ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Complexity 19 (2003) 1–18

Journal of COMPLEXITY

http://www.elsevier.com/locate/jco

On a problem in quantum summation

Stefan Heinrich[a,*] and Erich Novak[b]

[a] Fachbereich Informatik, Universität Kaiserslautern, Postfach 3049, D-67653 Kaiserslautern, Germany
[b] Mathematisches Institut, Universität Jena, D-07740 Jena, Germany

Received 5 September 2001; accepted 11 July 2002

Abstract

We consider the computation of the mean of sequences in the quantum model of computation. We determine the query complexity in the case of sequences which satisfy a p-summability condition for $1 \leqslant p < 2$. This settles a problem left open in Heinrich (J. Complexity 18 (2002) 1).

1. Introduction

Computation of the mean of sequences and, equivalently, summation of sequences, is an important numerical task, in particular for huge number of summands occurring in many numerical applications such as, e.g., high-dimensional integration. The larger the number of summands (the larger the dimension), the less these problems are tractable. It is therefore an interesting and challenging task to understand to which extent a quantum computer could bring speed-ups. First, results for the summation of bounded sequences are due to Grover [6], Nayak and Wu [11], Brassard et al. [3]. The case of sequences satisfying a p-summability condition, which arises in various problems like integration of functions from L_p and Sobolev classes, was studied in [8]. Up to logarithmic factors for $p = 2$, in the case $2 \leqslant p < \infty$ the query complexity of the summation problem was determined. For the case $1 \leqslant p < 2$, matching upper and lower bounds were obtained only under an additional restriction. The bounds for the remaining case did not match. In this

*Corresponding author. Tel.: +49-631-205-2992; fax: +49-631-205-3270.

E-mail addresses: heinrich@informatik.uni-kl.de (S. Heinrich), novak@mathematik.uni-jena.de (E. Novak).

0885-064X/02/$ - see front matter © 2002 Elsevier Science (USA). All rights reserved.
PII: S0885-064X(02)00003-1

paper we settle this problem and determine the query complexity in the full range of parameters.

Applications of our results to the quantum complexity of integration of functions from Sobolev classes are given in [9]. The use of quantum summation for integration was first pointed out by Abrams and Williams [1]. The quantum complexity of integration was studied in [15], later in [8,10]. Path integration is discussed in [18]. Furthermore, we refer to the surveys [4,17], and to the monographs [7,12,16] for general reading on quantum computation.

Our analysis is based on the framework introduced in [8] of quantum algorithms for the approximate solution of problems of analysis. This approach is an extension of the framework of information-based complexity theory (see [13,19] and, more formally, [14]) to quantum computation. It also extends the binary black box model of quantum computation (see, e.g., [2]) to situations where mappings from spaces of functions to the scalar field (such as the mean or the integral) have to be computed. Let us recall the main notions here. For more details and background discussion we refer to [8].

2. Notation

Let D, K be nonempty sets, let $\mathscr{F}(D,K)$ denote the set of all functions from D to K, and let $F \subseteq \mathscr{F}(D,K)$ be a nonempty subset. Let $\mathbf{K}$, the scalar field, be either $\mathbf{R}$ or $\mathbf{C}$, the field of real or complex numbers, let G be a normed space over $\mathbf{K}$, and let $S : F \to G$ be a mapping. We seek to approximate $S(f)$ for $f \in F$ by means of quantum computations. Let H_1 be the two-dimensional complex Hilbert space $\mathbf{C}^2$, with its unit vector basis $\{e_0, e_1\}$, let

$$H_m = H_1 \otimes \cdots \otimes H_1$$

be the tensor product of m copies of H_1, endowed with the tensor Hilbert space structure. The following notation is convenient:

$$\mathbf{Z}[0,N) := \{0, \ldots, N-1\}$$

for $N \in \mathbf{N}$ (as usual, $\mathbf{N} = \{1,2,\ldots\}$ and $\mathbf{N}_0 = \mathbf{N} \cup \{0\}$). Let $\mathscr{C}_m = \{|i\rangle : i \in \mathbf{Z}[0,2^m)\}$ be the canonical basis of H_m, where $|i\rangle$ stands for $e_{j_0} \otimes \cdots \otimes e_{j_{m-1}}$, $i = \sum_{k=0}^{m-1} j_k 2^{m-1-k}$ the binary expansion of i. Denote the set of unitary operators on H_m by $\mathscr{U}(H_m)$.

A quantum query on F is given by a tuple

$$Q = (m, m', m'', Z, \tau, \beta), \tag{1}$$

where $m, m', m'' \in \mathbf{N}, m' + m'' \leqslant m, Z \subseteq \mathbf{Z}[0, 2^{m'})$ is a nonempty subset, and

$$\tau : Z \to D$$

$$\beta : K \to \mathbf{Z}[0, 2^{m''})$$

are arbitrary mappings. Denote $m(Q) := m$, the number of qubits of Q.

Given such a query Q, we define for each $f \in F$ the unitary operator Q_f by setting for $|i\rangle|x\rangle|y\rangle \in \mathscr{C}_m = \mathscr{C}_{m'} \otimes \mathscr{C}_{m''} \otimes \mathscr{C}_{m-m'-m''}$:

$$Q_f|i\rangle|x\rangle|y\rangle = \begin{cases} |i\rangle|x \oplus \beta(f(\tau(i)))\rangle|y\rangle & \text{if } i \in Z, \\ |i\rangle|x\rangle|y\rangle & \text{otherwise,} \end{cases} \tag{2}$$

where $\oplus$ means addition modulo $2^{m''}$.

A quantum algorithm on F with no measurement is a tuple

$$A = (Q, (U_j)_{j=0}^{n}),$$

where Q is a quantum query on F, $n \in \mathbf{N}_0$ and $U_j \in \mathscr{U}(H_m)$ $(j = 0, \ldots, n)$, with $m = m(Q)$. Given $f \in F$, we let $A_f \in \mathscr{U}(H_m)$ be defined as

$$A_f = U_n Q_f U_{n-1} \ldots U_1 Q_f U_0. \tag{3}$$

We denote by $n_q(A) := n$ the number of queries and by $m(A) = m = m(Q)$ the number of qubits of A. Let $(A_f(x,y))_{x,y \in \mathbf{Z}[0,2^m)}$ be the matrix of the transformation A_f in the canonical basis $\mathscr{C}_m$, that is, $A_f(x,y) = (A_f|y\rangle, |x\rangle)$.

A quantum algorithm on F with output in G (or shortly, from F to G) with k measurements is a tuple

$$A = ((A_\ell)_{\ell=0}^{k-1}, (b_\ell)_{\ell=0}^{k-1}, \varphi),$$

where $k \in \mathbf{N}$, and A_ℓ $(\ell = 0, \ldots, k-1)$ are quantum algorithms on F with no measurements,

$$b_0 \in \mathbf{Z}[0, 2^{m_0}),$$

for $1 \leqslant \ell \leqslant k-1$, b_ℓ is a function

$$b_\ell : \prod_{i=0}^{\ell-1} \mathbf{Z}[0, 2^{m_i}) \to \mathbf{Z}[0, 2^{m_\ell}),$$

where we denoted $m_\ell := m(A_\ell)$, and φ is a function with values in G

$$\varphi : \prod_{\ell=0}^{k-1} \mathbf{Z}[0, 2^{m_\ell}) \to G.$$

The output of A at input $f \in F$ will be a probability measure $A(f)$ on G, defined as follows: First put

$$p_{A,f}(x_0, \ldots, x_{k-1}) = |A_{0,f}(x_0, b_0)|^2 |A_{1,f}(x_1, b_1(x_0))|^2 \ldots |A_{k-1,f}(x_{k-1}, b_{k-1}(x_0, \ldots, x_{k-2}))|^2. \tag{4}$$

Then define $A(f)$ by setting for any subset $C \subseteq G$

$$A(f)(C) = \sum_{\varphi(x_0, \ldots, x_{k-1}) \in C} p_{A,f}(x_0, \ldots, x_{k-1}). \tag{5}$$

By $n_q(A) := \sum_{\ell=0}^{k-1} n_q(A_\ell)$ we denote the number of queries used by A.

Informally, such an algorithm A starts with a fixed basis state b_0 and, at input f, applies in an alternating way unitary transformations U_{0j} (not depending on f) and the operator Q_f of a certain query. After a fixed number of steps the resulting state is measured, which gives a (random) basis state, say ξ_0. This state is memorized and then transformed (e.g., by a classical computation, which is symbolized by b_1) into a new basis state $b_1(\xi_0)$. This is the starting state to which the next sequence of quantum operations is applied (with possibly another query and number of qubits). The resulting state is again measured, which gives the (random) basis state ξ_1. This state is memorized, $b_2(\xi_0, \xi_1)$ is computed (classically), and so on. After k such cycles, we obtain $\xi_0, \dots, \xi_{k-1}$. Then finally an element of G is computed (e.g., again on a classical computer) from the results of all measurements: $\varphi(\xi_0, \dots, \xi_{k-1})$. The probability measure $A(f)$ is its distribution. For details, see [8].

The error of A is defined as follows: Let $0 \leqslant \theta < 1, f \in F$, and let ζ be any random variable with distribution $A(f)$. Then put

$$e(S, A, f, \theta) = \inf\{\varepsilon \mid \mathbf{P}\{\|S(f) - \zeta\| > \varepsilon\} \leqslant \theta\}.$$

Consequently, $e(S, A, f, \theta) \leqslant \varepsilon$ iff the algorithm A computes $S(f)$ with error at most ε and probability at least $1 - \theta$. Associated with this we introduce the error over the class F as

$$e(S, A, F, \theta) = \sup_{f \in F} e(S, A, f, \theta).$$

It is customary to consider these quantities at a fixed error probability level: We denote

$$e(S, A, f) = e(S, A, f, 1/4)$$

and

$$e(S, A, F) = e(S, A, F, 1/4).$$

The choice $\theta = 1/4$ is arbitrary—any fixed $\theta < 1/2$ would do. The nth minimal query error is defined for $n \in \mathbf{N}_0$ as

$$e_n^q(S, F) = \inf\{e(S, A, F) \mid A \text{ is any quantum algorithm with } n_q(A) \leqslant n\}.$$

This is the minimal error which can be reached using at most n queries. The query complexity is defined for $\varepsilon > 0$ by

$$\begin{aligned} &\mathrm{comp}_\varepsilon^q(S, F) \\ &\quad = \min\{n_q(A) \mid A \text{ is any quantum algorithm with } e(S, A, F) \leqslant \varepsilon\}. \end{aligned}$$

The quantities $e_n^q(S, F)$ and $\mathrm{comp}_\varepsilon^q(S, F)$ are inverse to each other in the following sense: For all $n \in \mathbf{N}_0$ and $\varepsilon > 0$, $e_n^q(S, F) \leqslant \varepsilon$ if and only if $\mathrm{comp}_{\varepsilon_1}^q(S, F) \leqslant n$ for all $\varepsilon_1 > \varepsilon$. Thus, determining the query complexity is equivalent to determining the nth minimal error. Henceforth, we will deal only with $e_n^q(S, F)$.

3. The main result

Let $N \in \mathbf{N}$ and set $D = \mathbf{Z}[0, N)$, $K = \mathbf{R}$, $G = \mathbf{R}$. For $1 \leqslant p \leqslant \infty$ let L_p^N denote the space of all functions $f : D \to \mathbf{R}$, equipped with the norm

$$||f||_{L_p^N} = \left(\frac{1}{N} \sum_{i=0}^{N-1} |f(i)|^p \right)^{1/p}$$

if $p < \infty$ and

$$||f||_{L_\infty^N} = \max_{0 \leqslant i \leqslant N-1} |f(i)|.$$

Define $S_N : L_p^N \to \mathbf{R}$ by

$$S_N f = \frac{1}{N} \sum_{i=0}^{N-1} f(i)$$

and let

$$F = \mathscr{B}_p^N := \{ f \in L_p^N \mid ||f||_{L_p^N} \leqslant 1 \}.$$

Let us summarize the known results about the order of $e_n^q(S_N, \mathscr{B}_p^N)$ (and thus the query complexity of computing the mean of p-summable sequences) in Theorem 1. The case $p = \infty$ is due to Grover [6], Brassard et al. [3] (upper bounds) and Nayak and Wu [11] (lower bounds). The results in the case $1 \leqslant p < \infty$ are due to Heinrich [8]. Note that throughout the paper we often use the same symbols for possibly different constants. Also, log always means $\log_2$.

Theorem 1. *Let $1 \leqslant p \leqslant \infty$. There are constants $c_0, c_1, c_2, c_3 > 0$ such that for all $n, N \in \mathbf{N}$ with $2 < n \leqslant c_1 N$,*

$$c_2 n^{-1} \leqslant e_n^q(S_N, \mathscr{B}_p^N) \leqslant c_3 n^{-1} \quad \text{if } 2 < p \leqslant \infty,$$

$$c_2 n^{-1} \leqslant e_n^q(S_N, \mathscr{B}_2^N) \leqslant c_3 n^{-1} \log^{3/2} n \log \log n$$

and

$$c_2 n^{-2(1-1/p)} \leqslant e_n^q(S_N, \mathscr{B}_p^N) \leqslant c_3 n^{-2(1-1/p)} \quad \text{if } 1 \leqslant p < 2, \ n \leqslant c_0 \sqrt{N}.$$

The case $1 \leqslant p < 2$, $n \geqslant c_0 \sqrt{N}$ was left open. We will settle it here by proving

Theorem 2. *Let $1 \leqslant p < 2$. There are constants $c_0, c_1, c_2, c_3 > 0$ such that for all $n, N \in \mathbf{N}$ with $c_0 \sqrt{N} \leqslant n \leqslant c_1 N$,*

$$c_2 n^{-2/p} N^{2/p-1} \leqslant e_n^q(S_N, \mathscr{B}_p^N) \leqslant c_3 n^{-2/p} N^{2/p-1} \max(\log(n/\sqrt{N}), 1)^{2/p-1}.$$

It is interesting to mention the consequences for the case $p = 1$ separately:

Corollary 1. *There are constants* $c_1, c_2, c_3 > 0$ *such that*

$$c_2 \leqslant e_n^q(S_N, \mathscr{B}_1^N) \leqslant 1$$

if $0 \leqslant n < \sqrt{N}$, *and*

$$c_2 n^{-2} N \leqslant e_n^q(S_N, \mathscr{B}_1^N) \leqslant c_3 n^{-2} N \max(\log(n/\sqrt{N}), 1)$$

if $\sqrt{N} \leqslant n \leqslant c_1 N$.

Hence for $p = 1$ the decay essentially starts only beyond $\sqrt{N}$. Note that the corresponding quantities for the classical deterministic and randomized setting remain $\Omega(1)$ also in the range $\sqrt{N} \leqslant n \leqslant c_1 N$, see [10].

Combining the theorem above with the respective result in Theorem 1, we can cover the full range $n \leqslant c_1 N$. This result is a direct consequence of Theorems 1 and 2 and the monotonicity of $e_n^q(S_N, \mathscr{B}_p^N)$ in n.

Corollary 2. *Let* $1 \leqslant p < 2$. *There are constants* $c_1, c_2, c_3 > 0$ *such that for all* $n, N \in \mathbf{N}$ *with* $n \leqslant c_1 N$,

$$\begin{aligned} & c_2 \min(n^{-2(1-1/p)}, n^{-2/p} N^{2/p-1}) \\ & \quad \leqslant e_n^q(S_N, \mathscr{B}_p^N) \\ & \quad \leqslant c_3 \min(n^{-2(1-1/p)}, n^{-2/p} N^{2/p-1}) \max(\log(n/\sqrt{N}), 1)^{2/p-1}. \end{aligned}$$

The following two sections contain the proof of Theorem 2.

4. Upper bounds

For any $M \in \mathbf{N}$ we define

$$S_{N,M} f = \frac{1}{N} \sum_{i \in \mathbf{Z}[0,N), |f(i)| < M} f(i)$$

and

$$S'_{N,M} f = S_N f - S_{N,M} f = \frac{1}{N} \sum_{i \in \mathbf{Z}[0,N), |f(i)| \geqslant M} f(i).$$

Proposition 1. *Let* $1 \leqslant p < \infty$. *Then there is a constant* $c > 0$ *such that for all* $n, M, N \in \mathbf{N}$ *with*

$$n \geqslant c M^{-p/2} N \max(\log(M^{-p} N), 1),$$

we have

$$e_n^q(S'_{N,M}, \mathscr{B}_p^N) = 0.$$

Proof. We may assume that

$$M^p \leqslant N, \tag{6}$$

because otherwise $S'_{N,M}f = 0$ for all $f \in \mathscr{B}_p^N$, so $e_0^q(S'_{N,M}) = 0$. Let

$$m' = \lceil \log N \rceil. \tag{7}$$

We define a quantum algorithm A_0 from $\mathscr{B}_p^N$ to $\mathbf{Z}[0, 2^{m'}) \times \mathbf{R}$. To specify its quantum query, fix any $m'' > m' + 1$ and define the mapping $\beta : \mathbf{R} \to \mathbf{Z}[0, 2^{m''})$ by setting for $z \in \mathbf{R}$

$$\beta(z) = \begin{cases} 2^{m''-1} & \text{if } |z| < M, \\ \lfloor 2^{m''-m'-1}(z + 2^{m'}) \rfloor & \text{if } M \leqslant |z| < 2^{m'}, \\ 2^{m''} - 1 & \text{if } z \geqslant 2^{m'}, \\ 0 & \text{if } z \leqslant -2^{m'}. \end{cases}$$

It follows that for $M \leqslant |z| \leqslant 2^{m'}$,

$$-2^{m'} + 2^{-m''+m'+1}\beta(z) \leqslant z \leqslant -2^{m'} + 2^{-m''+m'+1}(\beta(z) + 1) \tag{8}$$

and

$$\beta(z) = 2^{m''-1} \text{ if and only if } |z| < M. \tag{9}$$

In connection with this definition let us mention that for $f \in \mathscr{B}_p^N$,

$$|f(i)| \leqslant N^{1/p} \leqslant N \leqslant 2^{m'} \quad (i = 0, \dots, N-1). \tag{10}$$

Put $Z = \mathbf{Z}[0, N)$, let $\tau : Z \to \mathbf{Z}[0, 2^{m'})$ be the identical embedding, $m = m' + m''$, and define the query by

$$Q = (m, m', m'', Z, \tau, \beta). \tag{11}$$

Let $H_m = H_{m'} \otimes H_{m''}$, and let

$$|i\rangle|x\rangle \quad (i \in Z[0, 2^{m'}), x \in Z[0, 2^{m''}))$$

be the respective representation of basis states. First we consider the simple case $n \geqslant N$, that is, we show

$$e_N^q(S'_{N,M}, \mathscr{B}_p^N) = 0. \tag{12}$$

Indeed, in this case we let the algorithm A_0 start in the classical state $b_0 = |0\rangle|0\rangle$. One application of the query maps this to $|0\rangle|\beta(f(0))\rangle$. Next we measure, from which we obtain $\beta(f(0))$. Now we start the next cycle with $b_1 = |1\rangle|0\rangle$ and obtain, after another query call and measurement, the value $\beta(f(1))$, etc. (That is, formally we work in the quantum model, but, in fact, we stay on the classical states only.) Finally, an appropriate classical computation φ produces from $(\beta(f(i)))_{i=0}^{N-1}$ a suitable approximation to $S'_{N,M}f$ (taking into account (8) and (9)). This proves (12).

Now we assume $n<N$. It follows that, modifying c, if necessary, it suffices to prove the result for

$$M \geqslant M_0, \tag{13}$$

where $M_0 > 0$ is a constant, which will be specified later on.

Let us explain the idea of the following algorithm. It is based on Grover's search algorithm [5], which for an unknown subset of our index set $\mathbf{Z}[0,N)$ (accessible just by a suitable use of the quantum query) allows to produce an element of this subset, with high probability. We use this procedure repeatedly to find all i with $|f(i)| \geqslant M$ and the respective, $\beta(f(i))$. Having accomplished this, it remains to compute an approximation to $S'_{N,M}$ classically. Let us now turn to the details.

Let $W_0 \in \mathscr{U}(H_{m'})$ be the Walsh–Hadamard transform, and let $X_0 \in \mathscr{U}(H_{m'})$ be defined by

$$X_0|i\rangle = \begin{cases} -|i\rangle & \text{if } i=0, \\ |i\rangle & \text{otherwise.} \end{cases}$$

Consider the following unitary transforms on H_m, defined by:

$$W|i\rangle|x\rangle = (W_0|i\rangle)|x\rangle,$$

$$X|i\rangle|x\rangle = (X_0|i\rangle)|x\rangle,$$

$$T|i\rangle|x\rangle = \begin{cases} |i\rangle|x\rangle & \text{if } i \in Z \text{ and } x \neq 2^{m''-1}, \\ -|i\rangle|x\rangle & \text{otherwise,} \end{cases}$$

$$J|i\rangle|x\rangle = |i\rangle|\ominus x\rangle.$$

Here $\ominus x$ stands for $(2^{m''}-x) \bmod 2^{m''}$. Note that $W_0^{-1} = W_0$, and hence $W^{-1} = W$. For $f \in \mathscr{B}_p^N$ put

$$Y_f = WXWQ_fJTQ_f. \tag{14}$$

Denote

$$D_f = \{i \mid i \in Z, |f(i)| \geqslant M\}.$$

It follows from the definitions above and from (9) that

$$Q_fJTQ_f|i\rangle|0\rangle = \begin{cases} |i\rangle|0\rangle & \text{if } i \in D_f, \\ -|i\rangle|0\rangle & \text{otherwise,} \end{cases}$$

where Q is as defined in (11) above. A_0 will be an algorithm with one measurement. We define its unitary transform as

$$Q_f Y_f^L W, \tag{15}$$

where $L \in \mathbf{N}$ will be specified later. The starting state will be $|b_0\rangle = |0\rangle|0\rangle$, and the mapping $\varphi : \mathbf{Z}[0,2^{m'}) \times \mathbf{Z}[0,2^{m''}) \to \mathbf{Z}[0,2^{m'}) \times \mathbf{R}$ will be given by

$$\varphi(i,x) = (i, -2^{m'} + 2^{-m''+m'+1}x). \tag{16}$$

This completes the definition of algorithm A_0. Clearly, Y_f is the Grover iterate for the set D_f, and the whole algorithm is Grover's search algorithm [5], or amplitude amplification, in the terminology of Brassard et al. [3], with respect to the $H_{m'}$ component, followed by one more query Q_f. Observe that by (8) and (10) each run of the algorithm A_0 produces a pair $(i, y) \in \mathbf{Z}[0, 2^{m'}) \times \mathbf{R}$ with

$$y \leqslant f(i) \leqslant y + 2^{-m''+m'+1} \quad \text{if } i \in D_f \tag{17}$$

and

$$y = 0 \text{ if and only if } i < N \text{ and } i \notin D_f. \tag{18}$$

The final algorithm A is defined as $\psi(A_0^{L^*})$, which means that we repeat A_0 L^* times and compose the outputs by the mapping

$$\psi : (\mathbf{Z}[0, 2^{m'}) \times \mathbf{R})^{L^*} \to \mathbf{R},$$

see [8], Section 2, for a formal definition. The number $L^* \in \mathbf{N}$ will be specified later. The mapping ψ is defined as follows: Let

$$(i_\ell, y_\ell)_{\ell=0}^{L^*-1} \in (\mathbf{Z}[0, 2^{m'}) \times \mathbf{R})^{L^*}$$

be the outputs of the L^* runs of A_0. We exclude all pairs with $i_\ell \notin D_f$ (which amounts to checking if $i \geqslant N$ or $y = 0$, by (18)), as well as all repetitions of any $i_\ell \in D_f$ (by a suitable sorting algorithm). For the remaining set we add the second components and divide by N (if the remaining set is empty, we output 0).

Now we show that with a suitable choice of the parameters m'', L, L^*, the algorithm outputs $S'_{N,M}f$ with error at most $2^{-m''+m'+1}$ with probability at least 3/4. This follows from (17) if we prove that with probability at least 3/4 the set of remaining indices equals D_f. If $D_f = \emptyset$, this is trivial, so we assume $D_f \neq \emptyset$. First we analyze A_0. Denote $\mu_f = |D_f|$, hence $\mu_f \geqslant 1$, and let $0 < \theta_f \leqslant \pi/2$ be defined by

$$\sin^2 \theta_f = 2^{-m'} \mu_f. \tag{19}$$

Finally, let

$$|\psi_{f,1}\rangle = 2^{-m'/2} \sum_{i \in D_f} |i\rangle$$

and

$$|\psi_{f,0}\rangle = 2^{-m'/2} \sum_{i \in \mathbf{Z}[0,2^{m'}) \setminus D_f} |i\rangle.$$

By the analysis of Brassard et al. [3] and relation (8),

$$\begin{aligned} Y_f^L W|0\rangle|0\rangle = {} & (2^{-m'}\mu_f)^{-1/2} \sin((2L+1)\theta_f)|\psi_{f,1}\rangle|0\rangle \\ & + (1 - 2^{-m'}\mu_f)^{-1/2} \cos((2L+1)\theta_f)|\psi_{f,0}\rangle|0\rangle \end{aligned}$$

(where the second term is replaced by 0 if $\mu_f = 2^{m'}$). It follows that for any $i_0 \in D_f$, the algorithm A_0 outputs $(i_0, \beta(f(i_0)))$ with probability

$$\varrho_{i_0} = \mu_f^{-1} \sin^2((2L+1)\theta_f). \tag{20}$$

In the sequel, we use the elementary relation

$$2x/\pi \leqslant \sin x \leqslant x \quad (x \in [0, \pi/2]). \tag{21}$$

Since $f \in \mathscr{B}_p^N$, we have

$$N^{-1} M^p |D_f| \leqslant 1,$$

hence

$$\mu_f = |D_f| \leqslant M^{-p} N \tag{22}$$

and

$$2^{-m'} \mu_f \leqslant M^{-p} N 2^{-m'} \leqslant M^{-p}.$$

Therefore, by (21) and (19)

$$4\pi^{-2}\theta_f^2 \leqslant M^{-p}$$

and hence

$$\theta_f \leqslant 2^{-1} \pi M^{-p/2}. \tag{23}$$

Now we put

$$M_0 = \lceil 6^{2/p} \rceil \tag{24}$$

and define L by

$$L = \lfloor 3^{-1} M^{p/2} \rfloor. \tag{25}$$

Since we assumed $M \geqslant M_0$, we get from (24) and (25),

$$1 \leqslant \tfrac{1}{6} M^{p/2} \leqslant L \leqslant \tfrac{1}{3} M^{p/2}. \tag{26}$$

It follows from (23) and (26) that

$$(2L+1)\theta_f \leqslant 3L\theta_f \leqslant \pi/2. \tag{27}$$

On the other hand, by (26) and (19),

$$(2L+1)\theta_f > 2L\theta_f \geqslant \tfrac{1}{3} M^{p/2} \sin\theta_f = \tfrac{1}{3} M^{p/2} (2^{-m'} \mu_f)^{1/2}.$$

From (20), (21), (27) and the relation above,

$$\begin{aligned} \varrho_{i_0} &\geqslant \frac{4}{\pi^2} \mu_f^{-1} (2L+1)^2 \theta_f^2 \\ &\geqslant \frac{4}{9\pi^2} M^p 2^{-m'} \\ &\geqslant \frac{2}{9\pi^2} M^p N^{-1} = c_2 M^p N^{-1}, \end{aligned}$$

where in the last line we used (7) and set $c_2 = 2/(9\pi^2)$. It follows that after L^* repetitions of algorithm A_0 the probability of $(i_0, \beta(f(i_0)))$ not being among the results is less than or equal to

$$(1 - c_2 M^p N^{-1})^{L^*} \leqslant e^{-c_2 M^p N^{-1} L^*},$$

where we used that $1 + x \leqslant e^x$ for $x \in \mathbf{R}$. The probability that at least one $i_0 \in D_f$ is not among the results is less than or equal to

$$\mu_f e^{-c_2 M^p N^{-1} L^*} \leqslant M^{-p} N e^{-c_2 M^p N^{-1} L^*},$$

where we used (22). Now we choose L^* in such a way that this probability is not greater than 1/4. This requires (recall that log means $\log_2$)

$$(c_2 \log e) M^p N^{-1} L^* \geqslant \log(M^{-p} N) + 2,$$

which is satisfied if

$$L^* = \left\lceil \frac{3}{c_2 \log e} M^{-p} N \max(\log(M^{-p} N), 1) \right\rceil.$$

We put $c_3 = 3/(c_2 \log e)$ and observe that the above combined with (6) implies

$$L^* \leqslant (c_3 + 1) M^{-p} N \max(\log(M^{-p} N), 1).$$

Together with (26), this implies that algorithm A makes

$$(2L + 1) L^* \leqslant 3 L L^* \leqslant (c_3 + 1) M^{-p/2} N \max(\log(M^{-p} N), 1)$$

queries to compute $S'_{N,M} f$ up to error $2^{-m'' + m' + 1}$ with probability at least 3/4. Since m'' was arbitrary, the result follows. □

One remark concerning the conclusion of this proposition seems appropriate. The relation $e_n^q(S'_{N,M}, \mathscr{B}_p^N) = 0$ means that there is a sequence of quantum algorithms with error tending to zero, each using at most n quantum queries. Decreasing the error, however, requires increasing the number of qubits (logarithmically, see the comment section at the end of the paper for more details).

Next we express M in terms of n and N:

Corollary 3. *Let $1 \leqslant p < \infty$. There is a constant $c \geqslant 1$ such that for all $n, M, N \in \mathbf{N}$,*

$$e_n^q(S'_{N,M}, \mathscr{B}_p^N) = 0$$

whenever

$$M \geqslant c (N/n)^{2/p} \max(\log(n/\sqrt{N}), 1)^{2/p}.$$

Proof. Let c_0 be the constant from Proposition 1. We put

$$c = \max((2c_0)^{2/p}, 1). \tag{28}$$

Assume

$$M \geqslant c (N/n)^{2/p} \max(\log(n/\sqrt{N}), 1)^{2/p}.$$

It follows that

$$M^{-p/2}N \leqslant c^{-p/2}n/\max(\log(n/\sqrt{N}), 1). \tag{29}$$

Squaring and dividing by N gives

$$M^{-p}N \leqslant c^{-p}n^2N^{-1}/\max(\log(n/\sqrt{N}), 1)^2,$$

and hence

$$\begin{aligned}
&\max(\log(M^{-p}N), 1)\\
&\quad\leqslant \max(\log(c^{-p}) + 2\log(n/\sqrt{N}) - 2\log(\max(\log(n/\sqrt{N}), 1)), 1)\\
&\quad\leqslant 2\max(\log(n/\sqrt{N}), 1).
\end{aligned} \tag{30}$$

Relations (28)–(30) give

$$c_0 M^{-p/2}N\max(\log(M^{-p}N), 1) \leqslant 2c_0c^{-p/2}n \leqslant n,$$

which, by Proposition 1, implies

$$e_n^q(S'_{N,M}, \mathscr{B}_p^N) = 0. \qquad \square$$

Proposition 2. *Let* $1 \leqslant p < 2$. *There is a constant* $c > 0$ *such that for all* $k, n, N \in \mathbf{N}$,

$$e_n^q(S_{N,2^k}, \mathscr{B}_p^N) \leqslant c(2^{(1-p/2)k}n^{-1} + 2^k n^{-2}).$$

Proof. This is a direct consequence of the method of proof of Theorem 1 in [8]. The idea is to split the sum into dyadic levels, so that each level corresponds to a suitably scaled summation for the case $p = \infty$. This allows to apply a modification of the counting algorithm of Brassard et al. [3] to each level. A proper balancing over the levels leads to the desired error estimate. For the sake of completeness, we recall some key steps.

Since trivially $e_n^q(S_{N,2^k}, \mathscr{B}_p^N) \leqslant 1$ for all $n \in \mathbf{N}_0$ (just use the zero algorithm), it suffices to prove the result under the assumption

$$n \geqslant 2^{(1-p/2)k}. \tag{31}$$

Define $S_N^{\ell,\sigma} : L_p^N \to \mathbf{R}$ for $\ell = 0, \ldots, k, \sigma = 0, 1$ as

$$S_N^{\ell,\sigma}f = (-1)^\sigma 2^{-\ell}N^{-1} \sum_{2^{\ell-1} \leqslant (-1)^\sigma f(i) < 2^\ell} f(i)$$

if $\ell \geqslant 1$ and

$$S_N^{0,\sigma}f = (-1)^\sigma N^{-1} \sum_{0 \leqslant (-1)^\sigma f(i) < 1} f(i).$$

It is shown in [8] (based on the counting algorithm of Brassard et al. [3]), that there is a constant $c > 0$ such that for each choice of $\nu_\ell, n_\ell \in \mathbf{N}$ $(\ell = 0, \ldots, k)$, there are algorithms $A_{\ell,\sigma}$ $(\ell = 0, \ldots, k,\ \sigma = 0, 1)$ with $n_q(A_{\ell,\sigma}) \leqslant \nu_\ell n_\ell$ and

$$e(S_N^{\ell,\sigma}, A_{\ell,\sigma}, \mathscr{B}_p^N, 2^{-\nu_\ell}) \leqslant c(2^{-p\ell/2}n_\ell^{-1} + n_\ell^{-2})$$

(use the relation following (27) in [8], together with (21) and (22) of that paper). Now choose

$$n_\ell = \lceil 2^{-(1/2-p/4)(k-\ell)} n \rceil$$

and

$$v_\ell = \lceil 2 \log(k-\ell+1) \rceil + 4.$$

Due to (31),

$$n_\ell < 2^{-(1/2-p/4)(k-\ell)+1} n. \tag{32}$$

Let the algorithm A be defined by

$$A = \sum_{0 \leqslant \ell \leqslant k \sigma=0,1} (-1)^\sigma 2^\ell A_{\ell,\sigma}.$$

(We refer again to Heinrich [8, Section 2], for a formal definition.) Taking into account (32), it follows that

$$n_q(A) \leqslant 2 \sum_{\ell=0}^{k} (\lceil 2 \log(k-\ell+1) \rceil + 4) \lceil 2^{-(1/2-p/4)(k-\ell)} n \rceil \leqslant c_1 n. \tag{33}$$

Moreover, since

$$2 \sum_{\ell=0}^{k} 2^{-v_\ell} \leqslant \frac{1}{8} \sum_{\ell=0}^{k} (k-\ell+1)^{-2} < \frac{1}{4},$$

we get

$$\begin{aligned} & e(S_{N,2^k}, A, \mathscr{B}_p^N) \\ & \quad \leqslant c \sum_{\ell=0}^{k} (2^{(1-p/2)\ell + (1/2-p/4)(k-\ell)} n^{-1} + 2^{\ell + (1-p/2)(k-\ell)} n^{-2}) \\ & \quad \leqslant c \sum_{\ell=0}^{k} (2^{(1/2-p/4)(k+\ell)} n^{-1} + 2^{k-p(k-\ell)/2} n^{-2}) \\ & \quad \leqslant c_2 (2^{(1-p/2)k} n^{-1} + 2^k n^{-2}) \end{aligned}$$

which together with (33) and a suitable scaling of n implies the desired result. □

Theorem 3. *Let $1 \leqslant p < 2$. There are constants $c_0, c > 0$ such that for all $n, N \in \mathbf{N}$ with $n \geqslant c_0 \sqrt{N}$*

$$e_n^q(S_N, \mathscr{B}_p^N) \leqslant c n^{-2/p} N^{2/p-1} \max(\log(n/\sqrt{N}), 1)^{2/p-1}.$$

Proof. The key idea is as follows: We choose a suitable k so that computing S_N reduces to computing $S_{N,2^k}$, by the multilevel splitting of Proposition 2, and $S'_{N,2^k}$, by the search procedure from Proposition 1.

First note that

$$e_N^q(S_N, \mathscr{B}_p^N) = 0. \tag{34}$$

Next, observe that it follows readily from Lemma 3 in [8] (reducing the error probability by repeating the algorithm and computing the median) that there is a constant $c_0 \in \mathbf{N}$ such that for all $n, k, N \in \mathbf{N}$,

$$e_{c_0 n}^q(S_N, \mathscr{B}_p^N) \leqslant e_n^q(S_{N,2^k}, \mathscr{B}_p^N) + e_n^q(S'_{N,2^k}, \mathscr{B}_p^N). \tag{35}$$

Now let n satisfy

$$\sqrt{N} \leqslant n < N \tag{36}$$

and choose $k \in \mathbf{N}$ in such a way that

$$2^{k-1} < c_1 (N/n)^{2/p} \max(\log(n/\sqrt{N}), 1)^{2/p} \leqslant 2^k,$$

where $c_1 \geqslant 1$ is the constant from Corollary 3. Consequently, we have

$$e_n^q(S'_{N,2^k}, \mathscr{B}_p^N) = 0. \tag{37}$$

Moreover, with c_2 being the constant from Proposition 2,

$$\begin{aligned} & e_n^q(S_{N,2^k}, \mathscr{B}_p^N) \\ & \quad \leqslant c_2 (2^{(1-p/2)k} n^{-1} + 2^k n^{-2}) \\ & \quad \leqslant c_3 \left((N/n)^{\frac{2}{p}(1-p/2)} n^{-1} \max\left(\log \frac{n}{\sqrt{N}}, 1 \right)^{\frac{2}{p}(1-p/2)} \right. \\ & \qquad \left. + (N/n)^{2/p} n^{-2} \max\left(\log \frac{n}{\sqrt{N}}, 1 \right)^{2/p} \right) \\ & \quad = c_3 \left(N^{2/p-1} n^{-2/p} \max\left(\log \frac{n}{\sqrt{N}}, 1 \right)^{2/p-1} \right. \\ & \qquad \left. + N^{2/p} n^{-2/p-2} \max\left(\log \frac{n}{\sqrt{N}}, 1 \right)^{2/p} \right). \end{aligned} \tag{38}$$

Using (again) $x \geqslant \ln(1+x)$ for $x > -1$, we have

$$\frac{n^2}{N} \geqslant \ln\left(\frac{n^2}{N} + 1 \right) \geqslant 2 \ln \frac{n}{\sqrt{N}} = \frac{2}{\log e} \log \frac{n}{\sqrt{N}} > \log \frac{n}{\sqrt{N}}.$$

Consequently, recalling our assumption $n \geqslant \sqrt{N}$, we get

$$\frac{n^2}{N} \geqslant \max\left(\log \frac{n}{\sqrt{N}}, 1 \right),$$

and therefore

$$N^{2/p-1} n^{-2/p} \max\left(\log \frac{n}{\sqrt{N}}, 1 \right)^{2/p-1} \geqslant N^{2/p} n^{-2/p-2} \max\left(\log \frac{n}{\sqrt{N}}, 1 \right)^{2/p}.$$

From (35), (37), (38), and the relation above we get

$$\begin{aligned} e^q_{c_0 n}(S_N, \mathscr{B}^N_p) &\leqslant e^q_n(S_{N,2^k}, \mathscr{B}^N_p) \\ &\leqslant 2c_3 N^{2/p-1} n^{-2/p} \max\left(\log \frac{n}{\sqrt{N}}, 1\right)^{2/p-1} \end{aligned} \tag{39}$$

for all n with $\sqrt{N} \leqslant n < N$. With a suitable scaling of n, the result follows from (39) and (34). □

5. Lower bounds

We need some general results from Section 4 of Heinrich [8], Let D and K be nonempty sets, let $L \in \mathbf{N}$, and let to each $u = (u_0, \dots, u_{L-1}) \in \{0,1\}^L$ an $f_u \in \mathscr{F}(D, K)$ be assigned such that the following is satisfied:

Condition (I): For each $t \in D$ there is an ℓ, $0 \leqslant \ell \leqslant L-1$, such that $f_u(t)$ depends only on u_ℓ, in other words, for $u, u' \in \{0,1\}^L$, $u_\ell = u_{\ell'}$ implies $f_u(t) = f_{u'}(t)$.

Define the function $\varrho(L, \ell, \ell')$ for $L \in \mathbf{N}$, $0 \leqslant \ell \neq \ell' \leqslant L$ by

$$\varrho(L, \ell, \ell') = \sqrt{\frac{L}{|\ell - \ell'|}} + \frac{\min_{j=\ell,\ell'} \sqrt{j(L-j)}}{|\ell - \ell'|}. \tag{40}$$

The following was proved in [8], using the polynomial method of Beals et al. [2] and based on a result of Nayak and Wu [11]:

Lemma 1. *There is a constant $c_0 > 0$ such that the following holds: Let D, K be nonempty sets, let $F \subseteq \mathscr{F}(D, K)$ be a set of functions, G a normed space, $S : F \to G$ a function, and $L \in \mathbf{N}$. Suppose $(f_u)_{u \in \{0,1\}^L} \subseteq \mathscr{F}(D, K)$ is a system of functions satisfying condition* (I). *Let finally $0 \leqslant \ell \neq \ell' \leqslant L$ and assume that*

$$f_u \in F \text{ whenever } |u| \in \{\ell, \ell'\}. \tag{41}$$

Then

$$e^q_n(S, F) \geqslant \tfrac{1}{2} \min\{\|S(f_u) - S(f_{u'})\| \,|\, |u| = \ell, |u'| = \ell'\} \tag{42}$$

for all n with

$$n \leqslant c_0 \varrho(L, \ell, \ell'). \tag{43}$$

The next result contains lower bounds matching the upper ones from Theorem 3 up to a logarithmic factor.

Theorem 4. *Let $1 \leqslant p < 2$. Then there are constants $c_0, c_1, c_2 > 0$ such that for all $n, N \in \mathbf{N}$ with $c_0 \sqrt{N} \leqslant n \leqslant c_1 N$,*

$$e^q_n(S_N, \mathscr{B}^N_p) \geqslant c_2 n^{-2/p} N^{2/p-1}.$$

Proof. Let c_0 be the constant from Lemma 1, and let

$$c_1 = c_0/\sqrt{12}. \tag{44}$$

By assumption,

$$c_0\sqrt{N} \leqslant n \leqslant c_1 N. \tag{45}$$

We set

$$L = N, \quad \ell = \lceil 2c_0^{-2}n^2N^{-1} \rceil, \quad \ell' = \ell + 1. \tag{46}$$

It follows from (45) that $\ell \geqslant 2$. Moreover, from (46),

$$n \leqslant c_0\sqrt{\ell N/2} \tag{47}$$

and, taking into account that $\ell \geqslant 2$,

$$\ell/2 \leqslant \ell - 1 < 2c_0^{-2}n^2N^{-1},$$

hence, by (44) and (45),

$$\ell + 1 \leqslant 3\ell/2 < 6c_0^{-2}n^2N^{-1} \leqslant 6c_0^{-2}c_1^2N = N/2. \tag{48}$$

We have, by (46)–(48).

$$n \leqslant c_0\sqrt{\ell N/2} \leqslant c_0 \min_{j=\ell,\ell+1} \sqrt{j(N-j)} \leqslant c_0\varrho(L, \ell, \ell'). \tag{49}$$

Now we define $\psi_j \in L_p^N$ $(j = 0, \ldots, L-1)$ as

$$\psi_j(i) = \begin{cases} (\ell+1)^{-1/p}N^{1/p} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

We have

$$S_N\psi_j = (\ell+1)^{-1/p}N^{1/p-1}.$$

For each $u = (u_0, \ldots, u_{L-1}) \in \{0,1\}^L$ define

$$f_u = \sum_{j=0}^{L-1} u_j\psi_j. \tag{50}$$

Since the functions ψ_j have disjoint supports, the system $(f_u)_{u\in\{0,1\}^L}$ satisfies condition (I). Moreover, $f_u \in \mathscr{B}_p^N$ whenever $|u| = \ell, \ell+1$. Lemma 1, relation (49) and the left and middle part of (48) give

$$\begin{aligned} e_n^q(S_N, \mathscr{B}_p^N) &\geqslant \tfrac{1}{2}\min\{|S_Nf_u - S_Nf_{u'}| \,|\, |u| = \ell, |u'| = \ell+1\} \\ &= \tfrac{1}{2}(\ell+1)^{-1/p}N^{1/p-1} \geqslant \tfrac{1}{2}(6c_0^{-2}n^2N^{-1})^{-1/p}N^{1/p-1} \\ &= \frac{c_0^{2/p}}{2\cdot 6^{1/p}}n^{-2/p}N^{2/p-1}. \qquad \square \end{aligned}$$

6. Comments

Let us first mention that there remains another gap in the order of the quantity $e_n^q(S_N, \mathscr{B}_p^N)$ in all the results of Theorems 1, 2, and Corollaries 1, 2, namely, the region $c_1 N \leqslant n < N$. As we mentioned before, we have $e_n^q(S_N, \mathscr{B}_p^N) = 0$ for $n \geqslant N$ (classical computation of the sum). Hence filling this gap means determining how fast $e_n^q(S_N, \mathscr{B}_p^N)$ goes to zero in the region close to classical computation. We did not consider this problem further. It is theoretically interesting, but one should also mention that its solution would not say much about the speed-up due to quantum computation: With an effort, just by a constant factor higher, the problem can be solved with the same error (in fact, even up to any needed precision) by classical computation.

Finally, we discuss the cost of our algorithm in the bit model of computation. Here we assume that both N and n are powers of two. The algorithm behind Proposition 1 and Corollary 3 needs $\mathcal{O}(nm'')$ quantum gates (see [12, Chapter 4], for basics on quantum gates), $\mathcal{O}(m'')$ qubits, and makes $\mathcal{O}(n^2 N^{-1}/\max(\log(n/\sqrt{N}), 1))$ measurements to reach error $\mathcal{O}(2^{\log N - m''})$. The bit cost of the classical computations is negligible as compared to the number of quantum gates: We need $\mathcal{O}(n^2 N^{-1} m'')$ classical bit operations to sort out the wrong elements and to add the right ones. The bit cost of the algorithm in connection with Proposition 2 was already analyzed in [8]. It amounts to $\mathcal{O}(n \log N)$ quantum gates, $\mathcal{O}(\log N)$ qubits, and $\mathcal{O}(k \log k)$ (which is $\mathcal{O}(\log n \log\log n)$) measurements. The number of classical bit operations is $\mathcal{O}(\log n \log\log n \log N)$, and thus, again dominated by the number of quantum gates. Summarizing this for the algorithm of Theorem 3, we see that we can implement it with $\mathcal{O}(n \log N)$ quantum gates, on $\mathcal{O}(\log N)$ qubits, and with

$$\mathcal{O}(n^2 N^{-1}/\max(\log(n/\sqrt{N}), 1) + \log(N/n) \log\log(N/n))$$

measurements. Thus the quantum bit cost differs by at most a logarithmic factor from the quantum query complexity.

References

[1] D.S. Abrams, C.P. Williams, Fast quantum algorithms for numerical integrals and stochastic processes, Technical Report, 1999, http://arXiv.org/abs/quant-ph/9908083.

[2] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf, Quantum lower bounds by polynomials, Proceedings of 39th IEEE FOCS, 1998, pp. 352–361, see also http://arXiv.org/abs/quant-ph/9802049.

[3] G. Brassard, P. Høyer, M. Mosca, A. Tapp, Quantum amplitude amplification and estimation, Technical Report, 2000, http://arXiv.org/abs/quant-ph/0005055.

[4] A. Ekert, P. Hayden, H. Inamori, Basic concepts in quantum computation, 2000, see http://arXiv.org/abs/quant-ph/0011013.

[5] L. Grover, A fast quantum mechanical algorithm for database search, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, ACM Press, New York, 1996, pp. 212–219, see also http://arXiv.org/abs/quant-ph/9605043.

[6] L. Grover, A framework for fast quantum mechanical algorithms, Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, ACM Press, New York, 1998, pp. 53–62, see also http://arXiv.org/abs/quant-ph/9711043.

[7] J. Gruska, Quantum Computing, McGraw-Hill, London, 1999.

[8] S. Heinrich, Quantum summation with an application to integration, J. Complexity 18 (2002) 1–50, see also http://arXiv.org/abs/quant-ph/0105116.

[9] S. Heinrich, Quantum integration in Sobolev classes, J. Complexity 19 (2003) 19–42, see also http://arXiv.org/abs/quant-ph/0112153.

[10] S. Heinrich, E. Novak, Optimal summation and integration by deterministic, randomized, and quantum algorithms, in: K.-T. Fang, F.J. Hickernell, H. Niederreiter (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2000, Springer, Berlin, 2002, pp. 50–62, see also http://arXiv.org/abs/quant-ph/0105114.

[11] A. Nayak, F. Wu, The quantum query complexity of approximating the median and related statistics, STOC, May 1999, pp. 384–393, see also http://arXiv.org/abs/quant-ph/9804066.

[12] M.A. Nielsen, I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, 2000.

[13] E. Novak, Deterministic and Stochastic Error Bounds in Numerical Analysis, Lecture Notes in Mathematics, Vol. 1349, Springer, Berlin, 1988.

[14] E. Novak, The real number model in numerical analysis, J Complexity 11 (1995) 57–73.

[15] E. Novak, Quantum complexity of integration, J. Complexity 17 (2001) 2–16, see also http://arXiv.org/abs/quant-ph/0008124.

[16] A.O. Pittenger, Introduction to Quantum Computing Algorithms, Birkhäuser, Boston, 1999.

[17] P.W. Shor, Introduction to Quantum Algorithms, 2000, see http://arXiv.org/abs/quant-ph/0005003.

[18] J.F. Traub, H. Woźniakowski, Path integration on a quantum computer, 2001, see http://arXiv.org/abs/quant-ph/0109113.

[19] J.F. Traub, G.W. Wasilkowski, H. Woźniakowski, Information-Based Complexity, Academic Press, New York, 1988.