Quantum Complexity of Numerical Problems

S. Heinrich
Dept. Computer Science
University of Kaiserslautern
Germany
e-mail: heinrich@informatik.uni-kl.de
homepage:
http://www.uni-kl.de/AG-Heinrich

Abstract

A challenging question in the overlap of computer science, mathematics, and physics, is the exploration of potential capabilities of quantum computers. Milestones were the factoring algorithm of Shor (1994) and the search algorithm of Grover (1996). So far, major research was concentrated on discrete and algebraic problems. Much less was known about computational problems of analysis, including such a prominent example as high dimensional numerical integration, which is well-studied in the classical settings. We seek to understand how efficiently this and related problems can be solved in the quantum model of computation (that is, on a quantum computer) and how the outcome compares to the efficiency of deterministic or randomized algorithms on a classical (i. e. non-quantum) computer. In this paper we give a survey of the state of the art in this field, including also a brief introduction to the general ideas of quantum computing.

1 Introduction

A quantum computer is a computing device based on quantum mechanical laws of the (sub)atomic world. The idea of such a computer was developed by Feynman [8] in 1982. He emphasized that simulating quantum mechanics on a classical computer is extremly hard, probably infeasible. So why not try to simulate quantum mechanics using quantum devices themselves. (Thoughts in this direction were also expressed by Manin [21] in 1980, see also [22].) In 1985 Deutsch [6] presented a formal model of computation for quantum computing. A sensational breakthrough of quantum computing

was Shor's [29] result of 1994: He produced a polynomial (in the number of bits) algorithm for factoring large integers (no polynomial classical – deterministic or randomized – algorithm is known). Another seminal result is due to Grover [9] in 1996: Given a function $f : \{0, \ldots, N-1\} \to \{0, 1\}$ with the property that there is a unique i_0 with $f(i_0) = 1$, the task is to find this i_0. The function is given as a black box, which means that function values are only available at request, by query calls. Classical deterministic or randomized algorithms cannot solve this problem with less than $\Omega(N)$ queries to f. Grover's quantum algorithm needs $\mathcal{O}(\sqrt{N})$ quantum queries (we explain this notion later).

These developments triggered an explosion of efforts in quantum computing. The challenges of quantum computing to physicists are to find quantum systems suitable for computation, i.e., to built a quantum computer. In recent years, various realizations are tested in laboratories, but so far only a small number of system components (qubits) is possible.

What are the challenges of quantum computing to mathematicians and computer scientists? To find more problems for which quantum algorithms are better than all known classical algorithms (Shor's result belongs to this category). And even stronger: Find more problems for which quantum algorithms are **provably** better than all possible classical algorithms (like in Grover's result). In this vein all kinds of discrete problems are being investigated. Furthermore, Feynman's original idea was realized: various quantum algorithms for quantum mechanical simulations were suggested which are better than known classical algorithms (however, no proof of superiority over all possible classical algorithms was given for these types of numerical problems).

The study of numerical problems from the stronger point of view of provable superiority was begun by Brassard, Høyer, Mosca, and Tapp [5, 4]. They exhibited a quantum algorithm for computing the mean of $\{0, 1\}$-valued sequences, provably superior to all classical algorithms. That this algorithm is even optimal among all quantum algorithms was shown by Nayak and Wu [23]. First ideas of how to appy quantum algorithms for integration were expressed by Abrams and Williams [1]. Novak [26] carried out a first systematic study of integration, including lower bounds and provable superiority. He considered integration of functions from Hölder spaces. In the sequel this study was continued and widely extended. The author considered the mean of p-summable sequences and integration in L_p [12] and in Sobolev spaces [13]. A quantum complexity theory for continuous problems of numerical analysis, that is, the quantum setting of information-based complexity theory, was developed in [12]. Novak and the author

[17] completed the analysis of mean computation for p-summable sequences. Path integration was considered by Traub and Woźniakowski [32]. First approaches towards approximation of functions by quantum algorithms were made by Novak, Sloan, and Woźniakowski [27] for high dimensional Hilbert function classes. The first matching bounds for approximation are given by the author in [15], where the case of Sobolev embeddings is considered.

Complexity of numerical problems is studied in the general framework of information-based complexity theory. By now for many important problems of numerical analysis, including high dimensional integration in various function spaces, matching upper and lower complexity bounds 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.

After a short review of crucial notions from the classical deterministic and randomized setting of information-based complexity theory, we explain basic ideas of quantum computation and present the quantum setting. Then we discuss recent results on quantum algorithms, lower bounds, and complexity for various integration problems, like approximating the mean of p-summable sequences and the integral of functions from Hölder and Sobolev spaces. It turns out that in many cases there is a quadratic speed-up of quantum algorithms over classical randomized ones, and, with increasing dimension of the integration domain, a polynomial speed-up of arbitrarily large degree of quantum algorithms over classical deterministic ones. Finally, we also give an example of a function class in which quantum integration gives an exponential speedup over deterministic algorithms and discuss some recent new directions.

For further reading on quantum computation we recommend the surveys by Aharonov [2], Ekert, Hayden, and Inamori [7], Shor [30], and the monographs by Pittenger [28], Gruska [10], and Nielsen and Chuang [24].

For notions and results in information-based complexity theory see the monographs by Traub, Wasilkowski, and Woźniakowski [31] and Novak [25], and the survey by the author [11] of the randomized setting. For a first overview of quantum complexity theory for numerical problems we refer to Heinrich and Novak [16], while the connections to Monte Carlo algorithms are emphasized in a survey by the author [14].

2 Numerical Problems in the Classical Settings

Let D be a non-empty set, F a set of real-valued functions on D, G a normed space, and let

$$S : F \to G \tag{1}$$

be a mapping, which we refer to as the solution operator. $S(f) \in G$ represents the exact solution of our numerical problem in consideration at input $f \in F$.

As an example, consider the computation of the mean (or, equivalently – up to the weighting factor – the summation of finite sequences). Let $D = \{0, \ldots, N-1\}$ and $G = \mathbf{R}$. For a function $f : D \to \mathbf{R}$ define

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

To specify the set of inputs, consider the **discrete L_p-classes**: For $1 \leq p \leq \infty$, let $L_p^N = \mathbf{R}^N$, equipped with the norm

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

if $1 \leq p < \infty$, and

$$\|f\|_{L_\infty^N} = \max_{0 \leq i < N} |f(i)|.$$

Now we let

$$F = \mathcal{B}(L_p^N) = \{f \in L_p^N \,:\, \|f\|_{L_p^N} \leq 1\}$$

be the unit ball of L_p^N.

Our second example is multivariate integration. Here we let $d \in \mathbf{N}$, $D = [0,1]^d$, and $G = \mathbf{R}$. For a Lebesgue integrable function $f : [0,1]^d \to \mathbf{R}$ put

$$S(f) = I_d f := \int_{[0,1]^d} f(t) dt.$$

As input sets we consider two basic types of classes:

Hölder classes: For $r \in \mathbf{N}_0$ and $0 < s \leq 1$ define

$$\begin{aligned} F = \mathcal{B}(F_d^{r,s}) \;=\; & \{f \in C^r([0,1]^d),\ \|f\|_\infty \leq 1, \\ & |\partial^\alpha f(x) - \partial^\alpha f(y)| \leq |x-y|^s, |\alpha| = r\}. \end{aligned}$$

Here $C^r([0,1]^d)$ stands for the set of r times continuously differentiable functions.

Sobolev classes: For $r, d \in \mathbf{N}$, and $1 \leq p \leq \infty$, satisfying $r/d > 1/p$ (the Sobolev embedding condition) let

$$F = \mathcal{B}(W_{p,d}^r) = \{f \in L_p([0,1]^d) \,:\, \|\partial^\alpha f\|_{L_p} \leq 1,\ |\alpha| \leq r\},$$

where ∂^α denotes the weak partial derivative.

In the classical deterministic setting we will consider the following general form of an algorithm which uses n function values:

$$A_n(f) = \varphi(f(x_1), \ldots, f(x_n)).$$

Here $x_i \in D \quad (i = 1, \ldots, n)$ are any points, and $\varphi : \mathbf{R}^n \to G$ is an arbitrary mapping (both chosen by the algorithm designer). The error of A_n over F is defined as

$$e(S, A_n, F) = \sup_{f \in F} \|S(f) - A_n(f)\|_G \,.$$

The crucial quantity of complexity analysis in this setting is the deterministic n-th minimal error

$$e_n^{\mathrm{det}}(S, F) = \inf_{A_n} e(S, A_n, F) \,,$$

which is the minimal possible error reachable among all algorithms which use at most n function values.

In the classical randomized setting the general form of an algorithm which uses n function values is the following: $A_n = (A_n^\omega)_{\omega \in \Omega}$, where

$$A_n^\omega(f) = \varphi^\omega(f(x_1^\omega), \ldots, f(x_n^\omega)) \,,$$

$(\Omega, \Sigma, \mathbf{P})$ is a probability space, $x_i^\omega \in D$ are random points in D, and $\varphi^\omega : \mathbf{R}^n \to G \quad (\omega \in \Omega)$ is a random mapping. Then the error of A_n is defined as

$$e(S, A_n, F) = \sup_{f \in F} (\mathbf{E}\|S(f) - A_n^\omega(f)\|_G^2)^{1/2} \,.$$

This leads to the randomized n-th minimal error

$$e_n^{\mathrm{ran}}(S, F) = \inf_{A_n} e(S, A_n, F) \,,$$

that is, the minimal possible (mean-square) error among all randomized algorithms using at most n function values. So far a short overview of the classical settings. For further reading we refer to [31, 25, 11]. In section 4 we introduce the quantum setting.

3 Quantum Computation

The mathematical framework for the basic unit of quantum computing is the two dimensional complex Hilbert space $H_1 := \mathbf{C}^2$. The unit sphere of H_1 serves as the state space of quantum systems with two classical states. The classical states correspond to the elements of the unit vector basis $e_0, e_1 \in H_1$. Such systems are called **qubits** – quantum bits. Following quantum mechanics notation, we write $|0\rangle$ instead of e_0 and $|1\rangle$ instead of e_1.

A quantum computer is an m-**qubit system**, that is, a system of m interacting qubits (which can be manipulated, as will be explained below). Such a system is represented by the tensor product

$$H_m := \underbrace{H_1 \otimes H_1 \otimes \cdots \otimes H_1}_{m}.$$

This is the 2^m-dimensional complex Hilbert space, with its canonical basis

$$e_{i_0} \otimes e_{i_1} \otimes \cdots \otimes e_{i_{m-1}} \quad (i_0, i_1, \ldots, i_{m-1}) \in \{0,1\}^m.$$

Let us introduce further notational conventions:

$$\begin{aligned} e_{i_0} \otimes e_{i_1} \otimes \cdots \otimes e_{i_{m-1}} &=: |i_0\rangle\, |i_1\rangle \ldots |i_{m-1}\rangle \\ &=: |i\rangle \end{aligned}$$

where $i := (i_0 i_1 \ldots i_{m-1})_2 := \sum_{k=0}^{m-1} i_k 2^{m-1-k}$.

The vectors $|i\rangle = |i_0\rangle\, |i_1\rangle \ldots |i_{m-1}\rangle$ represent the classical states. A general state of the quantum system is given by the superposition

$$|\xi\rangle = \sum_{i=0}^{2^m-1} \alpha_i\, |i\rangle \qquad \left(\sum_{i=0}^{2^m-1} |\alpha_i|^2 = 1 \right).$$

(Let us make the following notational conventions connected with the (Dirac) notation of quantum mechanics: if $|\ldots\rangle$ contains a nonnegative integer inside, or a typical symbol denoting such a number, like $i, j, k, \ldots$, we mean the canonical basis vector corresponding to this number, if $|\ldots\rangle$ contains any other symbol, like $|\xi\rangle, |\varphi\rangle, |\psi\rangle$, we mean any vector of H_m).

How to use such m-qubit systems for computing? To explain this at a simple example, let us first go back to classical computations, and consider the addition of two binary numbers

$$(i_0 i_1 \ldots i_{m-1})_2 + (j_0 j_1 \ldots j_{m-1})_2 = (k_0 k_1 \ldots k_m)_2.$$

A classical implementation would look as follows:

$$\begin{array}{c} i_0, \ldots, i_{m-1}, j_0, \ldots, j_{m-1},\ 0, \ldots,\ 0 \\ \downarrow \\ i_0, \ldots, i_{m-1}, j_0, \ldots, j_{m-1}, k_0, \ldots, k_m \end{array}$$

where the computation of the bits of the sum $k_0, \ldots, k_m$ is realized using circuits of classical gates {`and, or, not, xor`} in the usual way: add the last bit, then the second last plus the carry bit etc.

How to operate m-qubit quantum systems? Which operations are allowed? Schrödinger's equation implies: all evolutions of a quantum system must be represented by unitary transforms of H_m.

Quantum computing assumes that we are able to perform a number of elementary (quantum) gates on the system

Next we present some common quantum gates. The simplest ones are the **one qubit gates**. They manipulate only one component of the tensor product $H_1 \otimes H_1 \otimes \cdots \otimes H_1$. Formally, a one qubit gate is given by a unitary operators on H_1. The action on the whole tensor product is then obtained by taking the tensor product of this operator with the identities on the other components. An important one qubit gate is the **Hadamard gate** defined by

$$\begin{aligned} |0\rangle &\to \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ |1\rangle &\to \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned}$$

(the values on the basis vectors define the unitary transform uniquely). Another example is the family of **phase shifts** given for any fixed $\theta \in [0, 2\pi]$ by

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle \to \alpha_0 |0\rangle + e^{i\theta} \alpha_1 |1\rangle .$$

In a similar way, **two qubit gates** are defined. They manipulate two components of $H_1 \otimes H_1 \otimes \cdots \otimes H_1$. We consider only one example, the **quantum xor gate** (also called **controlled-not gate**): Its unitary action from $H_1 \otimes H_1$ to $H_1 \otimes H_1$ is determined by

$$\begin{aligned} |0\rangle\,|0\rangle &\to |0\rangle\,|0\rangle \\ |0\rangle\,|1\rangle &\to |0\rangle\,|1\rangle \\ |1\rangle\,|0\rangle &\to |1\rangle\,|1\rangle \\ |1\rangle\,|1\rangle &\to |1\rangle\,|0\rangle \end{aligned}$$

That is, if the first bit is zero, nothing happens to the second, and if the first is one, the second is negated (controlled not).

A basic result of quantum computing states that these gates are already enough to approximate any unitary operator on $H_1 \otimes H_1 \otimes \cdots \otimes H_1$ (see, e.g. [24]):

The Hadamard gate, the phase shift $\theta = \pi/4$ and the xor gate form an approximately universal system of gates – each unitary transform of H_m can be approximated in the operator norm to each precision by a finite composition of these gates (up to a complex factor).

Hence, if one is able to realize these basic gates in physical systems, one can do quantum computing. Physicists are working on implementations of these gates in various quantum systems such as photons, trapped ions, magnetic resonance systems etc.

Of course, the statement above does not yet say anything about the efficiency of such an approximation, that is, how many of these elementary gates are needed. This is the theme of quantum complexity theory.

Let us come back to our examples and emphasize two important aspects:
1. These gates can transform classical states into superpositions. Example: The Hadamard gate applied to the first and then to the second qubit

$$|0\rangle\,|0\rangle \longrightarrow \frac{1}{2}\left(|0\rangle\,|0\rangle + |0\rangle\,|1\rangle + |1\rangle\,|0\rangle + |1\rangle\,|1\rangle\right).$$

2. They act also on superpositions. Examples:
The quantum xor:

$$\begin{array}{c} \alpha_0\,|0\rangle\,|0\rangle + \alpha_1\,|0\rangle\,|1\rangle + \alpha_2\,|1\rangle\,|0\rangle + \alpha_3\,|1\rangle\,|1\rangle \\ \downarrow \\ \alpha_0\,|0\rangle\,|0\rangle + \alpha_1\,|0\rangle\,|1\rangle + \alpha_2\,|1\rangle\,|1\rangle + \alpha_3\,|1\rangle\,|0\rangle \end{array}$$

Quantum addition of binary numbers:

$$\begin{array}{c} \sum \alpha_{ij}\,|i_0\rangle \ldots |i_{m-1}\rangle\,|j_0\rangle \ldots |j_{m-1}\rangle\,|0\rangle\ \ldots\ |0\rangle \\ \downarrow \\ \sum \alpha_{ij}\,|i_0\rangle \ldots |i_{m-1}\rangle\,|j_0\rangle \ldots |j_{m-1}\rangle\,|k_0\rangle \ldots |k_m\rangle \end{array}$$

That is, in the quantum world, we add all possible binary m-digit numbers in parallel.

So is a quantum computer an ideal parallel computer, with exponentially many processors? Not exactly, it is not that easy! The point is that we cannot access all components of the superposition. We have to measure the quantum system, which destroys the superposition.

Quantum computing assumes that we are able to access the results of the quantum computation process via measurement (with respect to the canonical basis).

This means the following: Measuring a system in a (superposition) state

$$|\psi\rangle = \sum_{i=0}^{2^m-1} \alpha_i |i\rangle \qquad \left(\sum_{i=0}^{2^m-1} |\alpha_i|^2 = 1\right)$$

results in one of the classical states:

$$|i\rangle \quad \text{with probability} \quad |\alpha_i|^2 \quad (i = 0, \ldots, 2^m - 1).$$

So, returning to our example of binary addition, after measurement we would get just

$$|i_0\rangle \ldots |i_{m-1}\rangle\, |j_0\rangle \ldots |j_{m-1}\rangle\, |k_0\rangle \ldots |k_m\rangle$$

with probability $|\alpha_{ij}|^2$. This simple example shows two typical features: A quantum computer is indeed a powerful device due to the exponential parallelizm of computation. On the other hand, we cannot "look into" the device to read off all parallel results. To exploit a quantum computer properly, one has to go deeper: Some elaborate algorithmic techniques are needed to transform the quantum state in such a way that the desired final result can be obtained via measurement (with high probability). The quantum Fourier transform, phase estimation, and Grover's iteration are the foremost tools to reach this goal (see the references mentioned above for details). This concludes our short introduction into quantum computing.

4 The Quantum Setting for Numerical Problems

Now we become more specific and describe the formal quantum model of computation (the general way a quantum algorithm should look like), suited to handle numerical problems. An important issue not discussed so far is this: How does the quantum algorithm get information about $f \in F$? (Remember, the quantum algorithm is supposed to approximate $S(f)$ for each $f \in F$.) Let us first look at the binary case, that is, we are dealing with Boolean functions $f : \{0, 1, \ldots, 2^{m_1} - 1\} \to \{0, 1\}$.

The classical (i.e., non-quantum) black box (also called query, or subroutine) maps $(i, 0, k)$ to $(i, f(i), k)$, that is, at request i the subroutine writes $f(i)$ into some memory space originally filled by 0. The entry $k \in \{0, 1, \ldots, 2^{m-m_1-1}-1\}$ just represents the contents of additional "work bits", which are needed for the rest of the computation and which are not touched by the query. This is the way we always describe numerical algorithms – e.g. in the case of integration, it is tacitly assumed that we have at our disposal the values of the funtion to integrate. The reflection above leads us to set

$$|i\rangle\,|0\rangle\,|k\rangle \to |i\rangle\,|f(i)\rangle\,|k\rangle\,.$$

in the quantum case. This is, however, not yet complete – to define a unitary operator $Q_f : H_m \to H_m$ uniquely, we have to extend the mapping above to a bijection of classical states. A standard (and convenient, from the point view of implementation by elementary quantum gates) way of doing this is the following, which gives us the **quantum (binary) query:**

$$Q_f : |i\rangle\,|j\rangle\,|k\rangle \to |i\rangle\,|j \oplus f(i)\rangle\,|k\rangle\,,$$

where $\oplus$ denotes addition modulo 2. This type of query is used in Grover's algorithm (to get information about the function f, see the introduction). The binary quantum query model was also studied intensively from the complexity point of view, see Beals, Buhrmann, Cleve, Mosca, and de Wolf [3].

Now we consider the general case, that is, of functions f on general domains D with values in $\mathbf{R}$, as we encounter them in numerical analysis.

A **quantum query** is given by $(m, m_1, m_2, \tau, \beta)$, where $m, m_1, m_2 \in \mathbf{N}$, $m_1 + m_2 \leq m$,

$$\tau : \{0, \ldots, 2^{m_1} - 1\} \to D$$

and

$$\beta : \mathbf{R} \to \{0, \ldots, 2^{m_2} - 1\}$$

are any mappings. The quantum query $Q_f : H_m \to H_m$ is defined as follows, where it is convenient to consider H_m as being represented as $H_m = H_{m_1} \otimes H_{m_2} \otimes H_{m-m_1-m_2}$.

$$Q_f : |i\rangle\,|j\rangle\,|k\rangle \to |i\rangle\,|j \oplus \beta(f(\tau(i)))\rangle\,|k\rangle\,,$$

with $\oplus$ standing for addition modulo 2^{m_2}. The role of τ is to map indices i to nodes $\tau(i) \in D$, while β encodes the real number $f(\tau(i))$ as a binary integer $\beta(f(\tau(i)))$. This type of query was introduced in [12].

For β one could take, for example,

$$\beta(x) = \begin{cases} 0 & \text{if} \quad x < a \\ \left\lfloor 2^{m_2} \frac{x-a}{b-a} \right\rfloor & \text{if} \quad a \leq x < b \\ 2^{m_2} - 1 & \text{if} \quad x \geq b \end{cases}$$

if we have to deal with functions taking values in $[a, b]$.

A **quantum algorithm** A_n (for the approximate solution of a problem of the type (1)) is given by

$$(m, m_1, m_2, \tau, \beta, U_0, \ldots, U_n, i_0, \varphi)$$

where $(m, m_1, m_2, \tau, \beta)$ is a tupel defining a quantum query Q_f as explained above, $U_0, \ldots, U_n$ are unitary operators on H_m, $0 \leq i_0 < 2^m$ is any number (describing the starting state), and $\varphi : \{0, \ldots, 2^m - 1\} \to G$ is any mapping (which produces the final output of the algorithm from the measurement in a classical computation). The computation acts as follows.

Quantum model of computation:

starting state:

$|i_0\rangle \in H_m$ (a classical state)

computation:

$$|i_0\rangle \to U_0 |i_0\rangle \to Q_f U_0 |i_0\rangle \to U_1 Q_f U_0 |i_0\rangle \to \ldots$$

$$\to U_n Q_f U_{n-1} \ldots Q_f U_1 Q_f U_0 |i_0\rangle =: |\xi\rangle$$

measurement:

$|\xi\rangle = \sum_{i=0}^{2^m-1} \alpha_i |i\rangle \to |i\rangle$ with probability $|\alpha_i|^2$

output:

$|i\rangle \to \varphi(i) =: A_n(f) \in G$

We refer to A_n as a quantum algorithm with n queries, the role of the queries being the same as the role of the function values in the classical

settings explained in section 2. So the Q_f (and only these) provide information about the input function f, while the unitaries $U_0, \ldots, U_n$ stand for the compositions of the quantum gates applied between queries to process the obtained information. Observe that $A_n(f)$ is a random variable, with values in the normed space G (which is always $\mathbf{R}$ for our purposes of studying integration). Therefore, the error of A_n at input $f \in F$ is defined in the probabilistic way:

$$e(S, A_n, f) = \inf \{\varepsilon : \mathbf{P}\{\|S(f) - A_n(f)\|_G \leq \varepsilon\} \geq 3/4\}.$$

Note that we specified the error probability to be not greater than $1/4$. This special choice is not essential, the number can be replaced by any other number strictly between 0 an $1/2$. The reason is that by repeating an algorithm k times and computing the median of the results, the error probability can be reduced to 2^{-ck} for some $c > 0$ not depending on k.

The error of A_n over the whole class F is

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

The crucial quantity for complexity analysis is the quantum n-th minimal error

$$e_n^{\mathrm{q}}(S, F) = \inf_{A_n} e(S, A_n, F)$$

that is, the minimal error reachable among all possible quantum algorithms that use not more than n quantum queries. This quantity neglects all combinatory cost (number of gates, classical operations) as it is customary in query-based complexity analysis. However, in deriving matching upper and lower bounds, the proof of the upper ones usually contains a concrete algorithm whose total cost can be counted and, for all the situations studied here, except for Theorem 6, turn out to be of the same order (at least up to logarithms) as the number of queries. We therefore just state the asymptotic estimate of $e_n^{\mathrm{q}}(S, F)$, keeping in mind these comments.

5 Mean Computation and Integration

The following result is due to Brassard, Høyer, Mosca, and Tapp [5, 4] (upper bound) and Nayak and Wu [23] (lower bound).

Theorem 1. *There is a constant $0 < c < 1$ such that for all n, N, $n < cN$,*

$$e_n^{\mathrm{q}}(S_N, \mathcal{B}(L_\infty^N)) \asymp n^{-1}.$$

The estimate of Theorem 1 says two things: First, there is a quantum algorithm which requires not more than n quantum queries and computes the mean for all sequences in $\mathcal{B}(L_\infty^N)$ with error not larger than $c_1 n^{-1}$, $c_1 > 0$ a constant not depending on n and N. (Note that we often use the same symbol for possibly different constants.) Second, it says that no algorithm that uses not more than n quantum queries (and maybe even an unlimited number of gates!) can have error less than $c_2 n^{-1}$, with $c_2 > 0$ another independent of n and N constant.

Let us say some words about the methods behind Theorem 1. The counting algorithm of Brassard, Høyer, Mosca, and Tapp [4], originally designed for computing the mean of $\{0,1\}$-valued sequences, is easily adapted to $[-1,1]$-valued sequences. This algorithm provides the upper estimate and is based on two fundamental techniques of quantum computing – the technique of Shor of using the quantum version of the discrete Fourier transform for estimating eigenvalues of certain unitary operators, and the Grover iterate, a crucial ingredient of Grover's algorithm. The counting algorithm can be implemented using $\mathcal{O}(n \log^2 n)$ elementary quantum gates (compare the comment at the end of the previous section).

The lower bounds come from the the polynomial method [3], which states that the succcess probability of a quantum algorithm is a certain polynomial of degree at most the number of queries. Starting from that, Nayak and Wu [23] use classical facts from approximation theory: the Bernstein and Markov inequality for polynomials.

Now we want to compare Theorem 1 to the known result for the classical deterministic setting. It is clear that the algorithm of direct computation of the mean has error 0. However, it needs all the N function values. What we are particularly interested in (also in view of its applications in high dimensional integration) is a small n and a huge N. Here deterministic algorithms fail completely. We have, for each constant $0 < c < 1$

$$e_n^{\mathrm{det}}(S_N, \mathcal{B}(L_\infty^N)) \asymp 1 \quad (n < cN),$$

that is, no deterministic algorithm using essentially less than N queries can have an error essentially better than the trivial one. In the classical randomized setting we have

$$e_n^{\mathrm{ran}}(S_N, \mathcal{B}(L_\infty^N)) \asymp n^{-1/2}.$$

We see that randomized classical algorithms reach non-trivial error for n essentially less than N. However, we also see the speedup of quantum algorithms. It is a quadratic one, like in Grover's search algorithm.

Now we present the first result about integration.

Theorem 2. (Novak [26]) *Let* $r \in \mathbf{N}_0$, $d \in \mathbf{N}$ *and* $0 < s \leq 1$. *Then*

$$e_n^{\mathrm{q}}(I_d, \mathcal{B}(F_d^{r,s})) \asymp n^{-\frac{r+s}{d}-1}.$$

Novak uses Theorem 1 and tools from information-based complexity theory. Moreover, a major ingredient in the proof is a technique from the field of Monte Carlo algorithms – a quantum analog of separation of the main part.

It is instructive to compare this to the classical settings: In the classical deterministic case,

$$e_n^{\mathrm{det}}(I_d, \mathcal{B}(F_d^{r,s})) \asymp n^{-\frac{r+s}{d}}.$$

Look at this statement for d huge, or, in other words, $(r+s)/d$ small. Then the best convergence rate of deterministic classical algorithms is practically negligible, while the quantum rate is smaller than n^{-1}. Let us consider this from the point of view of the number of queries needed to reach error $\varepsilon > 0$. It is readily calculated from the above, that in the classical deterministic setting, we need at least $\Omega((1/\varepsilon)^{d/(r+s)})$ queries. Consequently, the exponent of the cost is proportional to the dimension d. The respective number of quantum queries is $\Omega((1/\varepsilon)^{d/(r+s+d)})$, so this exponent is always smaller than 1. In this sense we can say that quantum algorithms can provide a polynomial speedup of arbitrarily large degree over classical deterministic algorithms. In the classical randomized setting we have

$$e_n^{\mathrm{ran}}(I_d, \mathcal{B}(F_d^{r,s})) \asymp n^{-\frac{r+s}{d}-1/2}.$$

Looking also at this from the point of view of high d, we see that quantum algorithms reach again essentially a quadratic speedup over classical randomized algorithms. (As already mentioned, the statements for the classical settings are known results from information-based complexity theory, and we refer to the sources [31, 25, 11].)

After these results have been obtained, another interesting question arose: The above are results for the L_∞-norm. What about weaker norms like the L_2-norm, that is, what happens for $f \in \mathcal{B}(L_2^N)$ and $f \in \mathcal{B}(W_2^r)$, or more generally $f \in \mathcal{B}(L_p^N)$, $f \in \mathcal{B}(W_p^r)$? We know that the L_2-case is the most important one for Monte Carlo algorithms, and that they preserve the rate of the L_∞-case even for the larger L_2-classes. Could it be that for $2 \leq p < \infty$, quantum algorithms loose gradually, and eventually, for $p = 2$, Monte Carlo algorithms turn out to be as good as quantum algorithms? This problem was solved in [12]. It turns out that there are quantum algorithms which also preserve the favorable rate of the L_∞-case (up to logarithmic factors, at least), and so, the speedup remains also for the L_2-case.

Theorem 3. (Heinrich [12]) *Let $1 \le p < \infty$. There is a constant $c > 0$ such that for all n, N, with $n < cN$,*

$$\begin{array}{lll} e_n^{\rm q}(S_N, \mathcal{B}(L_p^N)) & \asymp & n^{-1} \qquad \text{if} \quad 2 < p < \infty \\ e_n^{\rm q}(S_N, \mathcal{B}(L_p^N)) & \asymp_{\log} & n^{-2+2/p} \qquad \text{if} \quad 1 \le p \le 2 \text{ and } n < \sqrt{N}. \end{array}$$

We use the notation $\asymp_{\log}$ to indicate that the bounds are sharp up to logarithmic factors, that is, the upper and lower bound may contain differing terms of the form $c \log^\alpha n \log^\beta N$ where c, α, β may depend on the problem parameters p (and d and r later on) but do not depend on n and N.

For comparison, in the classical deterministic setting, we have

$$e_n^{\rm det}(S_N, \mathcal{B}(L_p^N)) \asymp 1,$$

and in the classical randomized setting

$$e_n^{\rm ran}(S_N, \mathcal{B}(L_p^N)) \asymp \begin{cases} n^{-1/2} & \text{if} \quad 2 \le p < \infty \\ n^{-1+1/p} & \text{if} \quad 1 \le p < 2. \end{cases}$$

The new quantum algorithms are based on a suitable multilevel splitting of sequences from L_p^N, distributing queries over levels, and combining the decay of the integral over the levels and precise error estimates for counting. The proof of the lower bound combines techniques of information-based complexity theory with the approach of Nayak and Wu [23].

Theorem 3 shows that so far the case of p-summable sequences for $1 \le p < 2$ and $\sqrt{N} \le n < N$ was left open. In fact, the upper bound holds true also for that range. So is it sharp? (Since we are interested in huge N and moderate n, this problem was at first glance a rather academic one.) It was a certain surprise that in this case a further improvement of the quantum algorithms of Theorem 3 is possible, and moreover, the result turned out to be a crucial ingredient to determine the sharp order in the Sobolev case discussed below.

Theorem 4. (Heinrich and Novak [17]) *Let $1 \le p < 2$. There is a constant $c > 0$ such that for all n, N with $\sqrt{N} \le n < cN$,*

$$e_n^{\rm q}(S_N, \mathcal{B}(L_p^N)) \asymp_{\log} n^{-2/p} N^{2/p-1}.$$

The optimal algorithm contains a new element. It uses Grover's search algorithm to handle a certain portion of the sequence, while the rest of it is taken care by the multilevel type algorithm developed for the proof of Theorem 3.

From Theorems 3 and 4, combined with a new discretization techniques, one can derive optimal quantum integration algorithms for functions from Sobolev spaces. This discretization technique, which is close in spirit to Maiorov's technique from approximation theory [20], allows to reduce the integration problem to a scale of discrete problems – of mean computation in $L_p^{N_l}$ $(l = 1, \ldots, k)$ for suitable k and N_l. This technique is useful also in the classical randomized setting, since it can be viewed as a multilevel variance reduction technique for Monte Carlo integration. Applications to the study of Monte Carlo algorithms which use few random bits will be given in [18].

Theorem 5. (Heinrich [13]) *Let* $1 \leq p < \infty$, $r, d \in \mathbf{N}$, $r/d > 1/p$. *Then*

$$e_n^{\mathrm{q}}(I_d, \mathcal{B}(W_{p,d}^r)) \asymp_{\log} n^{-r/d-1}.$$

For comparison, in the classical deterministic setting we have

$$e_n^{\mathrm{det}}(I_d, \mathcal{B}(W_{p,d}^r)) \asymp n^{-r/d}$$

and in the classical randomized setting

$$\begin{array}{ll} e_n^{\mathrm{ran}}(I_d, \mathcal{B}(W_{p,d}^r)) \asymp n^{-r/d-1/2} & \text{if} \quad 2 \leq p < \infty \\ e_n^{\mathrm{ran}}(I_d, \mathcal{B}(W_{p,d}^r)) \asymp n^{-r/d-1+1/p} & \text{if} \quad 1 \leq p < 2. \end{array}$$

Note the following interesting situation in the case $p = 1$. Quantum algorithms are by a factor of about n^{-1} better than classical randomized ones, while the latter yield no improvement over classical deterministic ones.

6 Summary and Further Comments

For a convenient overview we summarize the presented results in a table, including the known results about the classical settings. All rates are sharp (up to possible logarithmic factors, which we suppress, again). We present the $p = 1$ cases separately, because of the interesting relations between the settings.

	$e_n^{\det}$	e_n^{ran}	e_n^{q}
$\mathcal{B}(L_p^N)$, $2 \le p \le \infty$ $n \le cN$	1	$n^{-1/2}$	n^{-1}
$\mathcal{B}(L_p^N)$, $1 \le p < 2$ $n < n^{-1+1/p}\sqrt{N}$,	1		$n^{-2+2/p}$
$\mathcal{B}(L_p^N)$, $1 < p < 2$ $\sqrt{N} \le n \le cN$	1	$n^{-1+1/p}$	$n^{-2/p}N^{2/p-1}$
$\mathcal{B}(L_1^N)$, $n < \sqrt{N}$	1	1	1
$\mathcal{B}(L_1^N)$, $\sqrt{N} \le n \le cN$	1	1	$n^{-2}N$
$\mathcal{B}(F_d^{r,s})$	$n^{-(r+s)/d}$	$n^{-(r+s)/d-1/2}$	$n^{-(r+s)/d-1}$
$\mathcal{B}(W_{p,d}^r)$, $2 \le p \le \infty$	$n^{-r/d}$	$n^{-r/d-1/2}$	$n^{-r/d-1}$
$\mathcal{B}(W_{p,d}^r)$, $1 < p < 2$	$n^{-r/d}$	$n^{-r/d-1+1/p}$	$n^{-r/d-1}$
$\mathcal{B}(W_{1,d}^r)$	$n^{-r/d}$	$n^{-r/d}$	$n^{-r/d-1}$

As argued before, in the case of integration of Hölder functions, the speedup of quantum over deterministic algorithms can be polynomial of arbitrarily high degree. (Similar conclusions hold for the Sobolev case, as long as p is large enough.) The next result shows that there are function classes with even an exponential speedup.

Theorem 6. (Heinrich [12]) *Let $\mathcal{E}$ be the set of functions f on $[0,1]$ such that $\|f\|_{L_\infty} \le 1$ and for all $k \in \mathbf{N}$ and $s,t \in [0,1]$, $|s-t| \le 2^{-k}$ implies $|f(s)-f(t)| \le k^{-1}$. Then any classical deterministic integration algorithm of error $0 < \varepsilon < 1/32$ has cost at least $\Omega\left(2^{1/(32\varepsilon)}\right)$, while there is a quantum algorithm with error ε using $\mathcal{O}(1/\varepsilon)$ queries and $\mathcal{O}((1/\varepsilon)^2)$ qubits and gates.*

Another example with a similar speedup is path integration, which was considered by Traub and Woźniakowski [32]. Many interesting problems are related to this topic, in particular the study of broader function classes.

Note that in all problems considered so far the output was a single number. This raises an interesting question to be explored: What gain can quantum algorithms bring if the solution is not a number, but a family of numbers, a function. The extreme case is the approximation problem – here S is the identity embedding between some function spaces.

A first approach to approximation was made by Novak, Sloan, and Woźniakowski [27]. They study approximation in huge-dimensional reproducing kernel Hilbert spaces and clarify the conditions of tractability (polynomial dependence on the dimension) in the quantum setting. An interesting problem which is left open is to find matching upper and lower bounds.

Tight bounds for approximation of Sobolev embeddings in the quantum setting were obtained by the author [15].

An interesting numerical problem between integration and approximation is the computation of integrals depending on a (possibly multidimensional) parameter. The classical randomized setting was studied in [19]. The quantum setting was recently considered by Wiegand [33].

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] D. Aharonov (1998): Quantum computation – a review. In: Annual Review of Computational Physics, World Scientific, volume VI, ed. Dietrich Stauffer, see also http://arXiv.org/abs/quant-ph/9812037.

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

[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] D. Deutsch (1985): Quantum theory, the Church-Turing principle and the universal quantum computer. Proc. R. Soc. Lond., Ser. A 400, 97-117.

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

[8] R. Feynman (1982): Simulating physics with computers. Int. J. Theor. Phys. **21**, 467–488.

[9] 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.

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

[11] 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.

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

[13] S. Heinrich (2003): Quantum integration in Sobolev classes. J. Complexity (to appear). See also http://arXiv.org/abs/quant-ph/0112153.

[14] S. Heinrich (2003): From Monte Carlo to Quantum Computation. Proceedings of the 3rd IMACS Seminar on Monte Carlo Methods MCM2001, Salzburg, to appear in Mathematics and Computers in Simulation. See also http://arXiv.org/abs/quant-ph/0112152.

[15] S. Heinrich (2003): Quantum Approximation of Sobolev Embeddings. Paper in preparation.

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

[17] S. Heinrich and E. Novak (2003): On a problem in quantum summation. J. Complexity (to appear), see also http://arXiv.org/abs/quant-ph/0109038.

[18] S. Heinrich, E. Novak, and H. Pfeiffer (2003): Paper in preparation.

[19] S. Heinrich and E. Sindambiwe (1999): Monte Carlo complexity of parametric integration. J. Complexity 15, 317-341

[20] V. E. Maiorov (1975) Discretization of the problem of diameters (in Russian), Usp. Mat. Nauk 30, No. 6 (186), 179-180.

[21] Yu. I. Manin (1980): Computable and uncomputable (in Russian). Sovetskoye Radio, Moscow.

[22] Yu. I. Manin (1999): Classical computing, quantum computing, and Shor's factoring algorithm. See http://arXiv.org/abs/quant-ph/9903008.

[23] 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.

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

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

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

[27] E. Novak, I. H. Sloan, and H. Wozniakowski: Tractability of Approximation for Weighted Korobov Spaces on Classical and Quantum Computers, see http://arXiv.org/abs/quant-ph/0206023.

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

[29] 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.

[30] P. W. Shor (2000): Introduction to quantum algorithms. See http://arXiv.org/abs/quant-ph/0005003.

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

[32] J. F. Traub and H. Woźniakowski (2001): Path integration on a quantum computer. See http://arXiv.org/abs/quant-ph/0109113.

[33] C. Wiegand (2003): Paper in preparation.