Interner Bericht

Quantum Summation with an Application to Integration

S. Heinrich

312/01

FACHBEREICH
INFORMATIK

UNIVERSITÄT
KAISERSLAUTERN

Postfach 3049 · D-67653 Kaiserslautern

Quantum Summation with an Application to Integration

S. Heinrich

312/01

Fachbereich Informatik
Universität Kaiserslautern
D-67653 Kaiserslautern, Germany
e-mail: heinrich@informatik.uni-kl.de

Juni 2001

Paper submitted to the Journal of Complexity. See also
http://arXiv.org/abs/quant-ph/0105116

Quantum Summation with an Application to Integration

S. Heinrich
Fachbereich Informatik
Universität Kaiserslautern
D-67653 Kaiserslautern, Germany
e-mail: heinrich@informatik.uni-kl.de

Abstract

We study summation of sequences and integration in the quantum model of computation. We develop quantum algorithms for computing the mean of sequences which satisfy a p-summability condition and for integration of functions from Lebesgue spaces $L_p([0,1]^d)$ and analyze their convergence rates. We also prove lower bounds which show that the proposed algorithms are, in many cases, optimal within the setting of quantum computing. This extends recent results of Brassard, Høyer, Mosca, and Tapp (2000) on computing the mean for bounded sequences and complements results of Novak (2001) on integration of functions from Hölder classes.

1 Introduction

Quantum algorithms and complexity are by now well studied for various discrete problems. This includes such milestones as Shor's (1994) factorization and Grover's (1996) search algorithm. Much less is understood about numerical problems, computational problems of analysis. These problems are typically defined on a continuum and/or take values in a continuum, such as the field of real or complex numbers, domains in finite dimensional vector spaces or even infinite dimensional normed spaces like function spaces.

First results related to this direction concern the counting problem (Boyer, Brassard, Høyer, and Tapp, 1998) and the computation of the mean (Grover, 1998, Brassard, Høyer, Mosca, and Tapp, 2000) of finite sequences which satisfy a uniform bound (e.g. whose elements belong to the interval

$[0,1]$). Matching lower bounds were obtained by Nayak and Wu (1999) using the polynomial method of Beals, Buhrman, Cleve, and Mosca (1998). Abrams and Williams (1999) proposed certain quantum algorithms for integration. Novak (2001) was the first to provide quantum integration algorithms with matching upper and lower bounds. He studied an important class of integrands - functions which belong to Hölder spaces. His work is closely related to information-based complexity - a frame in which the complexity of numerical problems is studied (in the classical setting).

Mainly due to efforts within this theory, by now for many important problems of numerical analysis matching upper and lower complexity bounds (or in other words, optimal convergence rates) are known for both the classical deterministic and randomized setting. It is a challenging task to study these problems in the setting of quantum computation. Once such results are obtained, one can compare them to the deterministic and randomized classical ones to understand the possible speedups by quantum algorithms. Novak (2001) did the first step toward this, and the present paper as well as related work, Heinrich and Novak (2001a,b) and Heinrich (2001), go further along this line.

In the present paper we study quantum summation of sequences satisfying p-summability conditions. These classes are larger than that of uniformly bounded sequences (precise definitions are given in section 3) and cannot be handled by the previous algorithms. But the solution of this problem is needed for the understanding of quantum integration in various function spaces (different from Hölder classes) characterized by p-integrability conditions, such as the Lebesgue spaces $L_p([0,1]^d)$, studied here in section 5, and the Sobolev spaces analyzed in Heinrich (2001). In the present paper we therefore develop quantum algorithms for computing the sum of such sequences. We also prove lower bounds which are, in many cases, matching with the obtained upper bounds, showing the optimality of the algorithms. (The picture is completed in Heinrich and Novak 2001b, where the case is settled which is left open here.) These results enable us to completely determine (in one case, up to a logarithmic factor) the optimal order of convergence of quantum integration in Lebesgue spaces $L_p([0,1]^d)$.

Comparing the result both for summation and integration with the randomized classical setting, we observe a considerable gain by quantum computing – the quantum speed of convergence equals the square of the randomized classical one. The gain over deterministic classical algorithms can even be exponential (see the details in sections 5 and 6).

To put the problem formulations and the results on a firm mathematical basis it was necessary to extend the usual model of quantum computation (we follow Beals, Buhrman, Cleve, and Mosca, 1998) to the setting of numerical problems, to the fields of real or complex numbers, normed spaces of functions etc. This extension was widely inspired by the approach of information-based complexity theory to numerical problems in the classical settings and can be viewed, in fact, as a quantum setting of this theory.

The paper is organized as follows. The general approach is presented in section 2. Upper bounds for summation of p-summable sequences and respective algorithms are contained in section 3. General results concerning lower bounds as well as their application to summation are given in section 4. Section 5 is devoted to the application of the previous results to integration of functions from the Lebesgue spaces $L_p([0,1]^d)$. Finally, in section 6 we give comparisons to results in the classical deterministic and randomized settings and comment on some further related issues.

For background reading in quantum computing we refer to the surveys Ekert, Hayden, and Inamori (2000), Shor (2000), and the monographs Pittenger (1999), Gruska (1999) and Nielsen and Chuang (2000). For notions and results in information-based complexity theory see the monographs Traub, Wasilkowski, and Woźniakowski (1988), Novak (1988), and the survey of the randomized setting Heinrich (1993).

2 A General Framework for Numerical Quantum Algorithms

We are given nonempty sets D, K, a nonempty set F of functions on D with values in K and a function S from F to a normed space G. By a normed space we always mean a normed linear space over $\mathbf{K}$, where $\mathbf{K}$ is either $\mathbf{R}$ or $\mathbf{C}$, the field of real or complex numbers. We seek to compute (approximately) $S(f)$ for $f \in F$, where f can only be accessed through its values (that is, we assume that f is given as a black box – given $t \in D$, this black box returns $f(t) \in K$).

This general framework includes, on one hand, the binary case, where $D = \{0, \dots, N-1\}$, $K = \{0,1\}$, F consists of all Boolean functions, i. e. all functions from D to K, and S maps F to $G = \mathbf{R}$ (which contains $\{0,1\}$). On the other hand, in numerical problems, D is usually some subset of $\mathbf{R}^d$, $K = \mathbf{K}$, F is usually a subset of a normed linear space of functions (or tuples of functions) from D to $\mathbf{K}$, and S is a mapping (also called the solution operator) from F to G, where G is either $\mathbf{K}$ or a normed space of functions.

We want to study algorithms and complexity of solving these problems on a quantum computer. For this purpose, we adopt standard notation of quantum computing. Let H_1 be the two-dimensional complex Hilbert space $\mathbf{C}^2$, $\{e_0, e_1\}$ its unit vector basis, let

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

be the Hilbertian tensor product of m copies of H_1. We use the standard identifications such as writing e_i or $|i\rangle$ for $e_{j_0} \otimes \cdots \otimes e_{j_{m-1}}$, where $i = \sum_{k=0}^{m-1} j_k 2^{m-1-k}$ is the binary expansion of i. When identifying H_m with $H_{m_1} \otimes \cdots \otimes H_{m_\ell}$, where $\sum_{k=1}^{\ell} m_j = m$, we also identify e_i with the respective $e_{i_1} \otimes \cdots \otimes e_{i_\ell}$ and $|i_1\rangle \dots |i_\ell\rangle$, and finally also i itself with $(i_1, \dots, i_\ell)$ in the respective way. For convenience we use the following notation:

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

for $N \in \mathbf{N}$ (as usual, we let $\mathbf{N} = 1, 2, \dots, \mathbf{N}_0 = \mathbf{N} \cup \{0\}$). Let $\mathcal{C}_m = \{|i\rangle : i \in \mathbf{Z}[0, 2^m)\}$ be the set of basis vectors of H_m, also called classical states, or basis states, and let $\mathcal{U}(H_m)$ denote the set of unitary operators on H_m.

First we introduce the notion of a quantum query (in our setting of D, K, F, G and S). A quantum query on F is given by a tuple

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

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

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

are arbitrary mappings. The meaning of these components will be explained below. Such a tuple Q defines a query mapping (we use the same symbol Q)

$$Q : F \to \mathcal{U}(H_m)$$
$$f \to Q_f$$

as follows: Let any $h \in \mathcal{C}_m$ be represented as $h = |i\rangle |x\rangle |y\rangle$ with $|i\rangle \in \mathcal{C}_{m'}, |x\rangle \in \mathcal{C}_{m''}, |y\rangle \in \mathcal{C}_{m-m'-m''}$ (if $m = m' + m''$, we drop the last component). Then Q_f is the unitary operator defined uniquely by its action on $\mathcal{C}_m$:

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

where here and in the sequel $\oplus$ always means addition modulo the respective power of 2, here modulo $2^{m''}$. Let $m(Q)$ denote the first component of Q, that is, the total number of qubits. If $m(Q) = m$, we also say that Q is an m-qubit quantum query.

This notion contains the binary black box query typically used in quantum computation (see, e.g. Beals, Buhrman, Cleve, and Mosca, 1998) as a particular case: Such a binary query associates to a $\{0,1\}$-valued function κ defined on $\mathbf{Z}[0, 2^{m'})$ the operator Q_κ which maps $|i\rangle\,|x\rangle\,|y\rangle$ to $|i\rangle\,|x \oplus \kappa(i)\rangle\,|y\rangle$, where $i \in \mathcal{C}_{m'}$ and $x \in \mathcal{C}_1$. In our situation we have to deal with two more general domains: D and K. The mapping $\tau : i \to \tau(i) \in D$ describes the (chosen by the algorithm designer) correspondence of binary strings with certain elements of the domain of definition of functions from F. Since at request $\tau(i)$ the black box returns $f(\tau(i))$, which is an element of K, we need a second mapping β, which maps ("codes") elements from K into binary strings. (This is also chosen by the algorithm designer.) As usual, the untouched part $|y\rangle$ stands for "working bits".

Note that, by the definition, a quantum query on F is also a quantum query on any other nonempty subset $F_1 \subseteq \mathcal{F}(D,K)$, and in particular, on $\mathcal{F}(D,K)$ itself. Here $\mathcal{F}(D,K)$ denotes the set of all functions from D to K. Indeed, the mapping Q_f is defined for each $f \in \mathcal{F}(D,K)$.

Next we define quantum algorithms in the general framework of D, K, F, G and S. It will be convenient for us to introduce algorithms with multiple measurements. We show later in this section how they can be simulated by algorithms with one measurement. Let us first describe informally what we mean by a quantum algorithm with k measurements: Such an algorithm starts with a fixed basis state b_0 and applies in an alternating way unitary transformations (not depending on f) and a certain query, associated to the algorithm. After a fixed number of steps the resulting state is measured, which gives a (random) basis state ξ_0. This state is memorized and then transformed (e.g. by a classical computer) into a new basis state b_1. 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, and b_2 is computed from ξ_0 and ξ_1, and so on. After k such cycles, we obtained $\xi_0, \dots, \xi_{k-1}$. Then finally an element of G – the output of the algorithm – is computed (e.g. again on a classical computer) from the results of all measurements: $\varphi(\xi_0, \dots, \xi_{k-1})$.

Now we formalize this: 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 $U_j \in \mathcal{U}(H_m)$ $(j = 0, \ldots, n)$, with $m = m(Q)$ (in the case $n = 0$, no query Q is needed). Given such an A and $f \in F$, we let $A_f \in \mathcal{U}(H_m)$ 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 used by A. We also introduce the following notation. Let $A_f(x, y)$ for $x, y \in \mathbf{Z}[0, 2^m)$ be given by

$$A_f \left|y\right\rangle = \sum_{x \in \mathbf{Z}[0,2^m)} A_f(x, y) \left|x\right\rangle. \tag{4}$$

Hence $(A_f(x, y))_{x,y}$ is the matrix of the transformation A_f in the canonical basis $\mathcal{C}_m$.

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. To explain the other components, set $m_\ell = m(A_\ell)$. Then

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

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

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

and φ is a function with values in G

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

We also say that A is a quantum algorithm with measurement(s), or just a quantum algorithm.

Let $\mathcal{P}_0(G)$ denote the set of all probability measures on G whose support is a finite set. The output of A at input $f \in F$ will be an element $A(f) \in$

$\mathcal{P}_0(G)$ (we use the same symbol A for the mapping $A : F \to \mathcal{P}_0(G)$). We define $A(f)$ via a sequence of random variables $(\xi_{\ell,f})_{\ell=0}^{k-1}$ (we assume that all random variables are defined over a fixed – suitably large – probability space $(\Omega, \Sigma, \mathbf{P})$). So let $f \in F$ be fixed. Now let $\xi_{\ell,f}$ be such that

$$\mathbf{P}\{\xi_{0,f} = x\} = |A_{0,f}(x, b_0)|^2 \tag{5}$$

and, for $1 \le \ell \le k-1$,

$$\mathbf{P}\{\xi_{\ell,f} = x \,|\, \xi_{0,f} = x_0, \dots, \xi_{\ell-1,f} = x_{\ell-1}\} = |A_{\ell,f}(x, b_\ell(x_0, \dots, x_{\ell-1}))|^2. \tag{6}$$

Clearly, this defines the distribution of $(\xi_{\ell,f})_{\ell=0}^{k-1}$ uniquely. Let us define for $x_0 \in \mathbf{Z}[0, 2^{m_0}), \dots, x_{k-1} \in \mathbf{Z}[0, 2^{m_{k-1}})$

$$\begin{aligned} p_{A,f}(x_0, \dots, x_{k-1}) &= |A_{0,f}(x_0, b_0)|^2 |A_{1,f}(x_1, b_1(x_0))|^2 \dots \\ &\quad \dots |A_{k-1,f}(x_{k-1}, b_{k-1}(x_0, \dots, x_{k-2}))|^2. \end{aligned} \tag{7}$$

It follows from (5) and (6) that

$$\mathbf{P}\{\xi_{0,f} = x_0, \dots, \xi_{k-1,f} = x_{k-1}\} = p_{A,f}(x_0, \dots, x_{k-1}). \tag{8}$$

Finally we define the output of A at input f as

$$A(f) = \operatorname{dist}(\varphi(\xi_{0,f}, \dots, \xi_{k-1,f})),$$

the distribution of $\varphi(\xi_{0,f}, \dots, \xi_{k-1,f})$. This random variable takes only finitely many values in G, hence the support of $A(f)$ is finite (and no measurability problems related to the target space G will arise). It follows from (8) that for any subset $C \subseteq G$

$$A(f)\{C\} = \sum_{\varphi(x_0, \dots, x_{k-1}) \in C} p_{A,f}(x_0, \dots, x_{k-1}). \tag{9}$$

We note that, analogously to quantum queries, a quantum algorithm on F is automatically also a quantum algorithm on any nonempty $F_1 \subseteq \mathcal{F}(D, K)$.

The number $n_q(A) := \sum_{\ell=0}^{k-1} n_q(A_\ell)$ is called the number of queries used by A. This is the crucial quantity for the purposes of our query complexity analysis. (In section 6 we give some comments on the cost in the bit-model.)

Let $0 \le \theta < 1$. For an algorithm A as above we define the (probabilistic) error at $f \in F$ as follows. Let ζ be a random variable with distribution $A(f)$. Then

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

(note that this infimum is always attained). Hence $e(S, A, f, \theta) \leq \varepsilon$ iff the algorithm A computes $S(f)$ with error at most ε and probability at least $1 - \theta$. We put

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

(we allow the value $+\infty$ for this quantity). Furthermore, we set

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

and similarly,

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

The central quantity of our study is the n-th minimal (query) error, defined for $n \in \mathbf{N}_0$ by

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

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

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

(we put $\operatorname{comp}_\varepsilon^q(S, F) = +\infty$ if there is no such algorithm). It is easily checked that these functions are inverse to each other in the following sense: For all $n \in \mathbf{N}_0$ and $\varepsilon > 0$, $e_n^q(S, F) \leq \varepsilon$ if and only if $\operatorname{comp}_{\varepsilon_1}^q(S, F) \leq n$ for all $\varepsilon_1 > \varepsilon$. Hence it suffices to determine one of them. We shall usually choose the first one.

Our first general result shows the tight relation between algorithms with several measurements and (the conceptually simpler) algorithms with one measurement. It states that an algorithm with several measurements can always be represented equivalently by an algorithm with one measurement and twice the number of queries (at the expense of an increased number of qubits).

Lemma 1. *For each quantum algorithm A from F to G with k measurements there is a quantum algorithm $\widetilde{A}$ from F to G with one measurement such that $n_q(\widetilde{A}) = 2n_q(A)$ and*

$$\widetilde{A}(f) = A(f)$$

for all $f \in F$.

Proof. By 'ℓ-th quantum cycle' we mean the quantum operations in the original algorithm before the first measurement if $\ell = 0$, and between the ℓ-th and the $\ell+1$-st measurement if $1 \leq \ell \leq k-1$. The idea of the proof is easy: We simulate the k queries by one query and instead of intermediate measurements we 'store' the results of the cycles in different components until the final measurement (a pseudo-code is given below). Let us now formalize this and check that the corresponding probabilities coincide. Let the original algorithm be given by

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

where

$$A_\ell = (Q_\ell, (U_{\ell j})_{j=0}^{n_\ell}),$$

and

$$Q_\ell = (m_\ell, m'_\ell, m''_\ell, Z_\ell, \tau_\ell, \beta_\ell).$$

By adding, if necessary, qubits, which are set to zero and remain so during the whole ℓ-th cycle we may assume without loss of generality that $m'_\ell \equiv m'$. Let[1] $k_0 = \lceil \log k \rceil$, define $\widetilde{m}' = m' + k_0$, and $\widetilde{Z} \subset \mathbf{Z}[0, 2^{\widetilde{m}'})$ by

$$\widetilde{Z} = \{(\ell, i) \,|\, 0 \leq \ell \leq k-1,\, i \in Z_\ell\}.$$

Now we define

$$\widetilde{\tau} : \widetilde{Z} \to D$$
$$\widetilde{\tau}(\ell, i) = \tau_\ell(i) \quad \text{for} \quad (\ell, i) \in \widetilde{Z}.$$

Moreover, we set

$$\widetilde{m}'' = \sum_{\ell=0}^{k-1} m''_\ell,$$
$$\widetilde{\beta} : K \to \mathbf{Z}[0, 2^{\widetilde{m}''})$$
$$\widetilde{\beta}(s) = (\beta_0(s), \ldots, \beta_{k-1}(s)) \quad \text{for} \quad s \in K.$$
$$\widetilde{m} = k_0 + \widetilde{m}'' + \sum_{\ell=0}^{k-1} m_\ell$$
$$\widetilde{Q} = (\widetilde{m}, \widetilde{m}', \widetilde{m}'', \widetilde{Z}, \widetilde{\tau}, \widetilde{\beta}).$$

Let us fix the following notation: Consider the splitting

$$H_{\widetilde{m}} = H_{k_0} \otimes H_{\widetilde{m}''} \otimes H_{m_0} \otimes \cdots \otimes H_{m_{k-1}}.$$

[1] Throughout this paper log stands for $\log_2$.

The representation of a basis state

$$|i\rangle\,|u\rangle\,|x_0\rangle\ldots|x_{k-1}\rangle$$

refers to this splitting. We also need refined splittings. We represent

$$H_{\widetilde{m}''} = H_{m''_0} \otimes \cdots \otimes H_{m''_{k-1}},$$

and

$$|u\rangle = |u_0\rangle\ldots|u_{k-1}\rangle$$

corresponds to that splitting. Similarly,

$$H_{m_\ell} = H_{m'} \otimes H_{m''_\ell} \otimes H_{m-m'-m''_\ell}$$

with the respective

$$|x_\ell\rangle = |i_\ell\rangle\,|y_\ell\rangle\,|z_\ell\rangle\,.$$

Next we define the following unitary operators on $H_{\widetilde{m}}$ by their action on the basis states:

$$J\,|i\rangle\,|u\rangle\,|x_0\rangle\ldots|x_{k-1}\rangle = |i\rangle\,|\ominus u\rangle\,|x_0\rangle\ldots|x_{k-1}\rangle\,,$$

where $\ominus$ means subtraction modulo $2^{\widetilde{m}''}$ and $\ominus u$ stands for $0 \ominus u$,

$$C\,|i\rangle\,|u\rangle\,|x_0\rangle\ldots|x_{k-1}\rangle = |i \oplus 1\rangle\,|u\rangle\,|x_0\rangle\ldots|x_{k-1}\rangle\,,$$

for $\ell = 0, \ldots, k-1$, $j = 0, \ldots, n_\ell$,

$$\begin{aligned} &T_\ell\,|i\rangle\,|u_0\rangle\ldots|u_\ell\rangle\ldots|u_{k-1}\rangle\,|x_0\rangle\ldots|i_\ell\rangle\,|y_\ell\rangle\,|z_\ell\rangle\ldots|x_{k-1}\rangle \\ &= |i\rangle\,|u_0\rangle\ldots|u_\ell\rangle\ldots|u_{k-1}\rangle\,|x_0\rangle\ldots|i_\ell\rangle\,|y_\ell \oplus u_\ell\rangle\,|z_\ell\rangle\ldots|x_{k-1}\rangle \end{aligned}$$

$$\widetilde{U}_{\ell j}\,|i\rangle\,|u\rangle\,|x_0\rangle\ldots|x_\ell\rangle\ldots|x_{k-1}\rangle = |i\rangle\,|u\rangle\,|x_0\rangle\ldots(U_{\ell j}\,|x_\ell\rangle)\ldots|x_{k-1}\rangle\,,$$

$$\begin{aligned} &P_\ell\,|i\rangle\,|u\rangle\,|x_0\rangle\ldots|i_\ell\rangle\,|y_\ell\rangle\,|z_\ell\rangle\ldots|x_{k-1}\rangle \\ &= |i\rangle\,|i_\ell\rangle\,|u\rangle\,|x_0\rangle\ldots|y_\ell\rangle\,|z_\ell\rangle\ldots|x_{k-1}\rangle\,, \end{aligned}$$

and finally, for $\ell = 1, \ldots, k-1$,

$$\begin{aligned} &B_\ell\,|i\rangle\,|u\rangle\,|x_0\rangle\ldots|x_{\ell-1}\rangle\,|x_\ell\rangle\ldots|x_{k-1}\rangle \\ &= |i\rangle\,|u_0\rangle\,|x_0\rangle\ldots|x_{\ell-1}\rangle\,|x_\ell \oplus b_\ell(x_0, \ldots, x_{\ell-1})\rangle\ldots|x_{k-1}\rangle\,. \end{aligned}$$

Now we present the simulation of the queries $Q_{\ell,f}$ by $\widetilde{Q}_f$: Let $0 \le \ell \le k-1$. It is readily checked, that if we apply the operator $P_\ell^{-1}\widetilde{Q}_f P_\ell$ to the state

$$|\ell\rangle\,|0\rangle\,|x_0\rangle\dots|x_{k-1}\rangle\,,$$

we get

$$\begin{aligned} & |\ell\rangle\left|\widetilde{\beta}(f(\widetilde{\tau}(\ell,i_\ell)))\right\rangle|x_0\rangle\dots\dots|x_{k-1}\rangle \\ = \;& |\ell\rangle\,|\beta_0(f(\tau_\ell(i_\ell)))\rangle\dots|\beta_{k-1}(f(\tau_\ell(i_\ell)))\rangle\,|x_0\rangle\dots\dots|x_{k-1}\rangle\,, \end{aligned}$$

provided $i_\ell \in Z_\ell$. Applying then T_ℓ to this state gives

$$\begin{aligned} & |\ell\rangle\left|\widetilde{\beta}(f(\widetilde{\tau}(\ell,i_\ell)))\right\rangle|x_0\rangle\dots|i_\ell\rangle\,|y_\ell \oplus \beta_\ell(f(\tau_\ell(i_\ell)))\rangle\,|z_\ell\rangle\dots|x_{k-1}\rangle \\ = \;& |\ell\rangle\left|\widetilde{\beta}(f(\widetilde{\tau}(\ell,i_\ell)))\right\rangle|x_0\rangle\dots(Q_{\ell,f}\,|x_\ell\rangle)\dots|x_{k-1}\rangle\,. \end{aligned}$$

Next J is applied which yields

$$|\ell\rangle\left|\ominus\widetilde{\beta}(f(\widetilde{\tau}(\ell,i_\ell)))\right\rangle|x_0\rangle\dots(Q_{\ell,f}\,|x_\ell\rangle)\dots|x_{k-1}\rangle\,,$$

and finally the application of $P_\ell^{-1}\widetilde{Q}_f P_\ell$ produces

$$|\ell\rangle\,|0\rangle\,|x_0\rangle\dots(Q_{\ell,f}\,|x_\ell\rangle)\dots|x_{k-1}\rangle\,.$$

If $i_\ell \notin Z_\ell$, this also holds, which is checked in the same way. Hence we showed that

$$\begin{aligned} & P_\ell^{-1}\widetilde{Q}_f P_\ell J T_\ell P_\ell^{-1}\widetilde{Q}_f P_\ell\,|\ell\rangle\,|0\rangle\,|x_0\rangle\dots|x_\ell\rangle\dots|x_{k-1}\rangle \\ & \qquad = |\ell\rangle\,|0\rangle\,|x_0\rangle\dots(Q_{\ell,f}\,|x_\ell\rangle)\dots|x_{k-1}\rangle\,. \end{aligned} \tag{10}$$

The new algorithm can now be described as follows:

```
initialize $|0\rangle\,|0\rangle\,|b_0\rangle\,|0\rangle\dots|0\rangle$
for $\ell = 0,\dots,k-1$ do
    apply $\widetilde{U}_{\ell,0}$   (beginning of $\ell$-th cycle of original algorithm)
    for $j = 1,\dots,n_\ell$
        apply $P_\ell^{-1}\widetilde{Q}_f P_\ell J T_\ell P_\ell^{-1}\widetilde{Q}_f P_\ell$
        apply $\widetilde{U}_{\ell j}$   (end of $\ell$-th cycle of original algorithm)
    if $\ell \neq k-1$
        apply $B_{\ell+1}$   (computing $b_{\ell+1}$ as initial state of next
        cycle)
        apply $C$   (increasing the counter by one)
    measure all qubits corresponding to the components
    $H_{m_0}, \ldots, H_{m_{k-1}}$ (let $|x_0\rangle \ldots |x_{k-1}\rangle$ be the result)
    compute $\varphi(x_0, \ldots, x_{k-1})$.
```

The starting passage through the outer loop ($\ell = 0$) acts as follows:

$$|0\rangle\,|0\rangle\,|b_0\rangle\,|0\rangle^{(k-1)} \to |1\rangle\,|0\rangle \left(\sum_{x_0} A_{0,f}(x_0, b_0)\,|x_0\rangle\,|b_1(x_0)\rangle \right) |0\rangle^{(k-2)}.$$

The passage with index ℓ, $1 \leq \ell \leq k-2$, maps each basis state of the form

$$|\ell\rangle\,|0\rangle\,|x_0\rangle \ldots |x_{\ell-1}\rangle\,|y\rangle\,|0\rangle^{(k-\ell-1)}$$

into

$$|\ell+1\rangle\,|0\rangle\,|x_0\rangle \ldots |x_{\ell-1}\rangle \left(\sum_{x_\ell} A_{\ell,f}(x_\ell, y)\,|x_\ell\rangle\,|b_{\ell+1}(x_0, \ldots, x_\ell)\rangle \right) |0\rangle^{(k-\ell-2)}.$$

Finally, the last passage ($\ell = k-1$) acts as follows:

$$\begin{aligned} &|k-1\rangle\,|0\rangle\,|x_0\rangle \ldots |x_{k-2}\rangle\,|y\rangle \to \\ &\qquad |k-1\rangle\,|0\rangle\,|x_0\rangle \ldots |x_{k-2}\rangle \left(\sum_{x_{k-1}} A_{k-1,f}(x_{k-1}, y)\,|x_{k-1}\rangle \right). \end{aligned}$$

From this it follows that the overall result of the algorithm before measurement is the state

$$\begin{aligned} &\sum_{x_0, \ldots, x_{k-1}} A_{0,f}(x_0, b_0) A_{1,f}(x_1, b_1(x_0)) \ldots \\ &\qquad \ldots A_{k-1,f}(x_{k-1}, b_{k-1}(x_0, \ldots, x_{k-2}))\,|k-1\rangle\,|0\rangle\,|x_0\rangle \ldots |x_{k-1}\rangle. \end{aligned}$$

The probability of measuring $|x_0\rangle \ldots |x_{k-1}\rangle$ is thus

$$|A_{0,f}(x_0, b_0)|^2 |A_{1,f}(x_1, b_1(x_0))|^2 \ldots |A_{k-1,f}(x_{k-1}, b_{k-1}(x_0, \ldots, x_{k-2}))|^2,$$

which equals

$$\mathbf{P}\{\xi_{0,f} = x_0, \ldots, \xi_{k-1,f} = x_{k-1}\},$$

by (7) and (8). This proves the lemma. □

We will sometimes write that we repeat a quantum algorithm a number of times, or, more generally, that we apply to $f \in F$ a finite sequence of algorithms A_i from F to G_i ($i = 0, \ldots, M-1$) and combine the results by the help of a classical computation. Let

$$\psi : G_0 \times \cdots \times G_{M-1} \to G$$

be any mapping. Using our notion of a quantum algorithm with measurements, a formal representation of the composed algorithm A, which we write symbolically as

$$A = \psi(A_0, \ldots, A_{M-1}), \tag{11}$$

can easily be given as follows: Let

$$A_i = ((A_{i,\ell})_{\ell=0}^{k_i-1}, (b_{i,\ell})_{\ell=0}^{k_i-1}, \varphi_i),$$

put $k = \sum_{i=0}^{M-1} k_i$, let the set

$$\Upsilon = \{(i,\ell) \mid i = 0, \ldots, M-1,\, \ell = 0, \ldots, k_i - 1\}$$

be equipped with the lexicographical order, and let

$$\varphi = \psi(\varphi_0, \ldots, \varphi_{M-1}).$$

Then we define

$$\psi(A_0, \ldots, A_{M-1}) = ((A_v)_{v \in \Upsilon}, (b_v)_{v \in \Upsilon}, \varphi).$$

The next lemma gives some further description of the composition and is readily checked using the definition of a quantum algorithm. We need the following notation: For probability measures $\mu_0, \ldots, \mu_{M-1} \in \mathcal{P}_0(G)$ let $\psi(\mu_0, \ldots, \mu_{M-1}) \in \mathcal{P}_0(G)$ be the measure induced by $\mu_0 \times \cdots \times \mu_{M-1}$ via ψ on G, that is, for $C \subseteq G$,

$$\psi(\mu_0, \ldots, \mu_{M-1})(C) = (\mu_0 \times \cdots \times \mu_{M-1})(\psi^{-1}(C)).$$

Lemma 2. *For each $f \in F$,*

$$\psi(A_0, \ldots, A_{M-1})(f) = \psi(A_0(f), \ldots, A_{M-1}(f)),$$

or stated equivalently, if $(\zeta_i)_{i=0}^{M-1}$ are independent random variables with distribution $A_i(f)$ respectively, then

$$\psi(A_0, \ldots, A_{M-1})(f) = \operatorname{dist}(\psi(\zeta_0, \ldots, \zeta_{M-1})).$$

Moreover,

$$n_q(\psi(A_0,\dots,A_{M-1})) = \sum_{i=0}^{M-1} n_q(A_i).$$

The next lemma concerns the special case of repeating an algorithm. It describes a standard technique of boosting the success probability. For completeness, we include the short proof. Let $G = \mathbf{R}$, $M \in \mathbf{N}$ and denote by $\psi_0 : \mathbf{R}^M \to \mathbf{R}$ the mapping given by the median, that is, $\psi_0(a_0,\dots,a_{M-1})$ is the value of the of the $\lceil (M+1)/2 \rceil$-th element of the non-decreasing rearrangement of (a_i). For any algorithm A from F to $\mathbf{R}$ denote $\psi_0(A^M) := \psi_0(A,\dots,A)$.

Lemma 3. *Let A be any quantum algorithm and S be any mapping from F to $\mathbf{R}$. Then for each $f \in F$,*

$$e(S, \psi_0(A^M), f, e^{-M/8}) \le e(S, A, f).$$

Proof. Fix $f \in F$. Let $\zeta_0,\dots,\zeta_{M-1}$ be independent random variables with distribution $A(f)$. Let χ_i be the indicator function of the set $\{|S(f) - \zeta_i| > e(S,A,f)\}$. Then $\mathbf{P}\{\chi_i = 1\} \le 1/4$. Hoeffding's inequality, see e.g. Pollard (1984), p. 191, yields

$$\mathbf{P}\left\{\sum_{i=0}^{M-1} \chi_i \ge M/2\right\} \le \mathbf{P}\left\{\sum_{i=0}^{M-1} (\chi_i - \mathbf{E}\chi_i) \ge M/4\right\} \le e^{-M/8}.$$

Hence, with probability at least $1 - e^{-M/8}$,

$$\left|\{i \mid |S(f) - \zeta_i| \le e(S,A,f)\}\right| > M/2,$$

which implies

$$|S(f) - \psi_0(\zeta_0,\dots,\zeta_{M-1})| \le e(S,A,f).$$

□

Another way of building new algorithms from previous ones will also be important for us. To explain it, let $\emptyset \ne F \subseteq \mathcal{F}(D,K)$ and $\emptyset \ne \widetilde{F} \subseteq \mathcal{F}(\widetilde{D},\widetilde{K})$, where $D, \widetilde{D}, K, \widetilde{K}$ are nonempty sets. In the construction of a new algorithm A on F we sometimes construct from f a function $\widetilde{f} = \Gamma(f) \in \widetilde{F}$ to which we want to apply an already developed algorithm $\widetilde{A}$ on $\widetilde{F}$. By definition, the algorithm A on F can only use queries Q on F itself, while we need to use $\widetilde{Q}_{\Gamma(f)}$, where $\widetilde{Q}$ is a query on $\widetilde{F}$. Nevertheless often a solution can be found as follows: We simulate $\widetilde{Q}_{\Gamma(f)}$ either as Q_f with a suitable query Q

on F or as B_f, where B is an algorithm without measurement on F. The details are given below.

The first result covers the simple situation where one query is just replaced by another. Let $\eta : \widetilde{D} \to D$ and $\varrho : K \to \widetilde{K}$ be arbitrary mappings and define $\Gamma : F \to \widetilde{F}$ by

$$\Gamma(f) = \varrho \circ f \circ \eta. \tag{12}$$

Lemma 4. *Let Γ be a mapping of the form (12). Then for each query $\widetilde{Q}$ on $\widetilde{F}$ there is a query Q on F such that $m(Q) = m(\widetilde{Q})$ and for all $f \in F$*

$$Q_f = \widetilde{Q}_{\Gamma(f)}.$$

Proof. Let

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

Then we define

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

where $\tau = \eta \circ \widetilde{\tau}$ and $\beta = \widetilde{\beta} \circ \varrho$. Now the lemma follows directly from the query definition. □

The second result in this direction is slightly more technical. We assume that we are given a mapping $\Gamma : F \to \widetilde{F}$ of the following type: There are an $m^* \in \mathbf{N}$ and mappings

$$\begin{aligned} \eta &: \widetilde{D} \to D \\ \beta &: K \to \mathbf{Z}[0, 2^{m^*}) \\ \varrho &: \widetilde{D} \times \mathbf{Z}[0, 2^{m^*}) \to \widetilde{K} \end{aligned}$$

such that for $f \in F$ and $s \in \widetilde{D}$

$$\Gamma(f)(s) = \varrho(s, \beta \circ f \circ \eta(s)). \tag{13}$$

Lemma 5. *Let $\widetilde{Q}$ be a quantum query on $\widetilde{F}$ and let Γ be a mapping of the above form (13). Then there is a quantum algorithm without measurement B on F such that $n_q(B) = 2$, $m(B) = m(\widetilde{Q}) + m^*$ and for all $f \in F$, $x \in \mathbf{Z}[0, 2^{m(\widetilde{Q})})$,*

$$B_f \,|x\rangle \,|0\rangle_{m^*} = (\widetilde{Q}_{\Gamma(f)} \,|x\rangle) \,|0\rangle_{m^*},$$

where $|0\rangle_{m^*}$ stands for the zero state in $\mathbf{Z}[0, 2^{m^*})$.

Proof. Let

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

and put

$$m = \widetilde{m} + m^*, \quad m' = \widetilde{m}', \quad m'' = m^*,$$
$$Z = \widetilde{Z}, \quad \tau = \eta \circ \widetilde{\tau},$$

let β be as above and define

$$Q = (m, m', m'', Z, \tau, \beta).$$

We represent

$$H_m = H_{\widetilde{m}'} \otimes H_{\widetilde{m}''} \otimes H_{\widetilde{m}-\widetilde{m}'-\widetilde{m}''} \otimes H_{m^*},$$

a basis state of which will be written as

$$|i\rangle\, |x\rangle\, |y\rangle\, |z\rangle\, .$$

Define the permutation operator P by

$$P\, |i\rangle\, |x\rangle\, |y\rangle\, |z\rangle = |i\rangle\, |z\rangle\, |x\rangle\, |y\rangle\, ,$$

the operator of sign inversion

$$J\, |i\rangle\, |z\rangle\, |x\rangle\, |y\rangle = |i\rangle\, |\ominus z\rangle\, |x\rangle\, |y\rangle\, ,$$

and finally

$$T\, |i\rangle\, |z\rangle\, |x\rangle\, |y\rangle = |i\rangle\, |z\rangle \left| x \oplus \widetilde{\beta} \circ \varrho(\widetilde{\tau}(i), z) \right\rangle |y\rangle$$

if $i \in Z$, and

$$T\, |i\rangle\, |z\rangle\, |x\rangle\, |y\rangle = |i\rangle\, |z\rangle\, |x\rangle\, |y\rangle$$

if $i \notin Z$. We define B by setting for $f \in F$,

$$B_f = P^{-1} Q_f J T Q_f P.$$

Let us trace the action of B_f on

$$|i\rangle\, |x\rangle\, |y\rangle\, |0\rangle\, .$$

First we assume $i \in Z$. The transformation $Q_f P$ leads to

$$|i\rangle\, |\beta(f(\tau(i)))\rangle\, |x\rangle\, |y\rangle = |i\rangle\, |\beta \circ f \circ \eta \circ \widetilde{\tau}(i)\rangle\, |x\rangle\, |y\rangle\, .$$

Then the above is mapped by T to

$$\begin{aligned} & |i\rangle\,|\beta\circ f\circ\eta\circ\widetilde{\tau}(i)\rangle\,\Big|x\oplus\widetilde{\beta}\circ\varrho(\widetilde{\tau}(i),\beta\circ f\circ\eta\circ\widetilde{\tau}(i))\Big\rangle\,|y\rangle \\ = \; & |i\rangle\,|\beta(f(\tau(i)))\rangle\,\Big|x\oplus\widetilde{\beta}(\Gamma(f)(\widetilde{\tau}(i)))\Big\rangle\,|y\rangle\,, \end{aligned}$$

and $P^{-1}Q_f J$ gives

$$|i\rangle\,\Big|x\oplus\widetilde{\beta}(\Gamma(f)(\widetilde{\tau}(i)))\Big\rangle\,|y\rangle\,|0\rangle = \big(\widetilde{Q}_{\Gamma(f)}\,|i\rangle\,|x\rangle\,|y\rangle\,\big)\,|0\rangle\,.$$

The case $i\notin Z$ is checked analogously. □

Corollary 1. *Given a mapping $\Gamma : F\to\widetilde{F}$ as in (12) or (13), a normed space G and a quantum algorithm $\widetilde{A}$ from $\widetilde{F}$ to G, there is a quantum algorithm A from F to G with*

$$n_q(A) = \begin{cases} n_q(\widetilde{A}) & \textit{in case of (12)} \\ 2\,n_q(\widetilde{A}) & \textit{in case of (13)} \end{cases}$$

and for all $f\in F$

$$A(f) = \widetilde{A}(\Gamma(f)).$$

Consequently, if $\widetilde{S}:\widetilde{F}\to G$ is any mapping and $S=\widetilde{S}\circ\Gamma$, then for each $n\in\mathbf{N}_0$

$$\begin{aligned} e_n^q(S,F) &\le e_n^q(\widetilde{S},\widetilde{F}) \quad \textit{in case of (12), and} \\ e_{2n}^q(S,F) &\le e_n^q(\widetilde{S},\widetilde{F}) \quad \textit{in case of (13).} \end{aligned}$$

Proof. Let

$$\widetilde{A} = ((\widetilde{A}_\ell)_{\ell=0}^{k-1},(\widetilde{b}_\ell)_{\ell=0}^{k-1},\widetilde{\varphi}),\quad \widetilde{A}_\ell = (\widetilde{Q}_\ell,(\widetilde{U}_{\ell,j})_{j=0}^{n_\ell}),$$

and $\widetilde{m}_\ell = m(\widetilde{A}_\ell)$. Then for $f\in F$, $0\le\ell<k$,

$$\widetilde{A}_{\ell,\Gamma(f)} = \widetilde{U}_{\ell,n_\ell}\widetilde{Q}_{\ell,\Gamma(f)}\widetilde{U}_{\ell,n_\ell-1}\dots\widetilde{U}_{\ell,1}\widetilde{Q}_{\ell,\Gamma(f)}\widetilde{U}_{\ell,0}.$$

In case of (12) we obtain A by just replacing $\widetilde{Q}_\ell$ by Q_ℓ from Lemma 4. It follows from (7) and (9) that

$$A(f) = \widetilde{A}(\Gamma(f)).$$

In case of (13) we replace $\widetilde{Q}_\ell$ by B_ℓ from Lemma 5, $\widetilde{U}_{\ell,j}$ by $U_{\ell,j} = \widetilde{U}_{\ell,j} \otimes Id_{H_{m^*}}$, where $Id_{H_{m^*}}$ is the identity on H_{m^*}, the state $\left|\widetilde{b}_0\right\rangle$ by $|b_0\rangle = \left|\widetilde{b}_0\right\rangle |0\rangle_{m^*}$ and, for $1 \le \ell \le k-1$, the mappings

$$\widetilde{b}_\ell : \prod_{i=0}^{\ell-1} \mathbf{Z}[0, 2^{\widetilde{m}_i}) \to \mathbf{Z}[0, 2^{\widetilde{m}_\ell})$$

by

$$b_\ell : \prod_{i=0}^{\ell-1} \left(\mathbf{Z}[0, 2^{\widetilde{m}_i}) \times \mathbf{Z}[0, 2^{m^*})\right) \to \mathbf{Z}[0, 2^{\widetilde{m}_\ell}) \times \mathbf{Z}[0, 2^{m^*}),$$

defined by

$$b_\ell((x_0, y_0), \dots, (x_{\ell-1}, y_{\ell-1})) = \left(\widetilde{b}_\ell(x_0, \dots, x_{\ell-1}), 0\right).$$

Finally, we replace

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

by

$$\varphi : \prod_{\ell=0}^{k-1} \left(\mathbf{Z}[0, 2^{\widetilde{m}_\ell}) \times \mathbf{Z}[0, 2^{m^*})\right) \to G,$$

defined as

$$\varphi((x_0, y_0), \dots, (x_{k-1}, y_{k-1})) = \widetilde{\varphi}(x_0, \dots, x_{k-1}).$$

It follows that

$$A_{\ell,f}((x,y),(z,0)) = \begin{cases} \widetilde{A}_{\ell,\Gamma(f)}(x,z) & \text{if } \ y = 0 \\ 0 & \text{otherwise,} \end{cases}$$

and therefore, by (7),

$$\begin{aligned} & p_{A,f}((x_0, y_0) \dots, (x_{k-1}, y_{k-1})) \\ & \quad = \begin{cases} p_{\widetilde{A},\Gamma(f)}(x_0, \dots, x_{k-1}) & \text{if } \ y_0 = \dots = y_{k-1} = 0 \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

which together with (9) yields

$$A(f) = \widetilde{A}(\Gamma(f)).$$

This proves the first part of the statement. The second part is an obvious consequence. □

Finally we state some elementary but useful properties of e_n^q. For $\lambda \in \mathbf{K}$ define $\lambda S : F \to G$ by $(\lambda S)(f) = \lambda S(f)$ $(f \in F)$. Furthermore, in the case $K = \mathbf{K}$ we denote $\lambda F = \{\lambda f \mid f \in F\}$.

Lemma 6. *Let $S, T : F \to G$ be any mappings, $n \in \mathbf{N}_0$ and assume that $e_n^q(S, F)$ is finite. Then the following hold:*
(i)

$$e_n^q(T, F) \le e_n^q(S, F) + \sup_{f \in F} \|T(f) - S(f)\|.$$

(ii) For each $\lambda \in \mathbf{K}$

$$e_n^q(\lambda S, F) = |\lambda| e_n^q(S, F).$$

(iii) If $K = \mathbf{K}$ and S is a linear operator from $\mathcal{F}(D, K)$ to G, then for all $\lambda \in \mathbf{K}$

$$e_n^q(S, \lambda F) = |\lambda| e_n^q(S, F).$$

Proof. The first two statements are simple consequences of the definitions. Let us verify the third one. Let $\widetilde{F} = \lambda F$, $\Gamma : F \to \widetilde{F}$ be defined as $\Gamma(f) = \lambda f$, which is of the form (12). We assume $\lambda \neq 0$, the case $\lambda = 0$ follows trivially from (ii). Since S is linear, we have

$$\lambda^{-1} S \circ \Gamma = S,$$

and hence, by Corollary 1 and statement (ii) above,

$$e_n^q(S, F) \le |\lambda|^{-1} e_n^q(S, \widetilde{F}) = |\lambda|^{-1} e_n^q(S, \lambda F).$$

Replacing F by λF and λ by λ^{-1}, we get

$$e_n^q(S, \lambda F) \le |\lambda| e_n^q(S, F),$$

which completes the proof. □

3 Quantum Summation

In this section we study summation of sequences or, what is essentially the same, the computation of the mean, on a quantum computer. For a fixed $N \in \mathbf{N}$ we set $D = \mathbf{Z}[0, N)$, $K = \mathbf{R}$, $G = \mathbf{R}$, and for $1 \le p \le \infty$ 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

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

(Note that L_p^N is just the space $L_p(D,\mu)$, where μ is the equidistribution on D.) Define $S_N : L_p^N \to \mathbf{R}$ by

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

We let

$$F = \mathcal{B}_p^N := \mathcal{B}(L_p^N) = \{f \in L_p^N \mid \|f\|_{L_p^N} \le 1\}$$

be the unit ball of L_p^N. We also define

$$\mathcal{B}_{\infty,+}^N = \{f : D \to \mathbf{R} \,|\, 0 \le f(i) \le 1 \text{ for all } i\}$$

and

$$\mathcal{B}_{\infty,0}^N = \{f : D \to \{0,1\}\}.$$

When we consider $\mathcal{B}_{\infty,0}^N$, we put $K = \{0,1\}$. Clearly,

$$\mathcal{B}_{\infty,0}^N \subset \mathcal{B}_{\infty,+}^N \subset \mathcal{B}_p^N \subset \mathcal{B}_q^N$$

whenever $1 \le q < p < \infty$. Therefore, we will also consider S_N as acting on $\mathcal{B}_{\infty,0}^N$ and $\mathcal{B}_{\infty,+}^N$. We use the following standard representations depending on the range of $f \in F$: Given $a, b \in \mathbf{R}$, $a < b$, and $\kappa \in \mathbf{N}$, define $\beta_{\kappa,a,b} : \mathbf{R} \to \mathbf{Z}[0, 2^\kappa)$ by

$$\beta_{\kappa,a,b}(x) = \begin{cases} 2^\kappa - 1 & \text{if } \ x \ge b \\ 0 & \text{if } \ x < a \\ i & \text{if } \ \frac{x-a}{b-a} \in [\frac{i}{2^\kappa}, \frac{i+1}{2^\kappa}),\ i \in \mathbf{Z}[0,2^\kappa). \end{cases} \tag{14}$$

So for $a \le x < b$

$$\beta_{\kappa,a,b}(x) = \left\lfloor 2^\kappa \frac{x-a}{b-a} \right\rfloor,$$

and hence, for $a \le x \le b$,

$$a + (b-a)\,2^{-\kappa}\beta_{\kappa,a,b}(x) \le x \le a + (b-a)\,2^{-\kappa}(\beta_{\kappa,a,b}(x) + 1). \tag{15}$$

First we state the basic result on quantum counting due to Brassard, Høyer, Mosca, and Tapp (2000).

Lemma 7. *There is a constant $c > 0$ such that for all $n, N \in \mathbf{N}$ there is a quantum algorithm A from $\mathcal{B}^N_{\infty,0}$ to $\mathbf{R}$ such that $n_q(A) \le n$ and for each $f \in \mathcal{B}^N_{\infty,0}$*

$$e(S_N, A, f) \le c\left(\sqrt{S_N f}\, n^{-1} + n^{-2}\right).$$

Remark. Throughout this paper we often use the same symbol for possibly different constants. These constants are either absolute or may depend only on p – the summability parameter of the L_p-spaces considered (in all lemmas and theorems this is precisely described anyway by the order of the quantifiers).

Proof. We refer to Brassard, Høyer, Mosca, and Tapp (2000) for details of the algorithm, its analysis and the resulting estimates. For us, there remains one detail to be verified. Their algorithm makes use of the controlled application of the Grover iterate and assumes that an implementation of this procedure is available. This means, roughly, if Y stands for the Grover iterate, we must be able to implement an operation which maps an element $|i\rangle\,|k\rangle$ to $(Y^k\,|i\rangle)\,|k\rangle$ (that is, different basis elements may be subject to different powers of Y). Since Y involves a query call, it is not immediately clear, how this could be achieved within the rules delveloped in section 2, that is, in our model of computation and its way to use queries. So we supply the needed argument here. It is a simulation procedure, similar to the ones above.

The parameters of the algorithm will be the following. It has one measurement, and the query Q is determined by

$$m' = \lceil \log N \rceil, \quad m'' = 1, \quad m^* = \lceil \log n \rceil,$$
$$m = m' + 2m^* + 2, \quad Z = \mathbf{Z}[0, N),$$
$$\tau : Z \to Z[0, 2^{m'}) \quad \text{and} \quad \beta : \{0,1\} \to \{0,1\} \quad \text{the identities}$$

(recall that $K = \{0,1\}$). Let

$$H_m = H_{m'} \otimes H_1 \otimes H_{m^*} \otimes H_{m^*},$$

and let the basis state

$$|i\rangle\,|x\rangle\,|j\rangle\,|k\rangle$$

correspond to this splitting. Let Φ_{n,m^*} be the n-term quantum Fourier transform on m^* qubits,

$$\Phi_{n,m^*}\,|k\rangle = \begin{cases} \frac{1}{\sqrt{n}} \sum_{y=0}^{n-1} e^{2\pi\imath ky/n}\,|y\rangle & \text{if} \quad k < n \\ |k\rangle & \text{otherwise.} \end{cases}$$

Define $\Phi \in \mathcal{U}(H_m)$ by

$$\Phi |i\rangle |x\rangle |j\rangle |k\rangle = |i\rangle |x\rangle |j\rangle (\Phi_{n,m^*} |k\rangle).$$

Furthermore, let $V_0 \in \mathcal{U}(H_{m'})$ be the Walsh-Hadamard transform W_N, if N is a power of 2, and let $V_0 = \Phi_{N,m'}$, if not. Define $X_0 \in \mathcal{U}(H_{m'})$ by

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

and unitary transforms on H_m by

$$\begin{aligned} V |i\rangle |x\rangle |j\rangle |k\rangle &= (V_0 |i\rangle) |x\rangle |j\rangle |k\rangle , \\ X |i\rangle |x\rangle |j\rangle |k\rangle &= \begin{cases} (X_0 |i\rangle) |x\rangle |j\rangle |k\rangle & \text{if} \quad j < k \\ |i\rangle |x\rangle |j\rangle |k\rangle & \text{otherwise,} \end{cases} \\ T |i\rangle |x\rangle |j\rangle |k\rangle &= \begin{cases} (-1)^{x+1} |i\rangle |x\rangle |j\rangle |k\rangle & \text{if} \quad j < k \\ |i\rangle |x\rangle |j\rangle |k\rangle & \text{otherwise,} \end{cases} \\ C |i\rangle |x\rangle |j\rangle |k\rangle &= |i\rangle |x\rangle |j \oplus 1\rangle |k\rangle . \end{aligned}$$

Now we define the algorithm as follows. For $f \in \mathcal{B}^N_{\infty,0}$ set

$$Y_f = CVXV^{-1}Q_fTQ_f.$$

The unitary transform of the algorithm is given by

$$\Phi^{-1}Y_f^{n-1}\Phi V.$$

The initial state is

$$b = |0\rangle |0\rangle |0\rangle |0\rangle .$$

Let us now follow the action of the algorithm. The element b is transformed by ΦV into

$$(V_0 |0\rangle) |0\rangle |0\rangle (\Phi_{n,m^*} |0\rangle).$$

Note that this vector is a linear combination of basis states of the form

$$|i\rangle |0\rangle |0\rangle |k\rangle$$

with $i < N$ and $k < n$. Next consider the application of Y_f to a basis state of the form

$$|i\rangle |0\rangle |j\rangle |k\rangle \tag{16}$$

with $i < N$ and $k < n$. First we assume $j < k$. Then $Q_f T Q_f$ produces

$$(-1)^{f(i)+1} |i\rangle |0\rangle |j\rangle |k\rangle .$$

After the application of $CVXV^{-1}$ we get

$$(-1)^{f(i)+1}(V_0 X_0 V_0^{-1} |i\rangle) |0\rangle |j \oplus 1\rangle |k\rangle ,$$

which is a linear combination of vectors of the form

$$|i'\rangle |0\rangle |j+1\rangle |k\rangle$$

with $i' < N$. If $j \geq k$, the application of Y_f to (16) gives

$$|i\rangle |0\rangle |j \oplus 1\rangle |k\rangle .$$

It is now clear that $Y_f = CVXV^{-1}Q_f T Q_f$ realizes the Grover iterate on the first component if $j < k$, that Y_f^{n-1} is the controlled (by k) application of it and the whole algorithm, considered just on the first and last component $|i\rangle |k\rangle$, is the algorithm "Est_Amp" of Brassard, Høyer, Mosca, and Tapp (2000), if we define φ on the measured state

$$|y\rangle = |i\rangle |x\rangle |j\rangle |k\rangle$$

as

$$\varphi(y) = \sin^2\left(\pi \frac{k}{n}\right) .$$

The required estimate (with a concrete value of the constant) is contained in Theorem 12 of that paper. Since our implementation requires $2n$ queries, we rescale n and modify the constant appropriately. □

The next result is essentially a translation of Lemma 7 into the setting of $\mathcal{B}^N_{\infty,+}$. The idea of using comparison queries is due to Abrams and Williams (1999).

Lemma 8. *There is a constant $c > 0$ such that for all $\nu, n, N \in \mathbf{N}$ there is a quantum algorithm A from $\mathcal{B}^N_{\infty,+}$ to $\mathbf{R}$ such that $n_q(A) \leq \nu n$ and for each $f \in \mathcal{B}^N_{\infty,+}$*

$$e(S_N, A, f, 2^{-\nu}) \leq c\left(\sqrt{S_N f}\, n^{-1} + n^{-2}\right) .$$

Proof. Let $\kappa \in \mathbf{N}$ be such that $2^\kappa \geq n^2$ and put $N_0 = N\,2^\kappa$. We shall apply Corollary 1 with $F = \mathcal{B}^N_{\infty,+}$ and $\widetilde{F} = \mathcal{B}^{N_0}_{\infty,0}$. Let $\widetilde{A}$ be any algorithm from $\mathcal{B}^{N_0}_{\infty,0}$ to $\mathbf{R}$ with one measurement, which satisfies the conclusion of Lemma 7 with $n_q(\widetilde{A}) := \widetilde{n} \leq n$. Let $\widetilde{A}$ be given by

$$\widetilde{A} = (\widetilde{A}_0, \widetilde{b}, \widetilde{\varphi}), \quad \widetilde{A}_0 = (\widetilde{Q}, (\widetilde{U}_j)_{j=0}^{\widetilde{n}}),$$

with

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

where $\widetilde{Z} \subset \mathbf{Z}[0, 2^{\widetilde{m}'})$, $\widetilde{\tau} : \widetilde{Z} \to \mathbf{Z}[0, N_0)$, and $\widetilde{\beta} : \{0,1\} \to \mathbf{Z}[0, 2^{\widetilde{m}''})$. We identify

$$\mathbf{Z}[0, N_0) = \mathbf{Z}[0, N) \times \mathbf{Z}[0, 2^\kappa)$$

and write correspondingly for $z \in \widetilde{Z}$,

$$\widetilde{\tau}(z) = (i(z), y(z)). \tag{17}$$

Now let $\beta = \beta_{\kappa,0,1}$ as defined in (14). For each $f \in \mathcal{B}^N_{\infty,+}$ define $\Gamma(f) \in \mathcal{B}^{N_0}_{\infty,0}$ by setting for $(i,y) \in \mathbf{Z}[0,N) \times \mathbf{Z}[0,2^\kappa) = \mathbf{Z}[0,N_0)$

$$\Gamma(f)(i,y) = \begin{cases} 1 & \text{if} \quad y < \beta(f(i)) \\ 0 & \text{otherwise.} \end{cases}$$

Note that

$$|\{y : \Gamma(f)(i,y) = 1\}| = \beta(f(i)),$$

and consequently

$$S_{N_0}\Gamma(f) = N^{-1} 2^{-\kappa} \sum_{i=0}^{N-1} \beta(f(i)).$$

By (15),

$$S_{N_0}\Gamma(f) \leq S_N f \leq S_{N_0}\Gamma(f) + 2^{-\kappa} \leq S_{N_0}\Gamma(f) + n^{-2}.$$

The mapping $\Gamma : f \to \Gamma(f)$ is easily seen to be of the form (13) (with $\eta(i,y) = i$ and β as defined above). By Corollary 1 there is an algorithm A on $\mathcal{B}^N_{\infty,+}$ such that $n_q(A) = 2n_q(\widetilde{A})$ and $A(f) = \widetilde{A}(\Gamma(f))$. To estimate the error of A, fix any $f \in \mathcal{B}^N_{\infty,+}$ and let ζ be a random variable with distribution $\widetilde{A}(\Gamma(f))$. Then, with probability at least 3/4,

$$\begin{aligned} |S_N f - \zeta| &\leq |S_N f - S_{N_0}\Gamma(f)| + |S_{N_0}\Gamma(f) - \zeta| \\ &\leq n^{-2} + c\left(\sqrt{S_{N_0}\Gamma(f)}\, n^{-1} + n^{-2}\right) \\ &\leq c'\left(\sqrt{S_N f}\, n^{-1} + n^{-2}\right). \end{aligned}$$

Now we use Lemma 3 to boost the success probability by repeating A $c_1\nu$ times, where $c_1 = \lceil 8/\log e \rceil$, and computing the median, which gives the desired error estimate

$$e(S, A^*, f, 2^{-\nu}) \leq c'\left(\sqrt{S_N f}\, n^{-1} + n^{-2}\right)$$

for the algorithm $A^* = \psi_0(A^{c_1\nu})$, whose number of queries is bounded by $2c_1\nu n$. A scaling of n at the expense of enlarging the constant gives the result as required. □

Now we are ready to estimate the numbers $e^q_n(S_N, \mathcal{B}^N_p)$. Note that this is nontrivial only when $n < N$. For $n \geq N$ a classical computer suffices, or, to put it more formally into our framework, we have $e^q_n(S_N, \mathcal{B}^N_p) = 0$, since with N queries (and a suitable number of qubits) the sum can be determined up to each degree of precision by e.g. simulating a classical computation.

The following is the main result of this section. For the sake of later reference we also include the already known case $p = \infty$ due to Brassard, Høyer, Mosca, and Tapp (2000), which we deduce formally from the case $2 < p < \infty$, but which is, in fact, an immediate consequence of the previous two lemmas.

Theorem 1. *Let $1 < p \leq \infty$. Then there is a constant $c > 0$ such that for all $n, N \in \mathbf{N}$, $n > 2$*

$$e^q_n(S_N, \mathcal{B}^N_p) \leq c \begin{cases} n^{-1} & \text{for} \quad p > 2 \\ n^{-1}\log^{3/2} n \log\log n & \text{for} \quad p = 2 \\ n^{-2(1-1/p)} & \text{for} \quad p < 2. \end{cases}$$

Proof. Let $1 < p < \infty$. Fix $k \in \mathbf{N}_0$ (to be specified later) and define for $f \in L^N_p$

$$\mathcal{I}^k_f = \left\{ i \in \mathbf{Z}[0, N) \,\middle|\, |f(i)| \geq 2^k \right\},$$

for $\sigma = 0, 1$,

$$\mathcal{J}_f^{0,\sigma} = \left\{ i \in \mathbf{Z}[0,N) \,\middle|\, 0 \le (-1)^\sigma f(i) < 1 \right\},$$

and for $\ell = 1, \ldots, k$,

$$\mathcal{J}_f^{\ell,\sigma} = \left\{ i \in \mathbf{Z}[0,N) \,\middle|\, 2^{\ell-1} \le (-1)^\sigma f(i) < 2^\ell \right\}.$$

Note that

$$N^{-1} 2^{pk} |\mathcal{I}_f^k| \le \frac{1}{N} \sum_{i \in \mathcal{I}_f^k} |f(i)|^p \le \|f\|_{L_p^N}^p,$$

hence

$$|\mathcal{I}_f^k| \le N\, 2^{-pk} \|f\|_{L_p^N}^p. \tag{18}$$

Hölder's inequality together with (18) gives

$$\begin{aligned} \left| \frac{1}{N} \sum_{i \in \mathcal{I}_f^k} f(i) \right| &\le \left(\frac{1}{N} |\mathcal{I}_f^k| \right)^{1/p'} \left(\frac{1}{N} \sum_{i \in \mathcal{I}_f^k} |f(i)|^p \right)^{1/p} \\ &\le 2^{-pk/p'} \|f\|_{L_p^N}^{p/p'} \|f\|_{L_p^N} = 2^{-(p-1)k} \|f\|_{L_p^N}^p, \end{aligned} \tag{19}$$

where $1/p + 1/p' = 1$. Furthermore,

$$\frac{1}{N} \sum_{\substack{1 \le \ell \le k \\ \sigma = 0,1}} 2^{p(\ell-1)} |\mathcal{J}_f^{\ell,\sigma}| \le \|f\|_{L_p^N}^p, \tag{20}$$

which gives, in particular,

$$|\mathcal{J}_f^{\ell,\sigma}| \le N\, 2^{-p(\ell-1)} \|f\|_{L_p^N}^p \quad (\ell \ge 1). \tag{21}$$

Now define $g_f^{\ell,\sigma} \in \mathcal{B}_{\infty,+}^N$ for $0 \le \ell \le k$, $\sigma = 0, 1$,

$$g_f^{\ell,\sigma}(i) = \begin{cases} (-1)^\sigma 2^{-\ell} f(i) & i \in \mathcal{J}_f^{\ell,\sigma} \\ 0 & \text{otherwise.} \end{cases}$$

Consequently, $0 \le g_f^{\ell,\sigma} \le 1$, so $g_f^{\ell,\sigma} \in \mathcal{B}_{\infty,+}^N$. Clearly,

$$S_N g_f^{\ell,\sigma} \le N^{-1} |\mathcal{J}_f^{\ell,\sigma}| \tag{22}$$

and

$$\begin{aligned} S_N f &= \frac{1}{N}\Big(\sum_{\substack{0\le \ell \le k\\ \sigma=0,1}} \sum_{i\in \mathcal{J}_f^{\ell,\sigma}} f(i) + \sum_{i\in \mathcal{I}_f^k} f(i)\Big) \\ &= \sum_{\substack{0\le \ell \le k\\ \sigma=0,1}} (-1)^\sigma 2^\ell S_N g_f^{\ell,\sigma} + \frac{1}{N}\sum_{i\in \mathcal{I}_f^k} f(i). \end{aligned} \tag{23}$$

Now the idea is to compute $S_N g_f^{\ell,\sigma}$ by the algorithm from Lemma 8 for all ℓ and σ, and from the results (in a classical way) the first sum of equation (23). Fix $\nu_\ell, n_\ell \in \mathbf{N}$ (to be specified later) and let, according to Lemma 8, $\widetilde{A}_\ell$ be an algorithm on $\mathcal{B}_{\infty,+}^N$ such that $n_q(\widetilde{A}_\ell) \le \nu_\ell n_\ell$ and for all $g \in \mathcal{B}_{\infty,+}^N$,

$$e(S_N, \widetilde{A}_\ell, g, 2^{-\nu_\ell}) \le c\left(\sqrt{S_N g}\, n_\ell^{-1} + n_\ell^{-2}\right). \tag{24}$$

We define for $x \in \mathbf{R}$, $\sigma = 0,1$,

$$\varrho_{0,\sigma}(x) = \begin{cases} (-1)^\sigma x & \text{if } \ 0 \le (-1)^\sigma x < 1 \\ 0 & \text{otherwise,} \end{cases}$$

and for $\ell = 1, \dots, k-1$,

$$\varrho_{\ell,\sigma}(x) = \begin{cases} (-1)^\sigma 2^{-\ell} x & \text{if } \ 2^{\ell-1} \le (-1)^\sigma x < 2^\ell \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, we let η be the identity on $\mathbf{Z}[0,N)$. Then for each $f \in L_p^N$,

$$g_f^{\ell,\sigma} = \varrho_{\ell,\sigma} \circ f \circ \eta.$$

By Corollary 1 there is an algorithm $A_{\ell,\sigma}$ on L_p^N with

$$n_q(A_{\ell,\sigma}) = n_q(\widetilde{A}_\ell)$$

and

$$A_{\ell,\sigma}(f) = \widetilde{A}_\ell(g_f^{\ell,\sigma}) \tag{25}$$

for all $f \in L_p^N$. We define A as being composed of $A_{\ell,\sigma}$ (in the sense of (11)) as follows:

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

To estimate the error of A, fix any $f \in L_p^N$ and let $\{\zeta_{\ell,\sigma} \,|\, 0 \le \ell \le k,\ \sigma = 0,1\}$ be independent random variables with distribution $A_{\ell,\sigma}(f)$ respectively. Define

$$\zeta = \sum_{\substack{0\le\ell\le k\\ \sigma=0,1}} (-1)^\sigma 2^\ell \zeta_{\ell,\sigma}. \tag{26}$$

It follows from Lemma 2 that

$$A(f) = \operatorname{dist}(\zeta). \tag{27}$$

By (24) and (25), we have, with probability at least $1-2^{-\nu_\ell}$,

$$|S_N g_f^{\ell,\sigma} - \zeta_{\ell,\sigma}| \le c \left(\sqrt{S_N g_f^{\ell,\sigma}}\, n_\ell^{-1} + n_\ell^{-2} \right),$$

and therefore, with probability at least $1 - 2\sum_{\ell=0}^k 2^{-\nu_\ell}$

$$\Big| \sum_{\substack{0\le\ell\le k\\ \sigma=0,1}} (-1)^\sigma 2^\ell (S_N g_f^{\ell,\sigma} - \zeta_{\ell,\sigma}) \Big| \le c \sum_{\substack{0\le\ell\le k\\ \sigma=0,1}} 2^\ell \left(\sqrt{S_N g_f^{\ell,\sigma}}\, n_\ell^{-1} + n_\ell^{-2} \right),$$

hence, by (23) and (26),

$$|S_N f - \zeta| \le c \sum_{\substack{0\le\ell\le k\\ \sigma=0,1}} 2^\ell \left(\sqrt{S_N g_f^{\ell,\sigma}}\, n_\ell^{-1} + n_\ell^{-2} \right) + \Big| \frac{1}{N} \sum_{i\in\mathcal{I}_f^k} f(i) \Big|,$$

which gives together with (27), (19), (22) and (21)

$$\begin{aligned}
&e(S_N, A, f, 2\sum_{\ell=0}^k 2^{-\nu_\ell}) \\
&\le c \sum_{\substack{0\le\ell\le k\\ \sigma=0,1}} 2^\ell \left(\sqrt{S_N g_f^{\ell,\sigma}}\, n_\ell^{-1} + n_\ell^{-2} \right) + 2^{-(p-1)k} \|f\|_{L_p^N}^p \\
&\le c \sum_{\substack{0\le\ell\le k\\ \sigma=0,1}} 2^\ell \left(\sqrt{N^{-1}|\mathcal{J}_f^{\ell,\sigma}|}\, n_\ell^{-1} + n_\ell^{-2} \right) + 2^{-(p-1)k} \|f\|_{L_p^N}^p \qquad (28) \\
&\le 2c \sum_{\ell=1}^k \left(2^{(1-p/2)\ell} n_\ell^{-1} \|f\|_{L_p^N}^{p/2} + 2^\ell n_\ell^{-2} \right) \\
&\quad + 2c n_0^{-1} + 2^{-(p-1)k} \|f\|_{L_p^N}^p \qquad (29)
\end{aligned}$$

(recall the remark about constants after Lemma 7). Moreover, we have

$$n_q(A) \le 2\sum_{\ell=0}^{k} \nu_\ell n_\ell. \tag{30}$$

Now we choose the parameters k, ν_ℓ and n_ℓ in a suitable way and prove the error estimates. First we consider the case $2 < p < \infty$. Here we put

$$k = \left\lceil \frac{1}{p-1} \log n \right\rceil. \tag{31}$$

Define, furthermore, $\nu_\ell = \lceil 2\log(\ell+1)\rceil + 4$, hence

$$2\sum_{\ell=0}^{k} 2^{-\nu_\ell} \le \frac{1}{8}\sum_{\ell=0}^{k}(\ell+1)^{-2} < \frac{1}{4}. \tag{32}$$

Finally, let

$$n_\ell = \left\lceil 2^{(1/2-p/4)\ell} n \right\rceil. \tag{33}$$

This together with (30) implies

$$n_q(A) \le 2\sum_{\ell=0}^{k}(\lceil 2\log(\ell+1)\rceil + 4)\left\lceil 2^{(1/2-p/4)\ell} n\right\rceil \le c_1 n \tag{34}$$

for some constant $c_1 > 0$. It follows from (32), (29), (33) and (31) that

$$\begin{aligned}
&e(S_N, A, f)\\
&\le\ e(S_N, A, f, 2\sum_{\ell=0}^{k} 2^{-\nu_\ell})\\
&\le\ c\sum_{\ell=1}^{k}\left(2^{(1/2-p/4)\ell} n^{-1}\|f\|^{p/2}_{L_p^N} + 2^{p\ell/2} n^{-2}\right) + cn^{-1} + n^{-1}\|f\|^{p}_{L_p^N}\\
&\le\ c\left(n^{-1}\|f\|^{p/2}_{L_p^N} + 2^{(1-p/2)k} n^{-1} + n^{-1} + n^{-1}\|f\|^{p}_{L_p^N}\right)\\
&\le\ cn^{-1}\max(\|f\|^{p}_{L_p^N}, 1).
\end{aligned}$$

Consequently,

$$e(S_N, A, \mathcal{B}_p^N) \le cn^{-1},$$

which together with (34) implies the desired result in the case $2 < p < \infty$. Note that the case $p = \infty$ also follows since $\mathcal{B}_\infty^N \subseteq \mathcal{B}_p^N$ for any $p < \infty$.

Now we suppose $1 < p < 2$. Here we choose

$$k = \left\lceil \frac{2}{p} \log n \right\rceil, \tag{35}$$

$$n_\ell = \left\lceil 2^{-(1/2-p/4)(k-\ell)} n \right\rceil, \tag{36}$$

and $\nu_\ell = \lceil 2 \log(k - \ell + 1) \rceil + 4$, which implies that (32) holds again. Furthermore, by (30),

$$n_q(A) \le 2 \sum_{\ell=0}^{k} (\lceil 2 \log(k - \ell + 1) \rceil + 4) \left\lceil 2^{-(1/2-p/4)(k-\ell)} n \right\rceil \le c_1 n. \tag{37}$$

We get from (32), (29), (36) and (35)

$$\begin{aligned}
&e(S_N, A, f) \\
&\le c \sum_{\ell=1}^{k} \left(2^{(1-p/2)\ell + (1/2-p/4)(k-\ell)} n^{-1} \|f\|_{L_p^N}^{p/2} + 2^{\ell + (1-p/2)(k-\ell)} n^{-2} \right) \\
&\quad + c n_0^{-1} + 2^{-(p-1)k} \|f\|_{L_p^N}^p \\
&\le c \sum_{\ell=1}^{k} \left(2^{(1/2-p/4)(k+\ell)} n^{-1} \|f\|_{L_p^N}^{p/2} + 2^{k - p(k-\ell)/2} n^{-2} \right) \\
&\quad + c\, 2^{(1/2-p/4)k} n^{-1} + 2^{-(p-1)k} \|f\|_{L_p^N}^p \\
&\le c \left(2^{(1-p/2)k} n^{-1} \|f\|_{L_p^N}^{p/2} + 2^k n^{-2} + 2^{(1/2-p/4)k} n^{-1} + 2^{-(p-1)k} \|f\|_{L_p^N}^p \right) \\
&\le c n^{-2(1-1/p)} \max(\|f\|_{L_p^N}^p, 1).
\end{aligned} \tag{38}$$

Now (37) and (38) yield the needed result.

Finally, we consider the case $p = 2$. Here we define

$$n_\ell \equiv n_0 = \lceil n (\log n)^{-1} (\log \log n)^{-1} \rceil \tag{39}$$

(recall that we assumed $n > 2$, so n_0 is well-defined and $n_0 \ge 1$), furthermore

$$k = \lceil \log n_0 \rceil \tag{40}$$

and

$$\nu_\ell \equiv \nu_0 = \lceil \log(k+1) \rceil + 3. \tag{41}$$

It follows that

$$2\sum_{\ell=0}^{k} 2^{-\nu_\ell} \le \frac{1}{4}\sum_{\ell=0}^{k} \frac{1}{k+1} \le \frac{1}{4} \tag{42}$$

and, by (30),

$$n_q(A) \le 2(k+1)\nu_0 n_0 \le c_1 n. \tag{43}$$

By (42) and (28), the error satisfies

$$e(S_N, A, f) \le c \sum_{\substack{1\le\ell\le k\\ \sigma=0,1}} 2^\ell \sqrt{N^{-1}|\mathcal{J}_f^{\ell,\sigma}|}\, n_\ell^{-1} + c\sum_{\ell=1}^{k} 2^\ell n_\ell^{-2} + cn_0^{-1}(\|f\|_{L_2^N}^2 + 1).$$

Hölder's inequality, applied to the first sum, gives

$$\begin{aligned} e(S_N, A, f) &\le c\,(2k)^{1/2}\Big(N^{-1}\sum_{\substack{1\le\ell\le k\\ \sigma=0,1}} 2^{2\ell}|\mathcal{J}_f^{\ell,\sigma}|\Big)^{1/2} n_0^{-1} \\ &\quad + c\sum_{\ell=1}^{k} 2^\ell n_0^{-2} + cn_0^{-1}(\|f\|_{L_2^N}^2 + 1), \end{aligned}$$

and by (20), (39), and (40), we finally get

$$\begin{aligned} e(S_N, A, f) &\le c\left(k^{1/2} n_0^{-1}\|f\|_{L_2^N} + 2^k n_0^{-2} + n_0^{-1}(\|f\|_{L_2^N}^2 + 1)\right) \\ &\le cn^{-1}\log^{3/2} n \log\log n \max(\|f\|_{L_2^N}^2, 1). \end{aligned}$$

This implies the statement for $p = 2$. □

Remark. Since quantum algorithms are not linear, the statement of Theorem 1 does not give any information on $f \in L_p^N$ of norm greater than one. Our proof, however, does. It shows that the algorithm developed for fixed $1 < p < \infty$ and $n, N \in \mathbf{N}$ has the property that for all $f \in L_p^N$

$$e(S_N, A, f) \le c \begin{cases} n^{-1}\max(\|f\|_{L_p^N}^p, 1) & \text{if } \ 2 < p < \infty \\ n^{-1}\log^{3/2} n \log\log n \max(\|f\|_{L_2^N}^2, 1) & \text{if } \ p = 2 \\ n^{-2(1-1/p)}\max(\|f\|_{L_p^N}^p, 1) & \text{if } \ 1 < p < 2. \end{cases}$$

4 Lower Bounds

In this section we derive lower bounds on the quantities $e_n^q(S,F)$ first in the general setting and then for $F=\mathcal{B}_p^N$, $S=S_N$. 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 $f_u\in\mathcal{F}(D,K)$ be assigned such that the following is satisfied:

Condition (I): For each $t\in D$ there is an ℓ, $0\le\ell\le L-1$, 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 $f_u(t)=f_{u'}(t)$.

This type of function system will play a key rôle in our lower bound proofs. Condition (I) is easily seen to be equivalent to the following

Condition (Ia): There are functions $g_0,g_1\in\mathcal{F}(D,K)$ and a decomposition $D=\bigcup_{\ell=0}^{L-1}D_\ell$ with $D_\ell\cap D_{\ell'}=\emptyset$ $(l\neq l')$ such that for $t\in D_\ell$

$$f_u(t)=\begin{cases} g_0(t) & \text{if} \quad u_\ell=0\\ g_1(t) & \text{if} \quad u_\ell=1.\end{cases}$$

The first result is based on the polynomial method by Beals, Buhrman, Cleve, and Mosca (1998) and extends their Lemma 4.1 to our general setting.

Lemma 9. *Let $L\in\mathbf{N}$, let $(f_u)_{u\in\{0,1\}^L}\subseteq\mathcal{F}(D,K)$ be a system of functions satisfying condition (I), and let A be a quantum algorithm on $\mathcal{F}(D,K)$ without measurement, $m=m_q(A)$, $n=n_q(A)$. Then for all $x,b\in\mathbf{Z}[0,2^m)$, $A_{f_u}(x,b)$ (defined in (3) and (4)), considered as a function of u, is a complex multilinear polynomial in the variables $u_0,\dots,u_{L-1}$ of degree at most n.*

Proof. Let

$$A=(Q,(U_j)_{j=0}^n),\qquad Q=(m,m',m'',Z,\tau,\beta).$$

Fix $b\in\mathbf{Z}[0,2^m)$ and define w_j and $p_j(x,u)$ for $j=0,\dots,n$ by

$$w_j=U_jQ_{f_u}U_{j-1}Q_{f_u}\dots U_1Q_{f_u}U_0b=\sum_{x\in\mathbf{Z}[0,2^m)}p_j(x,u)\,|x\rangle\,.$$

Then

$$p_n(x,u)=A_{f_u}(x,b). \tag{44}$$

Of course, $p_0(x,u)$ are constants, so polynomials of degree 0 in u. Now we proceed by induction over j. Assume that for some j, $0 \le j < n$, the $p_j(x,u)$ are polynomials of degree $\le j$ in u. Define $q_j(x,u)$ by

$$Q_{f_u} w_j = \sum_{x \in \mathbf{Z}[0,2^m)} q_j(x,u)\, |x\rangle \,.$$

Since

$$Q_{f_u} w_j = Q_{f_u} \sum_{x \in \mathbf{Z}[0,2^m)} p_j(x,u)\, |x\rangle = \sum_{x \in \mathbf{Z}[0,2^m)} p_j(x,u)\, Q_{f_u}\, |x\rangle \,,$$

and since Q_{f_u} is a bijection on the basis states, we get

$$q_j(x,u) = p_j(Q_{f_u}^{-1} x, u).$$

Now fix $x \in \mathbf{Z}[0,2^m)$. Represent $|x\rangle$ as $|i\rangle\,|y\rangle\,|z\rangle$ with $i \in \mathbf{Z}[0,2^{m'})$, $y \in \mathbf{Z}[0,2^{m''})$ and $z \in \mathbf{Z}[0,2^{m-m'-m''})$. According to the query definition (2), we have $Q_{f_u}\,|x\rangle = |i\rangle\,|y\rangle\,|z\rangle$ if $i \notin Z$. Hence, in this case $q_j(x,u) = p_j(x,u)$, so $\deg q_j(x,\,\cdot\,) \le j$. If $i \in Z$,

$$Q_{f_u}^{-1}\,|x\rangle = |i\rangle\,|y \ominus \beta(f_u(\tau(i)))\rangle\,|z\rangle \,.$$

Let, according to condition (I) above, ℓ be such that $0 \le \ell \le L-1$ and $f_u(\tau(i))$ depends only on u_ℓ. We denote $f_u(\tau(i)) = s_0$ for $u_\ell = 0$ and $f_u(\tau(i)) = s_1$ for $u_\ell = 1$. It follows that

$$Q_{f_u}^{-1}\,|x\rangle = |i\rangle\,|y \ominus \beta(f_u(\tau(i)))\rangle\,|z\rangle = \begin{cases} |i\rangle\,|y \ominus \beta(s_0)\rangle\,|z\rangle := x_0 & \text{if} \quad u_\ell = 0 \\ |i\rangle\,|y \ominus \beta(s_1)\rangle\,|z\rangle := x_1 & \text{if} \quad u_\ell = 1. \end{cases}$$

Consequently,

$$q_j(x,u) = p_j(Q_{f_u}^{-1} x, u) = (1-u_\ell)p_j(x_0,u) + u_\ell p_j(x_1,u),$$

which implies $\deg q_j(x,\,\cdot\,) \le j+1$. Now

$$w_{j+1} = U_{j+1} Q_{f_u} w_j = U_{j+1} \sum_{y \in \mathbf{Z}[0,2^m)} q_j(y,u)\,|y\rangle \,,$$

which gives

$$p_{j+1}(x,u) = \sum_{y \in \mathbf{Z}[0,2^m)} U_{j+1}(x,y) q_j(y,u),$$

where $(U_{j+1}(x,y))_{x,y\in\mathbf{Z}[0,2^m)}$ is the matrix of the transformation U_{j+1} in the canonical basis. Since the $U_{j+1}(x,y)$ are scalars not depending on u, and since $\deg q_j(x,\,\cdot\,) \le j+1$, it follows that $\deg p_{j+1}(x,\,\cdot\,) \le j+1$. This completes the induction and shows that $\deg p_n(x,\,\cdot\,) \le n$. Now the lemma follows from (44) and the observation that, since the u_i take only the values 0 and 1, we can replace any polynomial by a multilinear one without changing its values on $\{0,1\}^L$. □

Corollary 2. *Let $L \in \mathbf{N}$ and assume that $(f_u)_{u\in\{0,1\}^L} \subseteq \mathcal{F}(D,K)$ satisfies condition (I). Let A be a quantum algorithm from $\mathcal{F}(D,K)$ to a normed space G. Then for each subset $C \subseteq G$,*

$$p(u) = A(f_u)\{C\}$$

is a real multilinear polynomial of degree at most $2n_q(A)$.

Proof. This follows readily from Lemma 9 and relations (7) and (9). □

The next lemma is based on the results of Nayak and Wu (1999). To state it, we introduce some further notation. Define the function $\varrho(L,\ell,\ell')$ for $L \in \mathbf{N}$, $0 \le \ell \ne \ell' \le L$ by

$$\varrho(L,\ell,\ell') = \sqrt{\frac{L}{|\ell-\ell'|}} + \frac{\min_{j=\ell,\ell'}\sqrt{j(L-j)}}{|\ell-\ell'|}. \tag{45}$$

Note that $j(L-j) = (L/2)^2 - (L/2-j)^2$, so this expression is minimized iff $|L/2 - j|$ is maximized. For $u \in \{0,1\}^L$ set $|u| = \sum_{\ell=0}^{L-1} u_\ell$.

Lemma 10. *There is a constant $c_0 > 0$ such that the following holds: Let D, K be nonempty sets, let $F \subseteq \mathcal{F}(D,K)$ be a set of functions, G a normed space, $S: F \to G$ a function, and $L \in \mathbf{N}$. Suppose $(f_u)_{u\in\{0,1\}^L} \subseteq \mathcal{F}(D,K)$ is a system of functions satisfying condition (I). Let finally $0 \le \ell \ne \ell' \le L$ and assume that*

$$f_u \in F \quad \text{whenever} \quad |u| \in \{\ell,\ell'\}. \tag{46}$$

Then

$$e_n^q(S,F) \ge \frac{1}{2}\min\left\{\|S(f_u) - S(f_{u'})\| \;\middle|\; |u| = \ell,\ |u'| = \ell'\right\} \tag{47}$$

for all n with

$$n \le c_0\,\varrho(L,\ell,\ell'). \tag{48}$$

Proof. Nayak and Wu (1999, Theorem 1.1) showed that there is a constant $c > 0$ such that for all $L \in \mathbf{N}$ and $0 \le \ell \ne \ell' \le L$ the following holds: If p is an L-variate real polynomial such that

$$-1/4 \le p(u) \le 5/4 \quad \text{for all} \quad u \in \{0,1\}^L,$$

$$3/4 \le p(u) \le 5/4 \quad \text{if} \quad u \in \{0,1\}^L,\ |u| = \ell,$$

and

$$-1/4 \le p(u) \le 1/4 \quad \text{if} \quad u \in \{0,1\}^L,\ |u| = \ell',$$

then

$$\deg p \ge c\varrho(L, \ell, \ell'), \tag{49}$$

where ϱ was defined in (45). Denote for $j = \ell, \ell'$

$$G_j = \{S(f_u) \,|\, |u| = j\} \tag{50}$$

and

$$\delta = d(G_\ell, G_{\ell'}), \tag{51}$$

where for $X, Y \subseteq G$,

$$d(X, Y) = \inf_{x \in X,\, y \in Y} \|x - y\|.$$

(For $x \in G$ we write $d(x, G)$ instead of $d(\{x\}, G)$.) Now let A be any quantum algorithm from F to G with $n_q(A) = n$ and

$$e(S, A, F) < \delta/2. \tag{52}$$

As we mentioned after the definition, a quantum algorithm on F is always also a quantum algorithm on $\mathcal{F}(D, K)$. For each $u \in \{0,1\}^L$, let ζ_u be a random variable with distribution $A(f_u)$. Define

$$p(u) = A(f_u)\{g \in G \,|\, d(g, G_\ell) < \delta/2\} = \mathbf{P}\{d(\zeta_u, G_\ell) < \delta/2\}.$$

It follows that

$$0 \le p(u) \le 1 \tag{53}$$

and, by Corollary 2, p is a real polynomial satisfying

$$\deg p \le 2n. \tag{54}$$

Because of (46) and (52), we have for $|u| = \ell$,

$$\begin{aligned} 3/4 &\leq \mathbf{P}\{\|S(f_u) - \zeta_u\| < \delta/2\} \\ &\leq \mathbf{P}\{d(\zeta_u, G_\ell) < \delta/2\} = p(u). \end{aligned} \tag{55}$$

On the other hand, for $|u| = \ell'$,

$$\begin{aligned} 1/4 &\geq \mathbf{P}\{\|S(f_u) - \zeta_u\| \geq \delta/2\} \\ &\geq \mathbf{P}\{d(\zeta_u, G_{\ell'}) \geq \delta/2\} \\ &\geq \mathbf{P}\{d(\zeta_u, G_\ell) < \delta/2\} = p(u). \end{aligned} \tag{56}$$

From (53 – 56) and (49), we infer

$$2n \geq \deg p \geq c\varrho(L, \ell, \ell').$$

Now choose any $c_0 < c/2$. Then $n \leq c_0\varrho(L, \ell, \ell')$ implies $e_n^q(S, F) \geq \delta/2$, which, because of (50) and (51), is the same as (47).

□

The following theorem is the main result of this section. The case $p = \infty$ is due to Nayak and Wu (1999), and the case $2 \leq p < \infty$ is a direct consequence. For the sake of completeness we include this part in the proof below. (Another reason for this is that we use a slightly more general notion of query, so this way we formally check that their bound holds true also for our model.)

Theorem 2. *Let $1 \leq p \leq \infty$. Then there are constants $c_0, c_1, c_2 > 0$ such that for $n, N \in \mathbf{N}$,*

$$e_n^q(S_N, \mathcal{B}_p^N) \geq c_2 \begin{cases} n^{-2(1-1/p)} & \text{if} \quad 1 \leq p < 2 \quad \text{and} \quad n \leq c_0\sqrt{N} \\ n^{-1} & \text{if} \quad 2 \leq p \leq \infty \quad \text{and} \quad n \leq c_1 N. \end{cases}$$

Proof. Let c_0 be the constant from Lemma 10. Let $1 \leq p < 2$ and

$$n \leq c_0\sqrt{N}. \tag{57}$$

Define

$$L = \lceil c_0^{-2} n^2 \rceil, \quad \ell = 0, \quad \ell' = 1.$$

It follows from (57) that $1 \leq L \leq N$. Moreover,

$$n \leq c_0\sqrt{L} = c_0\varrho(L, \ell, \ell') \tag{58}$$

and

$$L < c_0^{-2} n^2 + 1 \le (c_0^{-2} + 1) n^2. \tag{59}$$

Put $M = \lfloor L^{-1} N \rfloor$. Hence $1 \le M \le N$ and

$$M \le L^{-1} N \le 2M. \tag{60}$$

Define $\psi_j \quad (j = 0, \dots, L-1)$ by

$$\psi_j(i) = \begin{cases} (N/M)^{1/p} & \text{if} \quad jM \le i < (j+1)M \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\psi_j \in \mathcal{B}_p^N$ and

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

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

$$f_u = \sum_{j=0}^{L-1} u_j \psi_j. \tag{61}$$

Since the functions ψ_j have disjoint supports, the system $(f_u)_{u \in \{0,1\}^L}$ satisfies condition (I). Lemma 10 and relation (58) together with (60) and (59) give

$$\begin{aligned} e_n^q(S_N, \mathcal{B}_p^N) &\ge \frac{1}{2} \min \left\{ |S_N f_u - S_N f_{u'}| \,\middle|\, |u| = 0,\ |u'| = 1 \right\} \\ &= \frac{1}{2} \left(M N^{-1} \right)^{1-1/p} \ge \frac{1}{2} (2L)^{-(1-1/p)} \ge c_2 n^{-2(1-1/p)} \end{aligned}$$

for some constant $c_2 > 0$. This proves the statement in the first case.

Now we consider the case $2 \le p \le \infty$. Since $\mathcal{B}_\infty^N \subset \mathcal{B}_p^N$ whenever $p < \infty$, it suffices to prove the lower bound for $p = \infty$. We set $c_1 = 2^{-1}(c_0^{-1} + 2)^{-1}$ and assume $n \le c_1 N$. Let

$$L = 2 \left\lceil c_0^{-1} n + 1 \right\rceil, \quad \ell = L/2 - 1, \quad \ell' = \ell + 1 = L/2.$$

It follows that $L \ge 4$ and

$$\varrho(L, \ell, \ell') > \min_{j = l, l'} \sqrt{j(L-j)} = \sqrt{L^2/4 - 1} \ge c_0^{-1} n. \tag{62}$$

Moreover, since $1 \le n \le c_1 N$, we get

$$L = 2\left\lceil c_0^{-1} n + 1 \right\rceil \le 2(c_0^{-1} n + 2) \le 2(c_0^{-1} + 2)n \le N. \tag{63}$$

Now let $M = \lfloor L^{-1} N \rfloor$, then (60) holds again. Set

$$\psi_j(i) = \begin{cases} 1 & \text{if} \quad jM \le i < (j+1)M \\ 0 & \text{otherwise} \end{cases}$$

for $j = 0, \dots, L-1$, and let f_u be again defined by (61). Clearly, $(f_u)_{u \in \{0,1\}^L}$ satisfies condition (I) and $f_u \in \mathcal{B}_\infty^N$ for all $u \in \{0,1\}^L$. Lemma 10 together with relations (62), (60) and (63) gives

$$\begin{aligned} e_n^q(S_N, \mathcal{B}_\infty^N) &\ge \frac{1}{2} \min\left\{ |S_N f_u - S_N f_{u'}| \,\middle|\, |u| = \ell,\ |u'| = \ell + 1 \right\} \\ &= \frac{1}{2} M N^{-1} \ge \frac{1}{4L} \ge c_2 n^{-1} \end{aligned}$$

for some $c_2 > 0$. □

Remark. Comparing Theorem 2 with Theorem 1, we see that matching upper and lower bounds were obtained except for the case of $1 \le p < 2$, $n \ge c_0 \sqrt{N}$. This case is settled in Heinrich and Novak (2001b).

5 Integration in $L_p([0,1]^d)$

Here we present an application of the summation results to integration of functions. Further results will be contained in Heinrich (2001). Let $1 \le p \le \infty$, $d \in \mathbf{N}$, $D = [0,1]^d$ and let $L_p(D)$ denote the usual space of p-integrable with respect to the Lebesgue measure functions on D, equipped with the norm

$$\|f\|_{L_p(D)} = \left(\int_D |f(t)|^p \, dt \right)^{1/p}$$

if $p < \infty$ and

$$\|f\|_{L_\infty(D)} = \operatorname{ess\,sup}_{t \in D} |f(t)|.$$

Let $I_d : L_p(D) \to \mathbf{R}$ be the integration operator, defined for $f \in L_p(D)$ by

$$I_d f = \int_D f(t) \, dt.$$

In this chapter we will consider $G = \mathbf{R}$ and $S = I_d$. We want to integrate functions from the unit ball $\mathcal{B}(L_p(D))$ in the quantum model of computation developed in section 2. Strictly speaking, $L_p(D)$ consists of equivalence classes of functions being equal almost everywhere. Hence, function values are not well-defined, in general. This changes, however, if we consider subsets of $L_p(D)$ which consist of continuous functions, or more precisely, of equivalence classes which contain a (unique) continuous function. This is how we shall approach the integration problem – we study it for certain subsets $\mathcal{E} \subset \mathcal{B}(L_p(D))$. We shall assume that $\mathcal{E}$ is an equicontinuous set of functions on D. Since D is compact, equicontinuity is equivalent to uniform equicontinuity, and the latter means that for each $\varepsilon > 0$ there is a $\delta > 0$ such that for $s, t \in D$, $\|s - t\|_\infty \le \delta$ implies $|f(s) - f(t)| \le \varepsilon$ for all $f \in F$. Note also that it follows readily from the Arzelà-Ascoli theorem that $\mathcal{E} \subset \mathcal{B}(L_p(D))$ is equicontinuous iff $\mathcal{E}$ is relatively compact in the space $C(D)$ of continuous functions on D, equipped with the sup-norm. (A similar approach was chosen in Novak, 1988, to discuss restricted Monte Carlo methods.)

Theorem 3. *Let $1 \le p \le \infty$. Then there are constants $c_1, c_2 > 0$ such that for all $d, n \in \mathbf{N}$*

$$\begin{array}{rcccll}
c_1 n^{-1} & \le & \sup\limits_{\mathcal{E}\subset\mathcal{B}(L_p(D))} e_n^q(I_d, \mathcal{E}) & \le & c_2 n^{-1} & 2 < p \le \infty \\
c_1 n^{-1} & \le & \sup\limits_{\mathcal{E}\subset\mathcal{B}(L_2(D))} e_n^q(I_d, \mathcal{E}) & \le & c_2 n^{-1} \log^{3/2} n \log\log n & \\
c_1 n^{-2(1-1/p)} & \le & \sup\limits_{\mathcal{E}\subset\mathcal{B}(L_p(D))} e_n^q(I_d, \mathcal{E}) & \le & c_2 n^{-2(1-1/p)} & 1 \le p < 2,
\end{array}$$

where the supremum is taken over all equicontinuous subsets $\mathcal{E}$ of $\mathcal{B}(L_p(D))$.

Proof. First we prove the upper bounds. Let $\mathcal{E} \subset \mathcal{B}(L_p(D))$ be equicontinuous and let $n \in \mathbf{N}$. For $k \in \mathbf{N}$ let

$$D = \bigcup_{i=0}^{2^{dk}-1} D_i$$

be the partition of D into 2^{dk} congruent cubes of disjoint interior. Let s_i be the point in D_i with the smallest Euclidean norm. Let P_k be the operator of piecewise constant interpolation with respect to the partition $(D_i)_{i=0}^{2^{dk}-1}$ in the points $(s_i)_{i=0}^{2^{dk}-1}$ (to avoid ambiguity, if a point belongs to more than one of the sets D_i, we assign to it the value $f(s_i)$ for the smallest such i). Due to the equicontinuity of $\mathcal{E}$ there is a $k \in \mathbf{N}$ such that

$$\|f - P_k f\|_{L_\infty(D)} \le n^{-1} \tag{64}$$

for all $f \in \mathcal{E}$. Fix this k and put $N = 2^{dk}$. It follows that

$$\sup_{f\in\mathcal{E}} |I_d f - I_d(P_k f)| \le n^{-1}. \tag{65}$$

Moreover, defining

$$\Gamma : \mathcal{E} \to L_p^N \quad \text{by} \quad \Gamma(f)(i) = f(s_i) \quad (i = 0, \dots, N-1),$$

we get

$$I_d(P_k f) = \frac{1}{N}\sum_{i=0}^{N-1} f(s_i) = S_N \circ \Gamma(f). \tag{66}$$

Note that for $f \in \mathcal{E} \subset \mathcal{B}(L_p(D))$

$$\begin{aligned}\left(\frac{1}{N}\sum_{i=0}^{N-1} |\Gamma(f)(i)|^p\right)^{1/p} &= \left(\int_D |P_k f(s)|^p ds\right)^{1/p} = \|P_k f\|_{L_p(D)} \\ &\le \|f\|_{L_p(D)} + \|f - P_k f\|_{L_p(D)} \\ &\le \|f\|_{L_p(D)} + n^{-1} \le 2.\end{aligned}$$

Consequently, Γ maps $\mathcal{E}$ into $2\mathcal{B}_p^N$. Lemma 6, Corollary 1 and relations (65) and (66) imply

$$\begin{aligned} e_n^q(I_d, \mathcal{E}) &\le n^{-1} + e_n^q(S_N \circ \Gamma, \mathcal{E}) \le n^{-1} + e_n^q(S_N, 2\mathcal{B}_p^N) \\ &= n^{-1} + 2e_n^q(S_N, \mathcal{B}_p^N),\end{aligned}$$

hence Theorem 1 yields the upper bounds.

To verify the lower bounds, fix a σ with $0 < \sigma < 1$ and let ψ be a continuous function on $\mathbf{R}^d$ with

$$\operatorname{supp}\psi \subseteq [0,1]^d, \quad 0 \le \psi \le 1 \quad \text{and} \quad I_d\psi = \sigma.$$

Fix $n \in \mathbf{N}$ and choose $N = 2^{dk}$ in such a way that

$$c_0\sqrt{N} \ge n \quad \text{and} \quad c_1 N \ge n, \tag{67}$$

where c_0 and c_1 are the constants from Theorem 2. Set

$$\psi_i(t) = \psi(2^k(t - s_i)) \quad (i = 0, \dots, N-1),$$

with the s_i as in the preceding part of the proof. Consequently,

$$I_d\psi_i = 2^{-dk} I_d\psi = \sigma 2^{-dk} = \sigma N^{-1}. \tag{68}$$

Define

$$\Gamma : \mathcal{B}_p^N \to L_p(D) \quad \text{by} \quad \Gamma(f) = \sum_{i=0}^{N-1} f(i)\psi_i.$$

For $f \in \mathcal{B}_p^N$,

$$\begin{aligned} \|\Gamma(f)\|_{L_p(D)} &= \int_D \sum_{i=0}^{N-1} |f(i)|^p |\psi_i(t)|^p dt = \sum_{i=0}^{N-1} |f(i)|^p \int_D |\psi_i(t)|^p dt \\ &= 2^{-dk} \sum_{i=0}^{N-1} |f(i)|^p \int_D |\psi(t)|^p dt \le N^{-1} \sum_{i=0}^{N-1} |f(i)|^p \le 1. \end{aligned}$$

We define $\mathcal{E} = \Gamma(\mathcal{B}_p^N)$, which is a subset of $\mathcal{B}(L_p(D))$. Since the functions ψ_i are continuous, and $|f(i)| \le N^{1/p}$ for all $f \in \mathcal{B}_p^N$, the equicontinuity of $\mathcal{E}$ easily follows. Furthermore,

$$I_d \circ \Gamma(f) = I_d \sum_{i=0}^{N-1} f(i)\psi_i = \sum_{i=0}^{N-1} f(i) I_d\psi_i = \sigma N^{-1} \sum_{i=0}^{N-1} f(i) = \sigma S_N f.$$

Lemma 6 and Corollary 1 give

$$\sigma e_n^q(S_N, \mathcal{B}_p^N) = e_n^q(\sigma S_N, \mathcal{B}_p^N) \le e_n^q(I_d, \mathcal{E}),$$

and the result follows from relation (67) and Theorem 2. □

6 Comments

Our results were formulated in the language of information-based complexity theory – the minimal error at given cost (number of function values, functionals etc., in our case queries). Lower bounds in terms of the number of queries mean the more that no algorithm can have better arithmetic (bit) cost. On the other hand, if we have upper bounds on the number of queries, this does not necessarily mean a corresponding estimate of the cost in the bit model. However, for the problems considered in this paper we encounter a situation which is largely parallel to the experience in information-based complexity: As a rule, the developed algorithms, which are optimal in the query sense, show a similar behaviour (usually up to certain logarithmic terms) in their arithmetic (bit) cost. Let us have a closer look at our algorithms from this point of view.

The bit cost of one query of the type (1) we define to be $m' + m''$ (the number of bits to be processed). When we consider the bit cost, let us assume that both N and n are powers of two, which is no loss of generality since the other cases can be reduced to that. We also assume $n < N$, see the remarks before Theorem 1. The algorithm from Lemma 7 makes one measurement and can be implemented on $\mathcal{O}(\log N)$ qubits using $\mathcal{O}(n \log N)$ quantum gates. The algorithm of Lemma 8 requires $\mathcal{O}(\log N)$ qubits, $\mathcal{O}(\nu n \log N)$ gates and makes $\mathcal{O}(\nu)$ measurements. Finally, the algorithm from Theorem 1 needs $\mathcal{O}(\log N)$ qubits, $\mathcal{O}(n \log N)$ gates and $\mathcal{O}(\log n \log\log n)$ measurements for $p < \infty$ (one measurement if $p = \infty$).

To discuss the algorithm of Theorem 3, let us introduce the following quantity for an equicontinuous subset $\mathcal{E} \subset \mathcal{B}(L_p(D))$ and $\varepsilon > 0$:

$$\begin{aligned}\kappa(\mathcal{E}, \varepsilon) \quad = \quad & \min\{k \in \mathbf{N} \mid \\ & |f(s) - f(t)| \le \varepsilon \text{ whenever } f \in \mathcal{E},\ s, t \in D, \|s - t\|_\infty \le 2^{-k}\}.\end{aligned}$$

Then for a given $\mathcal{E} \subset \mathcal{B}(L_p(D))$ we have to compute the mean of $N = 2^{dk}$ numbers, where it suffices to take $k = \kappa(\mathcal{E}, 1/n)$. If $N \le n$, this can be done with $\mathcal{O}(\log n)$ qubits, $\mathcal{O}(N \log n)$ gates and one measurement (see the remarks before Theorem 1). If $n < N$, this can be implemented on $\mathcal{O}(d\,\kappa(\mathcal{E}, 1/n))$ qubits, with $\mathcal{O}(dn\,\kappa(\mathcal{E}, 1/n))$ gates and $\mathcal{O}(\log n \log\log n)$ measurements for $p < \infty$ and one measurement for $p = \infty$. (The constants in the $\mathcal{O}$-notation do not depend on $\mathcal{E}$ and d.)

Next let us compare the results obtained above to the classical deterministic and Monte Carlo setting. We denote the respective quantities by e_n^{det} and e_n^{mc}. This discussion is carried out in greater detail in Heinrich and Novak (2001a), where also the related definitions and references can be found. The following table contains the order of the respective quantities, that is, the behaviour up to constants. We also omitted the additional logarithmic factor in the case $p = 2$. Furthermore, we assume for the case $\mathcal{B}_p^N$ that $n \le c_1 N$, where in the classical settings, c_1 is any constant with $0 < c_1 < 1$, while in the quantum setting for $2 \le p \le \infty$, c_1 is the constant from Theorem 2. Moreover in the quantum setting for $1 \le p < 2$, we assume $n \le c_0\sqrt{N}$, with c_0 from Theorem 2, as well. Finally, when we write $\mathcal{B}_{L_p}$, we mean (in all three setttings) the supremum over all equicontinuous subsets $\mathcal{E} \subset \mathcal{B}(L_p([0,1]^d))$ as in the previous section.

	e_n^{det}	e_n^{mc}	e_n^q
$\mathcal{B}_p^N$, $2 \le p \le \infty$	1	$n^{-1/2}$	n^{-1}
$\mathcal{B}_p^N$, $1 \le p < 2$	1	$n^{-1+1/p}$	$n^{-2+2/p}$
$\mathcal{B}_{L_p}$, $2 \le p \le \infty$	1	$n^{-1/2}$	n^{-1}
$\mathcal{B}_{L_p}$, $1 \le p < 2$	1	$n^{-1+1/p}$	$n^{-2+2/p}$

The result on $\mathcal{B}_{L_p}$ in the randomized setting can be found in Heinrich (1993). The respective statement for the deterministic setting is easily derived using standard methods of information-based complexity theory. A little further below we indicate the proof of a somewhat stronger result.

It might be illustrative to formulate the results in terms of complexity. Here we impose the corresponding restrictions. We always assume $\varepsilon \le \varepsilon_0$ for some constant $\varepsilon_0 > 0$. In the quantum setting, the case $1 < p < 2$ holds only for $N \ge c(1/\varepsilon)^{p/(p-1)}$, for some constant $c > 0$. Again, the case $p = 2$ holds up to logarithmic terms.

	$\mathrm{comp}_\varepsilon^{det}$	$\mathrm{comp}_\varepsilon^{mc}$	$\mathrm{comp}_\varepsilon^q$
$\mathcal{B}_p^N$, $2 \le p \le \infty$	N	$\min((1/\varepsilon)^2, N)$	$\min((1/\varepsilon), N)$
$\mathcal{B}_p^N$, $1 < p < 2$	N	$\min((1/\varepsilon)^{p/(p-1)}, N)$	$\min((1/\varepsilon)^{p/(2(p-1))}, N)$
$\mathcal{B}_{L_p}$, $2 \le p \le \infty$	∞	$(1/\varepsilon)^2$	$(1/\varepsilon)$
$\mathcal{B}_{L_p}$, $1 \le p < 2$	∞	$(1/\varepsilon)^{p/(p-1)}$	$(1/\varepsilon)^{p/(2(p-1))}$

In the case $\mathcal{B}_{L_1}$ we have ∞ in all three settings. For $\mathcal{B}_1^N$ we have N in both classical settings, while in the quantum setting our results give the lower bound $\sqrt{N}$ and the (trivial) upper bound N. The question of the correct order of complexity in this case is answered in Heinrich and Novak (2001b).

We see that for the problems considered here quantum algorithms reach a quadratic speedup over classical randomized ones and – at least as far as the pure number of queries is concerned (disregarding the bit cost and number of qubits) – an arbitrarily large speedup over classical deterministic algorithms. Let us discuss this last point in some more detail and also address the bit issue again. Namely, we show that there are equicontinuous sets $\mathcal{E}$ in $\mathcal{B}(L_\infty([0,1]))$ with arbitrarily slowly decreasing $e_n^{det}(I_1,\mathcal{E})$. More precisely, for any sequence $(\varepsilon_n)_{n\in\mathbf{N}}$ with

$$0<\varepsilon_n\le 1, \quad \varepsilon_{n+1}\le\varepsilon_n\le 2\varepsilon_{2n}, \quad \text{and} \quad \lim_{n\to\infty}\varepsilon_n=0 \tag{69}$$

there is an equicontinuous set $\mathcal{E}\subset\mathcal{B}(L_\infty([0,1]))$ such that for all $n\in\mathbf{N}$

$$e_n^{det}(I_1,\mathcal{E})\ge\varepsilon_n/32. \tag{70}$$

Indeed, we define $\mathcal{E}$ as the set of functions f on $[0,1]$ such that for all $k\in\mathbf{N}$ and $s,t\in[0,1]$, $|s-t|\le 2^{-k}$ implies $|f(s)-f(t)|\le\varepsilon_{2^k}$. Let

$$\psi(t)=\begin{cases} t & \text{if } \ 0\le t\le 1/2 \\ (1-t) & \text{if } \ 1/2<t\le 1 \\ 0 & \text{otherwise,}\end{cases}$$

and put for $k\in\mathbf{N}$ and $0\le i\le 2^k-1$

$$\psi_{k,i}(t)=\varepsilon_{2^k}\psi(2^k(t-2^{-k}i)).$$

It is easily checked that for any $\alpha_i\in\{-1,1\}$ $\quad(i=0,\dots,2^k-1)$,

$$\sum_{i=0}^{2^k-1}\alpha_i\psi_{k,i}\in\mathcal{E}$$

and $I_1\psi_{k,i}=2^{-(k+2)}\varepsilon_{2^k}$. A standard argument from the deterministic setting of information-based complexity theory (see e.g. Novak, 1988, Prop. 1.3.5 b) yields

$$e_{2^{k-1}}^{det}(I_1,\mathcal{E})\ge\varepsilon_{2^k}/8\ge\varepsilon_{2^{k-1}}/16,$$

which implies (70). Recall, on the other hand, that by Theorem 3, $e_n^q(I_1,\mathcal{E})\le cn^{-1}$.

Now let us turn to the bit cost. We show that an exponential speedup is possible. Fix any γ with $0<\gamma\le 1$. We choose $\varepsilon_1=1$ and $\varepsilon_n=(\log n)^{-\gamma}$ $(n>1)$. This sequence satisfies (69). Let $\mathcal{E}\subset\mathcal{B}(L_\infty([0,1]))$ be the corresponding set constructed above, so that

$$e_n^{det}(I_1,\mathcal{E})\ge(\log n)^{-\gamma}/32 \quad (n>1),$$

which means that for any ε with $0<\varepsilon\le 1/32$ we need at least $2^{(1/(32\varepsilon))^{1/\gamma}}$, that is, exponentially many operations to reach error ε deterministically. By the construction of the set $\mathcal{E}$ we have

$$\kappa(\mathcal{E},1/n)\le\lceil n^{1/\gamma}\rceil,$$

which implies, by the discussion at the beginning of this section, that in the quantum setting, an error of ε can be reached with $\mathcal{O}(1/\varepsilon)$ queries, one measurement, $\mathcal{O}((1/\varepsilon)^{1/\gamma})$ qubits and $\mathcal{O}((1/\varepsilon)^{1/\gamma+1})$ gates, that is, with polynomial total cost.

Finally we discuss a topic concerning the relations to information-based complexity. A look at our notion of a query might lead to the impression that it covers only what is called standard information, that is, function values of f, while in information-based complexity also more general types of information are considered (e.g. arbitrary linear functionals or scalar products with certain basis functions). This could be relevant not only in finite element methods, but also in the case that function values are not well-defined. Let us show how our approach covers also this situation.

So let F and K be nonempty sets, $S : F \to G$ be a mapping from F to a normed space G and let Λ be a nonempty set of mappings from F to K. We seek to approximate S again, but now the algorithm is supposed to use information about $f \in F$ of the form $\lambda(f)$ for $\lambda \in \Lambda$. Let us define a Λ-based quantum algorithm from F to G to be simply a quantum algorithm A from $\mathcal{F}(\Lambda, K)$ to G. Introduce the mapping

$$\Psi : F \to \mathcal{F}(\Lambda, K)$$

defined for $f \in F$ by

$$\Psi(f)(\lambda) = \lambda(f) \quad (\lambda \in \Lambda).$$

The error of A at $f \in F$ is defined as follows. Let ζ be a random variable with distribution $A(\Psi(f))$. Put

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

Various further quantities like $e(S, A, F)$, $e_n^q(S, F, \Lambda)$ etc. can be defined on this basis as in section 2. The results of section 2 as well as the general results of section 4 remain valid for this situation if formulated appropriately, that is, if applied to A as an algorithm from $\mathcal{F}(\Lambda, K)$ to G. The resulting form of the unitary mappings associated with the query is worth mentioning: Let Q be one of the queries being part of A. Since A is an algorithm on $\mathcal{F}(\Lambda, K)$, its queries have the form (1), where everything is as specified there except that

$$\tau : Z \to \Lambda.$$

Let us denote $\lambda_i = \tau(i)$ for $i \in Z$. Then an element $f \in F$ gives rise to the following unitary operator implementing the query

$$Q_{\Psi(f)}\,|i\rangle\,|x\rangle\,|y\rangle = \begin{cases} |i\rangle\,|x \oplus \beta(\lambda_i(f))\rangle\,|y\rangle & \text{if} \quad i \in Z \\ |i\rangle\,|x\rangle\,|y\rangle & \text{otherwise.} \end{cases}$$

Acknowledgements. I am grateful for stimulating discussions with Erich Novak and Henryk Woźniakowski on quantum computing. Parts of this work were done while I was visiting the Department of Computer Science of the Columbia University, New York, and the Department of Mathematics of the Hong Kong Baptist University. I thank Joe Traub, Henryk Woźniakowski and Fred Hickernell for their hospitality.

References

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

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

[3] M. Boyer, P. Brassard, P. Høyer, and A. Tapp (1998): Tight bounds on quantum searching, Fortschritte der Physik **46**, 493 – 505, see also http://arXiv.org/abs/quant-ph/9605034.

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

[5] G. Brassard, P. Høyer, and A. Tapp (1998): Quantum counting. Lect. Notes in Comp. Science **1443**, 820 – 831, see also http://arXiv.org/abs/quant-ph/9805082.

[6] A. Ekert, P. Hayden, and H. Inamori (2000): Basic concepts in quantum computation. See http://arXiv.org/abs/quant-ph/0011013.

[7] L. Grover (1996): A fast quantum mechanical algorithm for database search. Proc. 28 Annual ACM Symp. on the Theory of Computing, 212–219, ACM Press New York. See also http://arXiv.org/abs/quant-ph/9605043.

[8] L. Grover (1998): A framework for fast quantum mechanical algorithms. Proc. 30 Annual ACM Symp. on the Theory of Computing, 53–62, ACM Press New York. See also http://arXiv.org/abs/quant-ph/9711043.

[9] J. Gruska (1999): Quantum Computing. McGraw-Hill, London.

[10] S. Heinrich (1993): Random approximation in numerical analysis. In: K. D. Bierstedt, A. Pietsch, W. M. Ruess, and D. Vogt, editors, Functional Analysis, 123 – 171, Marcel Dekker.

[11] S. Heinrich (2001): Quantum integration in Sobolev classes (in preparation).

[12] S. Heinrich and E. Novak (2001a): Optimal summation and integration by deterministic, randomized, and quantum algorithms, submitted to the Proceedings of the 4th International Conference on Monte Carlo and Quasi-Monte Carlo Methods, Hong Kong 2000.

[13] S. Heinrich and E. Novak (2001b): On a problem in quantum summation (in preparation).

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

[15] M. A. Nielsen and I. L. Chuang (2000): Quantum Computation and Quantum Information, Cambridge University Press.

[16] E. Novak (1988): Deterministic and Stochastic Error Bounds in Numerical Analysis. Lecture Notes in Mathematics **1349**, Springer.

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

[18] A. O. Pittenger (1999): Introduction to Quantum Computing Algorithms. Birkhäuser, Boston.

[19] D. Pollard (1984): Convergence of Stochastic Processes. Springer-Verlag, New York.

[20] P. W. Shor (1994): Algorithms for quantum computation: Discrete logarithms and factoring. Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, pp. 124–134. See also http://arXiv.org/abs/quant-ph/9508027.

[21] P. W. Shor (1998): Quantum computing. Documenta Mathematica, Extra Volume ICM 1998, I, 467–486.

[22] P. W. Shor (2000): Introduction to Quantum Algorithms.
See http://arXiv.org/abs/quant-ph/quant-ph/0005003.

[23] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski (1988): Information-Based Complexity. Academic Press.