Numerical Analysis on a Quantum Computer

Stefan Heinrich

Fachbereich Informatik
Universität Kaiserslautern
D-67653 Kaiserslautern, Germany
heinrich@informatik.uni-kl.de
http://www.uni-kl.de/AG-Heinrich

Abstract. We give a short introduction to quantum computing and its relation to numerical analysis. We survey recent research on quantum algorithms and quantum complexity theory for two basic numerical problems – high dimensional integration and approximation. Having matching upper and lower complexity bounds for the quantum setting, we are in a position to compare them with those for the classical deterministic and randomized setting, previously obtained in information-based complexity theory. This enables us to assess the possible speedups quantum computation could provide over classical deterministic or randomized algorithms for these numerical problems.

1 Introduction

A quantum computer is a computing device based on quantum mechanical laws of the (sub)atomic world. The first idea of such a computer is due to Manin [21] in 1980 (see also [22]), and Feynman [7] in 1982. An abstract, theoretical model of quantum computation was developed in 1985 by Deutsch [5]. The breakthrough of quantum computing occurred in 1994, when Shor [31] proved that a quantum computer could factor large integers N in $\mathcal{O}((\log N)^3)$ operations, while no polynomial in $\log N$ classical (deterministic or randomized) algorithm is known. (By "classical" we always mean "non-quantum".) Another important result was obtained by Grover [8] in 1996, who dealt with the following problem: Let $f : \{0, \dots, N-1\} \to \{0, 1\}$ with the property that there is a unique i_0 with $f(i_0) = 1$. Find this i_0. It is not difficult to show that classically (deterministically or randomized) one needs $\mathcal{O}(N)$ operations. In the quantum setting Grover showed that $\mathcal{O}(\sqrt{N})$ suffice.

This created a challenge to physicists: Find quantum systems suitable for computation, i.e., build a quantum computer. In recent years, various realizations are tested in laboratories. So far only systems with a small number of components (qubits) are possible.

The challenge of quantum computing to mathematicians and computer scientists, on the other hand, is: Find more problems for which quantum algorithms are (provably) better than all classical algorithms. In the sequel, all kinds of discrete problems were investigated. Much less was done for problems of analysis. A

natural question is the following: What could a quantum computer bring for the solution of numerical problems? Research in this direction was started in 1998 by Boyer, Brassard, Høyer, Mosca, and Tapp [3], [4], who developed a quantum algorithm for computing the mean of a sequence of numbers. Nayak and Wu [23] showed matching lower bounds. Novak [26] was the first to give a quantum complexity analysis for the integration of functions from Hölder spaces. Computing the mean of p-summable sequences, integration in Sobolev spaces, and the rigorous quantum setting of information-based complexity theory are due to Heinrich [11], [12].

Numerous further recent contributions include Traub, Woźniakowski [33] (path integration), Novak, Sloan, Woźniakowski [27] (integration and approximation in reproducing kernel Hilbert spaces), Heinrich [13], [14] (approximation of Sobolev embeddings), Wiegand [35] (parametric integration), Kacewicz [17] [18] (intitial value problems for systems of ordinary differential equations), Papageorgiou, Woźniakowski [28] (eigenvalue computation for the Sturm-Liouville-problem), Kwas [19] (Feynman-Kac path integrals), and Heinrich [15] (elliptic PDE).

Combining the results above with known results from information-based complexity theory about the classical deterministic and randomized setting [25], [34], [10], one can prove the superiority of quantum algorithms for many of these problems.

In this paper I want to give an introduction to the ideas of quantum computing and survey a few typical recent results concerning basic numerical problems: high dimensional integration and approximation. This will include a comparison of the potential of quantum algorithms with that of deterministic and randomized classical ones.

For further reading on quantum computation we refer to the surveys by Aharonov [1], Ekert, Hayden, and Inamori [6], Shor [32], and the monographs by Pittenger [30], Gruska [9], and Nielsen and Chuang [24]. Basic notions and results in information-based complexity theory can be found in the monographs by Traub, Wasilkowski, and Woźniakowski [34] and Novak [25], and the survey [10] of the randomized setting.

2 Quantum Computing

First we describe the mathematical framework of quantum computing. Let $H_1 := \mathbf{C}^2$ be the two-dimensional complex Hilbert space (the unit sphere of H_1 represents the state space of a **qubit** – quantum bit). Let $\{e_0, e_1\}$ be the unit vector basis. In accordance with quantum mechanical notation we write $|0\rangle$ instead of e_0 and $|1\rangle$ instead of e_1.

The basic quantum computing device is given by an m-qubit-system. Mathematically, it is represented by the tensor product

$$H_m := \underbrace{H_1 \otimes H_1 \otimes \ldots \otimes H_1}_{m},$$

with the basis

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

Thus, H_m is the 2^m-dimensional complex Hilbert space, its unit sphere is the state space of the m-qubit system, and a quantum computation is a trajectory through this state space according to specific rules, which we describe below.

We make the following further notational conventions:

$$e_{i_0} \otimes e_{i_1} \otimes \ldots \otimes e_{i_{m-1}} =: |i_0\rangle\, |i_1\rangle \ldots |i_{m-1}\rangle =: |i\rangle$$

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

The basis states $|i\rangle = |i_0\rangle\, |i_1\rangle \ldots |i_{m-1}\rangle$ represent the classical states of the system, the general quantum states of the m-qubit system are given by superpositions

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

How to use m-qubit quantum systems for computing? To explain this, let us first consider an example of a classical computation – the addition of two m-bit numbers, which we write as follows (the i's and j's denote the bits of the summands, the k's stand for the bits of the result):

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

This computation is realized using circuits of classical gates (`and, or, not, xor`) in the usual way: add the last bits, then the second last plus the carry bit etc. Let us emphasize here: *Classically, we add two numbers at a time.*

Now, which operations are allowed in a quantum system? 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 unitary transforms – (quantum) gates on the system.** A typial set is the following:

One qubit gates: They manipulate only one component of the tensor product $H_1 \otimes H_1 \otimes \ldots \otimes H_1$:

Hadamard gate: $H_1 \to H_1$

$$|0\rangle \to \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$|1\rangle \to \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

(a unitary transform is uniquely determined by its values on the elements of a basis)

phase shift: For each parameter $\theta \in [0, 2\pi)$ a quantum gate $H_1 \to H_1$ is defined by

$$|0\rangle \to |0\rangle$$
$$|1\rangle \to e^{\imath\theta} |1\rangle$$

Two qubit gates: They manipulate two components of $H_1 \otimes H_1 \otimes \ldots \otimes H_1$:
quantum xor gate (also called **controlled-not gate**): $H_1 \otimes H_1 \to H_1 \otimes H_1$

$$|0\rangle |0\rangle \to |0\rangle |0\rangle$$
$$|0\rangle |1\rangle \to |0\rangle |1\rangle$$
$$|1\rangle |0\rangle \to |1\rangle |1\rangle$$
$$|1\rangle |1\rangle \to |1\rangle |0\rangle$$

These gates form a universal system: Each unitary transform in H_m can be represented as a finite composition of these gates. If we restrict ourselves to one single phase shift with $\theta = \pi/4$, we obtain an approximately universal system: Each unitary transform can be approximated in the operator norm to arbitrary precision by suitable finite composition of these gates.

So once we can implement these gates we can carry out all unitary transforms (of course, the efficiency of such a representation or approximation is still an issue). Physicists are working on implementations of these gates in various quantum systems such as photons, trapped ions, magnetic resonance systems.

Let us mention two crucial features:

1. These gates can transform classical states into superpositions. Example: the Hadamard gate applied to the first and then to the second qubit

$$|0\rangle |0\rangle \longrightarrow \frac{1}{2} (|0\rangle |0\rangle + |0\rangle |1\rangle + |1\rangle |0\rangle + |1\rangle |1\rangle)$$

2. They act also on superpositions. Examples:

2.1. The quantum xor gate:

$$\alpha_0 |0\rangle |0\rangle + \alpha_1 |0\rangle |1\rangle + \alpha_2 |1\rangle |0\rangle + \alpha_3 |1\rangle |1\rangle$$
$$\downarrow$$
$$\alpha_0 |0\rangle |0\rangle + \alpha_1 |0\rangle |1\rangle + \alpha_2 |1\rangle |1\rangle + \alpha_3 |1\rangle |0\rangle$$

2.2. Quantum addition of binary numbers (a classical gate implementation can easily be turned into a quantum gate implementation):

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

Assuming that all $\alpha_{ij} \neq 0$, we see that starting with the superposition of all possible inputs, and carrying out the quantum implementation just once, we

obtain (by linearity of the quantum gates) the superposition of the results of all possible inputs.

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

But does that mean that we have an exponentially powerful parallel computer? No, because we cannot access all components of the superposition! According to quantum mechanics, we have to measure the system, which destroys the superposition. So:

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

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

Coming back to our example of quantum addition of binary numbers we would get

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

with probability $|\alpha_{ij}|^2$. This is not much of a gain: just one single result, and on top of that a random one!

The reasoning above indeed showed typical features of a quantum computation, but, on the other hand, made it also plausible, that in order to let a quantum computer behave more efficiently than a classical one, more ingenious and sophisticated techniques are required. We will briefly discuss some of them in section 5.

To study numerical problems in the quantum setting, a few more preparations are required. First of all, we shall view a numerical problem as given by an operator $S : F \to G$, the solution operator. Here F is a set (usually a set of functions), G is a normed space (either a space of functions or the scalar field), and $S(f) \in G$ is the (exact) solution of the problem at input $f \in F$.

If we consider the example of numerical integration, we have that F is a set of functions on some domain D, $G = \mathbf{R}$, and

$$S(f) = \int_D f(t)dt.$$

How does the quantum algorithm get information about $f \in F$? It is helpful to look first at the binary case: Let

$$f : \{0, 1, \ldots, 2^{m_1} - 1\} \to \{0, 1\}.$$

A classical black box (query, subroutine) produces $f(i)$ at request i, that is, maps $(i, 0) \to (i, f(i))$. Quite similarly, the quantum (binary) query:

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

This mapping has many extensions to a bijection of classical states, and hence, to a unitary operator $Q_f : H_m \to H_m$. The following is customary:

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

(where $\oplus$ stands for addition modulo 2).

For problems of analysis we have to consider the general case of functions f from a domain D to $\mathbf{R}$ (or, analogously, to $\mathbf{C}$). The appropriate quantum query $Q_f : H_m \to H_m$ is defined as follows:

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

where $1 \le m_1 < m$, H_m is identified with $H_{m_1} \otimes H_{m-m_1}$,

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

maps indices i to nodes $\tau(i) \in D$, the mapping

$$\beta : \mathbf{R} \to \{0, \dots, 2^{m-m_1} - 1\}$$

encodes the real number $f(\tau(i))$ as a binary integer $\beta(f(\tau(i)))$, and $\oplus$ stands for addition modulo 2^{m-m_1} (the choice of m_1, τ and β is part of the algorithm design). Thus we arrived at the **quantum model of computation for numerical problems:**

starting state:
$|i_0\rangle \in H_m$ (a classical state)
computation:
$|i_0\rangle \to U_0\,|i_0\rangle \to Q_f U_0\,|i_0\rangle \to U_1 Q_f U_0\,|i_0\rangle \to \dots$
$\to U_n Q_f U_{n-1} \dots Q_f U_1 Q_f U_0\,|i_0\rangle =: |\xi\rangle$
measurement:
$|\xi\rangle = \sum_{i=0}^{2^m-1} \alpha_i\,|i\rangle \to |i\rangle$ with probability $|\alpha_i|^2$
output:
$|i\rangle \to \varphi(i) =: A_n(f) \in G$

The U_i represent the composition of the quantum gates applied before, between, and after the queries, and φ symbolizes any classical computation performed on the measurement result i to obtain the final output. We call this a quantum algorithm with n queries. The output $A_n(f)$ is a random variable.

We introduce the (probabilistic) error of A_n at input f:

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

and the error of A_n over F by

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

Note that the choice of the probability threshold 3/4 is inessential: By repeating the algorithm k times and computing the median of the results, the success probability can be increased to $1 - 2^{-ck}$ for some $c > 0$ not depending on k.

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

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

It gives the minimal possible error among all quantum algorithms which use at most n quantum queries. For the problems we consider here, this is, up to logarithmic factors, also the best possible errors among all algorithms of cost (number of gates, queries, and measurements) at most n. Along with $e_n^{\mathrm{q}}(S,F)$ we shall also consider
$e_n^{\mathrm{det}}(S,F)$, the best possible error among all deterministic classical algorithms with cost (number of arithmetic operations, function values) $\le n$, and
$e_n^{\mathrm{ran}}(S,F)$, the best possible error among all randomized classical algorithms with cost (number of random generator calls, arithmetic operations, function values) $\le n$.

For detailed definitions and references for the respective results in the classical settings we refer to [25], [34], [10].

3 Multivariate Integration

We let $D = [0,1]^d$, $f : [0,1]^d \to \mathbf{R}$,

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

and consider the following function classes: Let $r \in \mathbf{N}_0$, $0 < s \le 1$ and define the Hölder classes by

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

Here $C^r([0,1]^d)$ is the set of r times continuously differentiable functions and ∂^α is the partial derivative corresponding to the multiindex α.

Next define the Sobolev classes for $r \in \mathbf{N}$, $1 \le p \le \infty$, satisfying $r/d > 1/p$ (Sobolev embedding condition), by

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

with ∂^α being the respective weak partial derivative. We will be particularly interested in the behaviour of the complexity in the various settings for large d.

The following result is due to Novak [26].

Theorem 1.

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

Let us compare this with the classical deterministic and randomized setting:

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

We write $a_n \asymp b_n$ for sequences of nonnegative reals (a_n) and (b_n) if there are constants $c_1, c_2 > 0$, $n_0 \in \mathbf{N}$, such that $c_1 a_n \leq b_n \leq c_2 a_n$ for all $n \in \mathbf{N}$ with $n \geq n_0$.

It is interesting to look at these rates for small $(r+s)/d$ (that is, for large d). We see that in the classical deterministic setting, the exponent is negligible, meaning there is no chance to solve this problem deterministically. This is the well-known curse of dimension. In the classical randomized settting the situation is different: The exponent is always smaller than $-1/2$ even for very small $(r+s)/d$, corresponding to the fact that randomization (Monte Carlo integration) allows to overcome the curse of dimension. Comparing, finally, the quantum setting with the classical randomized setting, we have essentially a quadratic speedup (the exponent close to $-1/2$ is replaced by an exponent close to -1). This is the same sort of speedup as in the Grover search algorithm mentioned in the beginning.

Novak's result settled the Hölder case, that is, function classes related to the maximum norm, but what about function spaces involving the L_2, or, more generally, the L_p norm for $1 \leq p < \infty$. Since L_2 is the natural space for Monte Carlo algorithms, it was an open, interesting question whether Monte Carlo algorithms could possibly be as good as quantum algorithms, say for $p = 2$? Furthermore, it was known that for $p = 1$, Monte Carlo algorithms do not yield a speedup over (classical) deterministic algorithms. Will quantum algorithms do so? These questions are answered in the following theorem from [12]. To suppress inessential for the purpose of our survey logarithmic factors, we write $a_n \asymp_{\log} b_n$ if there are constants $c_1, c_2 > 0$, $n_0 \in \mathbf{N}$, $\alpha_1, \alpha_2 \in \mathbf{R}$ such that

$$c_1(\log(n+1))^{\alpha_1} a_n \leq b_n \leq c_2(\log(n+1))^{\alpha_2} a_n$$

for all $n \in \mathbf{N}$ with $n \geq n_0$.

Theorem 2. *Let $1 \leq p < \infty$, $r, d \in \mathbf{N}$, $r/d > 1/p$. Then*

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

In the classical deterministic setting we have

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

while in the classical randomized setting the following holds:

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

We see that the same spedup of the quantum over the classical randomized setting (a gain of $-1/2$ in the exponent) holds through for all $p \geq 2$. For $1 \leq p < 2$ the gain is even greater, reaching -1 for $p = 1$, the case where classical randomization does not yield any gain over classical deterministic algorithms.

4 Approximation of Sobolev Embeddings

It was well-known that Monte Carlo methods are especially suited for problems whose output is a scalar (integration, computation of functionals of solutions of integral equations). The integration results presented above are of this kind, leaving the question how quantum algorithms would behave if the output were not a scalar, but a function. A particularly typical situation is function approximation – we are asked to compute an approximation to a function using (a limited number of) values of that function.

We let $1 \le p, q \le \infty$,

$$S = J_{pq} : W_p^r([0,1]^d) \to L_q([0,1]^d), \quad J_{pq}(f) = f,$$

and put $F = \mathcal{B}(W_{p,d}^r)$. Thus, given $f \in \mathcal{B}(W_{p,d}^r)$, we seek to approximate f in the norm of $L_q([0,1]^d)$. The following was shown in [13], [14].

Theorem 3. *Let $r, d \in \mathbf{N}$, $1 \le p, q \le \infty$ and assume $r/d > \max(1/p, 2/p - 2/q)$. Then*

$$e_n^q(J_{pq}, \mathcal{B}(W_p^r(D))) \asymp_{\log} n^{-r/d}.$$

Again it is instructive to compare with the classical deterministic and randomized setting:

$$\begin{aligned} e_n^{\det}(J_{pq}, \mathcal{B}(W_{p,d}^r)) &\asymp e_n^{\mathrm{ran}}(J_{pq}, \mathcal{B}(W_{p,d}^r)) \\ &\asymp \begin{cases} n^{-r/d} & \text{if} \quad p \ge q \\ n^{-r/d+1/p-1/q} & \text{if} \quad p < q. \end{cases} \end{aligned}$$

We observe a possible improvement of n^{-1} (for $p = 1$, $q = \infty$) of quantum algorithms over the classical deterministic and randomized case. This is the same speedup as in Theorem 2 for $p = 1$. We also see that there are regions of the parameter domain where the speedup is smaller, and others, where there is no speedup at all.

5 Further Comments

Let us summarize the results in a table (suppressing again constants and logarithmic factors).

	deterministic	random	quantum
Integration			
$\mathcal{B}(F_d^{r,s})$	$n^{-(r+s)/d}$	$n^{-(r+s)/d-1/2}$	$n^{-(r+s)/d-1}$
$\mathcal{B}(W_{p,d}^r)$, $2 \le p \le \infty$	$n^{-r/d}$	$n^{-r/d-1/2}$	$n^{-r/d-1}$
$\mathcal{B}(W_{p,d}^r)$, $1 \le p < 2$	$n^{-r/d}$	$n^{-r/d-1+1/p}$	$n^{-r/d-1}$
Approximation			
$\mathcal{B}(W_{p,d}^r) \to L_q$, $p \ge q$	$n^{-r/d}$	$n^{-r/d}$	$n^{-r/d}$
$\mathcal{B}(W_{p,d}^r) \to L_q$, $p < q$	$n^{-r/d+1/p-1/q}$	$n^{-r/d+1/p-1/q}$	$n^{-r/d}$

We mentioned in section 2 that a naive view on quantum computation does not bring us very far. So what *are* the algorithmic methods that make quantum computers superior to classical ones? There is, first of all, the quantum Fourier transform, a highly efficient implementation of the discrete Fourier transform on a quantum system. Based on this, Shor [31] developed a technique of estimating eigenvalues of unitary operators, which eventually lead to his seminal results on factoring. The crucial idea of the Grover search is an iterative amplification of the amplitude of the state we are interested in (the state $|i_0\rangle$ with i_0 such that $f(i_0) = 1$). Finally, Boyer, Brassard, Høyer, Mosca, and Tapp [3], [4], combined this by estimating the eigenvalues of the Grover transform using the Shor approach and this way produced an efficient counting algorithm (estimating the number of 1's in a huge sequence of bits, or, equivalently, estimating the mean of a sequence of bits). For a good, self-contained exposition of these basic techniques see [24]). The lower bound results by Nayak and Wu [23] are derived by the polynomial method [2]: the succcess probability is a polynomial (in the bits the mean of which is to be computed) of degree at most the number of queries. Interesting from the point of view of approximation theory: Nayak and Wu use the Bernstein and Markov inequalities for polynomials to get their result.

Novak's Theorem 1 is built on these results, combining them with techniques from information-based complexity [25], [34], in particular for the lower bound proof. The upper bound is shown by adopting a technique from Monte Carlo methods: separation of the main part (control variate).

These were L_∞-results exclusively. The step from $p = \infty$ to arbitrary p needed for Theorem 2 is based on respective results for mean computation in finite dimensional l_p^N spaces [11], [16]. Those are achieved by splitting the function into dyadic levels, distributing queries over levels, combining decay of means and precise error estimates for counting. In the case of $1 \le p < 2$ a combination with

the Grover search is used. A new discretization technique (inspired by Maiorov's technique [20] from approximation theory), reducing integration to a sequence of mean computation problems [12], leads to Theorem 2.

Related techniques (Grover search, multilevel splittings, discretization) are also used in [13], [14] to prove the upper bounds in Theorem 3. The lower bound technique is a new one: multiplicativity of minimal quantum errors. This was inspired by functional analysis – the multiplicativity of s-numbers [29].

References

1. D. Aharonov. Quantum computation – a review, in: D. Stauffer (Ed.) *Annual Review of Computational Physics*, vol. VI, World Scientific, Singapore, 1998, see also http://arXiv.org/abs/quant-ph/9812037.
2. R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf. Quantum lower bounds by polynomials, *Proceedings of 39th IEEE FOCS*, 352-361, 1998, see also http://arXiv.org/abs/quant-ph/9802049.
3. M. Boyer, P. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching, *Fortschritte der Physik*, 46, 493 – 505, 1998, see also http://arXiv.org/abs/quant-ph/9605034.
4. G. Brassard, P. Høyer, M. Mosca, A. Tapp. Quantum amplitude amplification and estimation, In: *Quantum Computation and Quantum Information: A Millennium Volume*, AMS Contemporary Mathematics Series, Volume 305, 2002, see also http://arXiv.org/abs/quant-ph/0005055.
5. D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer, *Proc. R. Soc. Lond.*, Ser. A 400, 97-117, 1985.
6. A. Ekert, P. Hayden, H. Inamori. Basic concepts in quantum computation, 2000, see http://arXiv.org/abs/quant-ph/0011013.
7. R. Feynman. Simulating physics with computers, *Int. J. Theor. Phys.*, 21, 467–488, 1982.
8. L. Grover. A fast quantum mechanical algorithm for database search, *Proc. 28 Annual ACM Symp. on the Theory of Computing*, 212–219, ACM Press New York, 1996, see also http://arXiv.org/abs/quant-ph/9605043.
9. J. Gruska. *Quantum Computing*, McGraw-Hill, London, 1999.
10. S. Heinrich. Random approximation in numerical analysis, in: K. D. Bierstedt, A. Pietsch, W. M. Ruess, D. Vogt (Eds.), *Functional Analysis*, Marcel Dekker, New York, 1993, 123–171.
11. S. Heinrich. Quantum summation with an application to integration, *J. Complexity*, 18, 1–50, 2002, see also http://arXiv.org/abs/quant-ph/0105116.
12. S. Heinrich. Quantum integration in Sobolev classes, *J. Complexity*, 19, 19–42, 2003, see also http://arXiv.org/abs/quant-ph/0112153.
13. S. Heinrich. Quantum Approximation I. Embeddings of Finite Dimensional L_p Spaces, *J. Complexity*, 20, 5–26, 2004, see also http://arXiv.org/abs/quant-ph/0305030.
14. S. Heinrich. Quantum Approximation II. Sobolev Embeddings, *J. Complexity*, 20, 27–45, 2004, see also http://arXiv.org/abs/quant-ph/0305031.
15. S. Heinrich. The quantum query complexity of elliptic PDE, in preparation.
16. S. Heinrich, E. Novak. On a problem in quantum summation, *J. Complexity*, 19, 1–18, 2003, see also http://arXiv.org/abs/quant-ph/0109038.

17. B. Kacewicz. Randomized and quantum algorithms yield a speed-up for initial-value problems, *J. Complexity*, 20, 821-834, 2004, see also http://arXiv.org/abs/quant-ph/0311148.
18. B. Kacewicz. Improved bounds on the randomized and quantum complexity of initial-value problems, see http://arXiv.org/abs/quant-ph/0405018.
19. M. Kwas. Complexity of multivariate Feynman-Kac path integration in randomized and quantum settings, see http://arXiv.org/abs/quant-ph/0410134.
20. V. E. Maiorov. Discretization of the problem of diameters, *Usp. Mat. Nauk*, 30, No. 6 (186), 1975, 179–180 (in Russian).
21. Yu. I. Manin. Computable and uncomputable, *Sovetskoye Radio*, Moscow, 1980 (in Russian).
22. Yu. I. Manin. Classical computing, quantum computing, and Shor's factoring algorithm, 1999, see http://arXiv.org/abs/quant-ph/9903008.
23. A. Nayak, F. Wu. 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, I. L. Chuang. *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.
25. E. Novak. *Deterministic and Stochastic Error Bounds in Numerical Analysis*, Lecture Notes in Mathematics 1349, Springer-Verlag, Berlin, 1988.
26. E. Novak. Quantum complexity of integration, *J. Complexity*, 17, 2–16, 2001, see also http://arXiv.org/abs/quant-ph/0008124.
27. E. Novak, I. H. Sloan, H. Woźniakowski. Tractability of approximation for weighted Korobov spaces on classical and quantum computers, *Found. Comput. Math.* 4, 121-156, 2004, see also http://arXiv.org/abs/quant-ph/0206023.
28. A. Papageorgiou, H. Woźniakowski. Classical and quantum complexity of the Sturm-Liouville eigenvalue problem, see http://arXiv.org/abs/ quant-ph/0502054.
29. A. Pietsch. *Eigenvalues and s-Numbers*, Cambridge University Press, 1987.
30. A. O. Pittenger. *Introduction to Quantum Computing Algorithms*, Birkhäuser, Boston, 1999.
31. P. W. Shor. 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, 1994, see also http://arXiv.org/abs/quant-ph/9508027.
32. P. W. Shor. Introduction to quantum algorithms, 2000, see http://arXiv.org/abs/quant-ph/0005003.
33. J. F. Traub, H. Woźniakowski. Path integration on a quantum computer, *Quantum Information Processing*, 1(5), 365-388, 2002, see also http://arXiv.org/abs/quant-ph/0109113.
34. J. F. Traub, G. W. Wasilkowski, H. Woźniakowski. *Information-Based Complexity*, Academic Press, New York, 1988.
35. C. Wiegand. Quantum complexity of parametric integration, *J. Complexity*, 20, 75-96, 2004, see also http://arXiv.org/abs/quant-ph/0305103.