 2t$. We denote $(k,h) \sim b$, if

$$(-1)^{\sum\limits_i x_{k_i} + \sum\limits_i x_{h_i}} = \chi_b(x).$$

Thus, for a t-query algorithm, we have the expression

$$L(\pi_1) = \sum_{|b| \leq 2t} \left| \sum_{(k,h) \sim b} \langle \Psi(k) | P_1 | \Psi(h) \rangle \right|, \tag{25}$$

by applying Lemma 1. Considering that each pair (k,h) is related to a unique b, we can obtain the following upper-bound for $L(\pi_1)$,

$$\widetilde{L}(\pi_1) = \sum_k \sum_h |\langle \Psi(k) | P_1 | \Psi(h) \rangle|. \tag{26}$$

These expressions are based on the state decomposition given by Definition 2. Lemma 1 implies that each quantum algorithm has its own state decomposition, thus next theorem relates metrics on such set of vectors with the gap between quantum and classical query.

Theorem 7 *Using the same hypothesis of Theorem 5, denoting $\#S$ as the cardinality of set S and defining $d = \#\{k : |\Psi(k)\rangle \neq 0\}$, we have*

$$\frac{R_\varepsilon(f)}{t} \leq F_\varepsilon\left(\widetilde{L}(\pi_1)\right), \tag{27}$$

$$\frac{R_\varepsilon(f)}{t} \leq F_\varepsilon\left(\left(\sum_k \| |\Psi(k)\rangle \|\right)^2\right), \tag{28}$$

$$\frac{R_\varepsilon(f)}{t} \leq F_\varepsilon(d), \tag{29}$$

and

$$\frac{R_\varepsilon(f)}{t} \leq F_\varepsilon\left(\frac{1}{\min\limits_k \langle \Psi(k) | \Psi(k)\rangle}\right). \tag{30}$$

Proof As F_ε is an increasing function, Eq. (27) follows directly from Eq. (26) and Theorem 5. Equation (28) is also derived from Eq. (26) by observing that

$$|\langle \Psi(k)| P_z |\Psi(h)\rangle| \leq \| |\Psi(k)\rangle \| \, \| |\Psi(h)\rangle \|, \tag{31}$$

which gives

$$L(\pi_1) \leq \left(\sum_k \| |\Psi(k)\rangle \|\right)^2. \tag{32}$$

Applying Lemma 1, we obtain

$$\langle \Psi | \Psi \rangle = \sum_{k,h} \langle \Psi(k) | \Psi(h)\rangle = 1. \tag{33}$$

Then, using it with $\sum\limits_k \| |\Psi(k)\rangle \| \leq \sqrt{d} \sum\limits_k \langle \Psi(k) | \Psi(k)\rangle$ and Eq. (32), we have

$$L(\pi_1) \leq d. \tag{34}$$

Finally, Eq. (30) follows from $d\left(\min\limits_k \langle \Psi(k) | \Psi(k)\rangle\right) \leq 1$. □

5 Alternative applications

Our results from Sect. 4 have a main theoretical motivation, which is showing the relation between quantum speed-up and quantum parallelism. Furthermore, the theorems are interesting for related subjects that we discuss below.

5.1 Upper-bounds for randomized complexity

Theorems 5 and 6 may be applied for finding upper-bounds on R_ε. For example, consider Deutsch–Jozsa algorithm, thereby we have the output probability

$$\pi_1(x) = \frac{1}{n^2}(n - 2|x|)^2$$

for inputs of size n. We obtain the terms $\{\alpha_b\}$ by applying the pairwise orthogonality between functions χ_b. The algorithm works by applying just one query. Thus, from the fact that $\deg(\pi_1(x)) \leq 2$ [10], we have that if $|b| > 2$ then $\alpha_b = 0$. This leaves us with three cases to analyze. First, if $|b| = 0$, then $\alpha_b = \frac{1}{n}$, notice that there is just one index b satisfying $|b| = 0$. Second, if $|b| = 1$, then $\alpha_b = 0$. Third, there are $\frac{n(n-1)}{2}$ indices b such that $|b| = 2$, in this case $\alpha_b = \frac{2}{n^2}$. Therefore, we have that $\sum |\alpha_b| = 1$, which implies

$$R_\varepsilon \leq \left\lceil \frac{-16\ln(\varepsilon)(2-\varepsilon)}{(1-2\varepsilon)^2} \right\rceil$$

by Eq. (21). This is not quite tight numerically because a classical decision tree applies 2 queries in order to solve Deutsch–Jozsa problem within error $\frac{1}{3}$. However, this is asymptotically tight and proves that Deutsch–Jozsa algorithm can be simulated classically using a constant number of queries and fixed error.

5.2 Lower-bounds for exact quantum complexity

Corollary 1 can be applied for finding lower-bounds on Q_E. For example, consider the total function $AND_n : \{0,1\}^n \to \{0,1\}$ where $AND_n(x) = 1$ if and only if $x_i = 1$ for all i. We denote *weight* of input x as the number of ones in x. A randomized decision tree computing AND_n must discriminate the input with weight n from the set of inputs with weight $n-1$. Suppose that some randomized decision tree computes AND_n with less than $\frac{n}{3}$ queries, then such randomized tree is a probabilistic distribution over a set of deterministic decision trees querying less than $\frac{n}{3}$ values in x. Then, in order to discriminate an input of weight n from the set of inputs with weight $n-1$, the randomized tree will find 0 for some x_i with expectation less than $\frac{1}{3}$. In this sense, $R_{\frac{1}{3}}(AND_n) \geq \frac{n}{3} - 1$.

Considering that $L(AND_n) = 1$, we have $Q_E(AND_n) \in \Omega(n)$ by Eq. (23), which is asymptotically tight [7].

5.3 Polynomial approximation by quantum algorithms

There is an equivalence between 1-query algorithms and degree-2 polynomials. That is, a partial Boolean function f can be approximated by a polynomial for some error bounded by $\varepsilon > \frac{1}{2}$ if and only if f can be computed by a quantum algorithm with error bounded by $\varepsilon' > \frac{1}{2}$ and a single query. However, the problem of transforming polynomials of higher degree to quantum algorithms still needs more results [4]. Theorem 3 implies that t-query algorithms compute any function approximated by degree-t polynomials with Fourier 1-norm bounded by a constant. Then, the high degree problem is reduced to find algorithms for polynomials with a high Fourier 1-norm.

6 Conclusion

In the present work, we identified a necessary property for a hard classical simulation of quantum query algorithms, namely a high Fourier 1-norm defined over the output probability. A remarkable feature about Fourier 1-norm is that it depends on both evolution and measurement steps. Properties like quantum entanglement are defined just on the quantum states, which implies that a poor measurement step can cancel advantages obtained in the evolution stage, where we assume that such evolution stage was hard to simulate. Nevertheless, the accuracy of Fourier 1-norm for approximating quantum gain depends on a simulation, whose relation with the most efficient classical simulation is unknown.

We also formalized the advantage given by quantum algorithms, as the quotient between the classical and quantum complexities for a given task. We have that such quotient is upper-bounded by an expression which depends quadratically on the Fourier 1-norm. Thus, a large factor produced between quantum and classical algorithms implies a large Fourier 1-norm. Our result suggests the following intuitions:

1. Output probabilities with large Fourier 1-norms imply that such output probability can be represented by a function whose shape is much different from any function in the Fourier basis—functions that can be efficiently simulated by classical means.
2. Output probabilities with high Fourier 1-norms imply that many functions from Fourier basis are acting simultaneously. This strongly suggests quantum parallelism.

We can also link Fourier 1-norm to quantum parallelism as follows. A quantum query algorithm can be viewed as a state decomposition by Lemma 1, which is denoted as a set of vectors associated to the algorithm. This formulation emphasizes the presence of quantum parallelism, because each combination of vectors in the decomposition represents a function in the Fourier basis, where such functions are added producing an output probability function. The Fourier 1-norm is related to this decomposition. Since a high Fourier 1-norm implies: (a) a big number of nonzero vectors in such decomposition, i.e., high values for $\#\left\{k : |\Psi(k)\rangle \neq 0\right\}$; and, (b) minimum product values that are not too big for such vectors, i.e., low values for $\left(\min_k \langle \Psi(k) |\, \Psi(k)\rangle\right)$; then (a) and (b) are also necessary conditions for a hard classical simulation. Both measures can be linked to quantum parallelism by the following intuition. If $\#\left\{k : |\Psi(k)\rangle \neq 0\right\}$ is low, then there are less combinations of vectors adding functions on the output probability function. Larger values for $\min_k \langle \Psi(k) |\, \Psi(k)\rangle$ imply lower values for $\#\left\{k : |\Psi(k)\rangle \neq 0\right\}$. However, it also implies that the output probability function has a shape closer to functions in the Fourier basis, hence such output probability has a cheap classical simulation. The intuition behind is that Fourier 1-norm is a measure for quantum parallelism and gives a quadratic upper-bound to the potential quantum speed-up.

Finally, the present work leaves the following open problems:

- An additional motivation for studying the Fourier 1-norm is that it is a parameter that we may control. Finding degree-2 polynomials is an alternative strategy for obtaining 1-query quantum algorithms [4]. Thus, developing a method for obtaining high 1-norm polynomials of degree 2 and bounded in $\{0, 1\}^n$ would help to find algorithms offering a potential advantage over classical algorithms.

- A high Fourier 1-norm implies a necessary condition for quantum query speed-up. Can we obtain a necessary and sufficient condition by adding another property?

Acknowledgements The authors thank R. Portugal, S. Collier, J. Szwarcfiter, and E. Galvão for useful discussions and suggestions. This work was initiated while SAG was at the Federal University of Rio de Janeiro, Brazil. This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)* - Finance Code 001, and *Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro - Brasil (FAPERJ)*, JCNE Grant E-26/203.235/2016.

References

1. Aaronson, S.: Limitations of quantum advice and one-way communication. In: Proceedings of 19th IEEE Annual Conference on Computational Complexity, pp. 320–332. IEEE (2004)
2. Aaronson, S.: BQP and the polynomial hierarchy. In: Proceedings of the Forty-Second ACM Symposium on Theory of Computing, pp. 141–150. ACM (2010)
3. Aaronson, S., Ambainis, A.: The need for structure in quantum speedups. Theory Comput. **10**(6), 133–166 (2014)
4. Aaronson, S., Ambainis, A., Iraids, J., Kokainis, M., Smotrovs, J.: Polynomials, quantum query complexity, and Grothendieck's inequality. In: Proceedings of the 31st Conference on Computational Complexity, CCC'16, pp. 25:1–25:19. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Germany (2016). https://doi.org/10.4230/LIPIcs.CCC.2016.25
5. Abbott, A.A., Calude, C.S.: Understanding the quantum computational speed-up via de-quantisation. In: Cooper, S.B., Panangaden, P., Kashefi, E. (eds.) Proceedings Sixth Workshop on Developments in Computational Models: Causality, Computation, and Physics, Edinburgh, Scotland, 9–10th July 2010, Electronic Proceedings in Theoretical Computer Science, vol. 26, pp. 1–12. Open Publishing Association (2010). https://doi.org/10.4204/EPTCS.26.1
6. Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM J. Comput. **37**(1), 210–239 (2007)
7. Ambainis, A., Gruska, J., Zheng, S.: Exact quantum algorithms have advantage for almost all Boolean functions. Quantum Inf. Comput. **15**(5–6), 435–452 (2015)
8. Angluin, D., Valiant, L.G.: Fast probabilistic algorithms for hamiltonian circuits and matchings. J. Comput. Syst. Sci. **18**(2), 155–193 (1979)
9. Barnum, H., Saks, M., Szegedy, M.: Quantum query complexity and semi-definite programming. In: Proceedings of the 18th IEEE Annual Conference on Computational Complexity, pp. 179–193 (2003). https://doi.org/10.1109/CCC.2003.1214419
10. Beals, R., Buhrman, H., Cleve, R., Mosca, M., De Wolf, R.: Quantum lower bounds by polynomials. J. ACM **48**(4), 778–797 (2001)
11. Bell, J.S.: On the problem of hidden variables in quantum mechanics. Rev. Mod. Phys. **38**(3), 447 (1966)
12. Buhrman, H., De Wolf, R.: Complexity measures and decision tree complexity: a survey. Theor. Comput. Sci. **288**(1), 21–43 (2002)
13. Datta, A., Shaji, A., Caves, C.M.: Quantum discord and the power of one qubit. Phys. Rev. Lett. **100**(5), 050502 (2008)
14. Datta, A., Vidal, G.: Role of entanglement and correlations in mixed-state quantum computation. Phys. Rev. A **75**(4), 042310 (2007)
15. De Wolf, R.: A brief introduction to fourier analysis on the Boolean cube. Theory Comput. Grad. Surv. **1**, 1–20 (2008)
16. Deutsch, D.: Quantum theory, the Church–Turing principle and the universal quantum computer. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 400, pp. 97–117. The Royal Society (1985)

17. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 439, pp. 553–558. The Royal Society (1992)
18. Grillo, S., Marquezino, F.: Quantum query as a state decomposition. Theor. Comput. Sci. **736**, 62–75 (2018). https://doi.org/10.1016/j.tcs.2018.03.017
19. Howard, M., Wallman, J., Veitch, V., Emerson, J.: Contextuality supplies the 'magic' for quantum computation. Nature **510**(7505), 351–355 (2014)
20. Jozsa, R.: Entanglement and quantum computation. In: Huggett, S.A., Mason, L.J., Tod, K.P., Tsou, S.T., Woodhouse, N.M.J. (eds.) The Geometric Universe: Science, Geometry, and the Work of Roger Penrose, vol. 27, pp. 369–379. Oxford University Press, Oxford (1998)
21. Jozsa, R., Linden, N.: On the role of entanglement in quantum-computational speed-up. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 459, pp. 2011–2032. The Royal Society (2003)
22. Knill, E., Laflamme, R.: Power of one bit of quantum information. Phys. Rev. Lett. **81**(25), 5672 (1998)
23. Kochen, S., Specker, E.P.: The problem of hidden variables in quantum mechanics. In: Hooker, C.A. (eds.) The Logico-Algebraic Approach to Quantum Mechanics, vol. 5a, pp. 293–328. Springer, Dordrecht (1975). https://doi.org/10.1007/978-94-010-1795-4_17
24. Montanaro, A.: Some applications of hypercontractive inequalities in quantum information theory. J. Math. Phys. **53**(12), 122206 (2012)
25. O'Donnell, R.: Analysis of Boolean Functions. Cambridge University Press, Cambridge (2014)
26. Ozols, M., Roetteler, M., Roland, J.: Quantum rejection sampling. ACM Trans. Comput. Theory **5**(3), 11 (2013)
27. Reichardt, B.W., Spalek, R.: Span-program-based quantum algorithm for evaluating formulas. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, pp. 103–112. ACM (2008)
28. Rötteler, M.: Quantum algorithms for highly non-linear Boolean functions. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 448–457. SIAM (2010)
29. Vidal, G.: Efficient classical simulation of slightly entangled quantum computations. Phys. Rev. Lett. **91**(14), 147902 (2003)