Opuscula Mathematica • Vol. 28 • No. 3 • 2008

Maciej Goćwin, Marek Szczęsny

RANDOMIZED AND QUANTUM ALGORITHMS FOR SOLVING INITIAL-VALUE PROBLEMS IN ORDINARY DIFFERENTIAL EQUATIONS OF ORDER k

Abstract. The complexity of initial-value problems is well studied for systems of equations of first order. In this paper, we study the ε-complexity for initial-value problems for scalar equations of higher order. We consider two models of computation, the randomized model and the quantum model. We construct almost optimal algorithms adjusted to scalar equations of higher order, without passing to systems of first order equations. The analysis of these algorithms allows us to establish upper complexity bounds. We also show (almost) matching lower complexity bounds. The ε-complexity in the randomized and quantum setting depends on the regularity of the right-hand side function, but is independent of the order of equation. Comparing the obtained bounds with results known in the deterministic case, we see that randomized algorithms give us a speed-up by 1/2, and quantum algorithms by 1 in the exponent. Hence, the speed-up does not depend on the order of equation, and is the same as for the systems of equations of first order.

We also include results of some numerical experiments which confirm theoretical results.

Keywords: k-th order initial-value problems, randomized computing, quantum computing, optimal algorithms, complexity.

Mathematics Subject Classification: 68Q25, 65L05, 68W20.

1. INTRODUCTION

The optimal solution of initial-value problems has been widely studied for systems of first order equations. Such systems were considered in the worst-case and asymptotic deterministic setting ([4]), and in the randomized and quantum settings in [5, 6]. For right-hand side functions with r continuous bounded derivatives, the ε-complexity was shown to be essentially $(1/\varepsilon)^{1/(r+\varphi)}$, where $\varphi = 0$ in the deterministic setting, $\varphi = 1/2$ in the randomized setting, and $\varphi = 1$ in the quantum setting (for details see

Section 3). A speed-up for systems of equations over deterministic computing is thus by 1/2 in the exponent in the randomized case, and by 1 in the quantum case.

In this paper, we deal with the ε-complexity of initial-value problems for scalar equations of order k. The complexity of equations of order k has so far been studied in the deterministic worst-case setting. It was shown in [9] that the ε-complexity for standard information is $\Theta\big((1/\varepsilon)^{1/r}\big)$. If the right-hand side function g depends on the solution function only, then the ε-complexity for linear information is $\Theta\big((1/\varepsilon)^{1/(r+k)}\big)$. This shows how the order k of the equation contributes to the ε-complexity in the worst-case setting.

In this paper, we show almost tight complexity bounds in the randomized and quantum settings. In particular, we explain the dependence of the ε-complexity on k. An improvement dependent on k in the deterministic worst-case setting was achieved by passing from the standard to linear (integral) information. One may hope that proper randomized or quantum approximation of the integrals involved in the computations will lead to algorithms with improved error bounds dependent on k. In this paper we show that such an improvement is not possible, and a speed-up in the randomized and quantum setting is independent of k. We show lower complexity bounds of order $(1/\varepsilon)^{1/(r+1/2)}$ in the randomized setting, and $(1/\varepsilon)^{1/(r+1)}$ in the quantum setting, no matter how large k is.

We define such algorithms designed for scalar initial-value problems of order k which do not require a transformation of the problem into a system of first order equations. The error analysis of these algorithms leads to (almost) matching upper complexity bounds.

Let us remark that, although a scalar equation of order k can be written as a special system of $k+1$ equations of the first order, we cannot directly apply the upper bounds from [6], since the right-hand side function is then unbounded.

Numerical tests performed in the randomized case confirm theoretical results. The speed-up with exponent $1/2-\gamma$ is observed, where γ is a small positive constant.

The paper is organized as follows. After introducing necessary definitions in the three settings (Section 2), we recall in Section 3 for further comparison known complexity bounds for initial-value problems. In Section 4 we define randomized and quantum algorithms for scalar initial-value problems of order k. In Section 5 we analyse the error and cost analysis of these algorithms. Lower bounds on the ε-complexity in the randomized and quantum settings are shown in Section 6. In Section 7 we present results of numerical tests in the randomized case, which confirm theoretical results.

2. PROBLEM DEFINITION AND BASIC NOTATION

We consider the complexity of a problem in the following form

$$\begin{cases} u^{(k)}(x) = g\left(x, u(x), u'(x), \ldots, u^{(q)}(x)\right), & x \in [a,b], \\ u^{(j)}(a) = u_a^j, \quad j = 0, 1, \ldots, k-1, \end{cases} \tag{1}$$

where $0 \le q < k$, $g : [a,b] \times \mathbb{R}^{q+1} \to \mathbb{R}$, $u : [a,b] \to \mathbb{R}$ $(a < b)$.

For $r \geq 1$ and given positive numbers $D_0, \ldots, D_r$, we consider the class of right-hand side functions g defined by

$$\begin{gathered}\mathcal{G}^r = \big\{ g \,|\, g \in C^{(r)}([a,b] \times \mathbb{R}^{q+1}), \quad |\partial^j g(x,y)| \leq D_j, \quad \text{for} \\ x \in [a,b], \quad y \in \mathbb{R}^{q+1}, \quad j = 0, 1, \ldots r \big\},\end{gathered} \tag{2}$$

where $\partial^j g$ represents all partial derivatives of order j of g.

Instead of (1) we can write an equivalent system of differential equations of the first order of the form:

$$\mathbf{u}'(x) = \begin{bmatrix} u_0'(x) \\ u_1'(x) \\ \vdots \\ u_{k-1}'(x) \\ u_k'(x) \end{bmatrix} = \begin{bmatrix} 1 \\ u_2(x) \\ \vdots \\ u_k(x) \\ g(u_0(x), u_1(x), \ldots, u_{q+1}(x)) \end{bmatrix} = \mathbf{g}(\mathbf{u}(x)), \; x \in [a,b], \tag{3}$$

with initial conditions

$$\mathbf{u}(a) = \left[a, u_a^0, \ldots, u_a^{k-1}\right]^T. \tag{4}$$

Then the solution $u(x)$ of (1) corresponds to the function $u_1(x)$.

Before we start analyzing the complexity of (1), we recall some definitions. We are interested in finding a bounded function $l = l(x)$ that approximates the solution of (1). The construction of l is based on certain information on the right-hand side function g. In the deterministic setting, we usually consider standard information, in which we compute values of g or its partial derivatives at some points, or linear information in which we know values of linear functionals of g.

In the randomized setting the values of g or its partial derivatives can be computed at randomly chosen points. In the quantum setting, information about g is gathered by applying a quantum query for g. The reader is referred to [2] for a detailed explanation of what a quantum query is.

To get an approximate solution $l(x)$, we use an algorithm A, which is a mapping from the information space into the space of bounded functions. We assume that A uses M information values.

In the deterministic setting, the worst-case error of an algorithm A in the class $\mathcal{G}^r$ is defined by

$$e_M^{\text{worst}}(A, \mathcal{G}^r) = \sup_{g \in \mathcal{G}^r} \sup_{x \in [a,b]} |u(x) - l(x)|. \tag{5}$$

In the randomized and quantum settings, the approximation obtained is random. Letting $(\Omega, \Sigma, \mathbf{P})$ be a probability space, an algorithm A provides us with an approximate solution l^ω, where $\omega \in \Omega$. The local error of the algorithm A at g is defined by

$$e_M^\omega(A, g) = \sup_{x \in [a,b]} |u(x) - l^\omega(x)| \tag{6}$$

(we assume that $e_M^{\omega}(A,g)$ is a random variable for each $g \in \mathcal{G}^r$). The error of A in the class $\mathcal{G}^r$ in the randomized setting is defined by

$$e_M^{\mathrm{rand}}(A,\mathcal{G}^r) = \sup_{g\in\mathcal{G}^r} \left(\mathbf{E}(e_M^{\omega}(A,g))^2\right)^{1/2}, \tag{7}$$

and in the quantum setting by

$$e_M^{\mathrm{quant}}(A,\mathcal{G}^r) = e_M^{\mathrm{quant}}(A,\mathcal{G}^r,\delta) = \sup_{g\in\mathcal{G}^r} \inf\{\alpha \mid \mathbf{P}(e_M^{\omega}(A,g) > \alpha) \le \delta\}. \tag{8}$$

The number $\delta \in (0,1/2)$ here denotes the failure probability. It is often assumed that $\delta = 1/4$. The success probability can then be increased by taking a median of the number of repetitions of an algorithm A (see [2] for algorithms whose outputs are real numbers and [3] when the outputs are elements of some normed space).

By the cost in the deterministic, randomized and quantum setting, we mean the number M of subroutine calls for g used by an algorithm A. Thus, in the deterministic and randomized setting, the cost is equal to the number of evaluations of g or its partial derivatives, while in the quantum setting it is the number of quantum queries. We will denote the cost in the respective setting by $\mathrm{cost}^{\mathrm{worst}}(A)$, $\mathrm{cost}^{\mathrm{rand}}(A)$ or $\mathrm{cost}^{\mathrm{quant}}(A)$.

For any $\varepsilon > 0$, by the ε-complexity of the problem we mean the minimal cost sufficient to solve the problem with error no larger than ε, where the minimum is taken over all algorithms solving the problem

$$\mathrm{comp}(\mathcal{G}^r,\varepsilon) \quad = \quad \min_A \{\mathrm{cost}(A) \mid e_M(A,\mathcal{G}^r) \le \varepsilon\}. \tag{9}$$

To denote the complexity in the deterministic, randomized or quantum settings, we will use a suitable superscript: "worst", "rand" or "quant". Additionally, to denote different types of information used in the deterministic setting, we will use indices: "worst-st" and "worst-lin" for standard and linear information, respectively.

3. KNOWN COMPLEXITY BOUNDS

In this section we briefly recall known complexity bounds for scalar equations of order k, as well as those for systems of the first order.

In [6], Kacewicz dealt with systems of equations of the first order of the form

$$z'(t) = f(z(t)), \quad t \in [a,b], \quad z(a) = \eta, \tag{10}$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ and $\eta \in \mathbb{R}^d$. He considered the Hölder class of functions

$$\begin{aligned}\mathcal{F}^{r,\rho} = \big\{ f : \mathbb{R}^d \to \mathbb{R}^d \mid f \in C^{(r)}(\mathbb{R}^d), \quad |\partial^i f^j(y)| \le D_i,\ i = 0,1,\dots,r,\\ |\partial^r f^j(y) - \partial^r f^j(z)| \le H\|y-z\|^{\rho},\ y,z \in \mathbb{R}^d, \quad j = 1,2,\dots,d\big\},\end{aligned} \tag{11}$$

where $\rho \in (0,1]$.

It was shown in [6] that the ε-complexity is

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{F}^{r,\rho},\varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+\rho+1/2-\gamma)}\right) \tag{12}$$

and

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{F}^{r,\rho},\varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+\rho+1-\gamma)}\right) \tag{13}$$

with an arbitrarily small positive parameter γ. (The constants in the big-O notation depend on γ and are independent of η). These bounds are almost optimal, i.e., they essentially match lower bounds on the complexity.

It is easy to see that the above bounds with $\rho = 0$ hold for the class $\mathcal{F}^r$, where

$$\mathcal{F}^r = \big\{ f : \mathbb{R}^d \to \mathbb{R}^d \mid f \in C^{(r)}(\mathbb{R}^d), \quad |\partial^i f^j(y)| \le D_i,\ i = 0,1,\ldots,r, \\ y \in \mathbb{R}^d, \quad j = 1,2,\ldots,d \big\}. \tag{14}$$

For systems (10) the ε-complexity in the class $\mathcal{F}^r$ is thus equal to

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{F}^r,\varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1/2-\gamma)}\right) \tag{15}$$

and

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{F}^r,\varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1-\gamma)}\right). \tag{16}$$

Although equation (1) can be transformed into system (3) of the first order, we cannot directly use the above results for our problem, since the right-hand side function g is unbounded.

In this paper we present such algorithms for solving problem (1) which do not require a transformation of the equation of order k into a system of first order equations. We also ask if bounds (15) and (16) can be improved for problem (1) due to the increased regularity of the solution. In some cases in the deterministic worst-case setting such a speed-up dependent on k can indeed be shown. It was shown in [9] that for standard information

$$\mathrm{comp}^{\text{worst-st}}(\mathcal{G}^r,\varepsilon) = \Theta\left(\left(\frac{1}{\varepsilon}\right)^{1/r}\right), \tag{17}$$

so that there is no dependence on k in this case. However, if we admit linear information on right-hand side function, we can achieve a better result. The use of integral information leads (for $q = 0$) to the upper bound

$$\mathrm{comp}^{\text{worst-lin}}(\mathcal{G}^r,\varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+k)}\right). \tag{18}$$

The complexity in this case significantly depends on k. (The constants in the "Θ" and "O" notation depend on initial values.)

Intuitively, one may expect a speed-up dependent on k by replacing integrals in deterministic algorithms by effective randomized or quantum approximations. In Section 6 we show lower bounds on the complexity in both settings, which proves this intuition to be wrong.

4. ALGORITHMS

In this section we describe algorithms using randomized or quantum computation which solve (1) in the class $\mathcal{G}^r$ of right-hand side functions. Obviously, we can solve (1) by transforming the equation of order k into a system of ordinary differential equations of the first order, see (3), and then use the almost optimal algorithms of Kacewicz [6]. We here construct slightly different algorithms which do not require a transformation into a first order system, but they make use of a specific form of problem (1). The general idea and the main points of the error analysis are similar to those in [6]. The differences result from a specific form of (1), so that the error analysis requires a modified proof technique.

As in [6], the algorithms are defined recursively on intervals of decreasing length. Let us denote by $A^s([c,d],n,Y)$ an algorithm solving the problem

$$\begin{cases} u^{(k)}(x) = g\left(x, u(x), u'(x), \ldots, u^{(q)}(x)\right), & x \in [c,d], \\ u^{(j)}(c) = y^j, & j = 0,1,\ldots,k-1, \end{cases} \tag{19}$$

on interval $[c,d]$ with a vector of initial conditions $Y = [y^0, \ldots, y^{k-1}] \in \mathbb{R}^k$. The parameter s denotes here the depth of recursion, and n, known as the *basic parameter*, tells us how many subintervals we consider in the interval $[c,d]$.

We will denote the approximations computed by the algorithm A^s by $\hat{l}^s$.

The idea of the recursive algorithms is as follows:

Algorithm $A^s([c,d],n,Y)$:

1. For $s = 1$, approximation $\hat{l}^1$ in $A^1([c,d],n,Y)$ is given by Taylor's algorithm (see equations (21), (23) below) on $[c,d]$ with step size $(d-c)/n$. Consider a uniform partition of $[c,d]$ with points $x_i := c + ih$, where $h := (d-c)/n$ and $i = 0,1,\ldots,n$. Let $y_0^j := y^j$, and for given y_i^j, let $\bar{u}_i$ be the solution of a local problem

$$\begin{cases} \bar{u}_i^{(k)}(x) = g\left(x, \bar{u}_i(x), \bar{u}_i'(x), \ldots, \bar{u}_i^{(q)}(x)\right), & x \in [x_i, x_{i+1}], \\ \bar{u}_i^{(j)}(x_i) = y_i^j, & j = 0,1,\ldots,k-1, \end{cases} \tag{20}$$

   and

$$\hat{l}_i^0(x) = \sum_{j=0}^{r+k-1} \frac{\bar{u}_i^{(j)}(x_i)}{j!}(x - x_i)^j, \qquad x \in [x_i, x_{i+1}], \tag{21}$$

   be its Taylor approximation.

We define

$$y_{i+1}^{j} := \hat{l}_i^{0\,(j)}(x_{i+1}). \tag{22}$$

The approximation $\hat{l}^1(x)$ is given as a continuous function on $[a,b]$ that coincides with $\hat{l}_i^0(x)$ on each subinterval,

$$\hat{l}^1(x) = \hat{l}_i^0(x) \qquad \text{for} \quad x \in [x_i, x_{i+1}]. \tag{23}$$

Suppose that the algorithm $A^{s-1}([x_i, x_{i+1}], m, Y_i)$ is defined for any $[x_i, x_{i+1}]$, m and initial vector Y_i. We now show how to get $A^s([c,d], n, Y)$ from $A^{s-1}([x_i, x_{i+1}], m, Y_i)$ for $s \geq 2$.

2. Divide an interval $[c,d]$ into n subintervals with the endpoints $x_i = c + i(d-c)/n$, $i = 0, 1, \ldots, n$.
3. Given Y, set $Y_0 = Y$.
4. Set $m = n^2$ in the randomized case and $m = n$ in the quantum case.
5. For $i = 0, \ldots, n-1$, given Y_i, compute an approximation $\hat{l}_i^{s-1}$ using algorithm $A^{s-1}([x_i, x_{i+1}], m, Y_i)$ on $[x_i, x_{i+1}]$, and then compute new initial values Y_{i+1} (see equation (30) below).
6. As a result, an algorithm $A^s([c,d], n, Y)$ returns an approximation $\hat{l}^s$ given by $\hat{l}^s(x) = \hat{l}_i^{s-1}(x)$ for $x \in [x_i, x_{i+1})$ and $\hat{l}^s(d) = \hat{l}_{n-1}^{s-1}(d)$.

Let $s = K$ be the index of the final algorithm. To solve problem (1), we run the recursive algorithm with $s = K$ for the interval $[a,b]$ and initial vector $Y = \left[u_a^0, \ldots, u_a^{k-1}\right]$. From the construction, the approximation given by algorithm A^K is a piecewise polynomial function formed by the joined results of Taylor's approximations over subintervals of $[a,b]$.

We now show how to compute the vector of initial values Y_{i+1} in Step 5 of the algorithm A^s. Assume that we are given the initial vector $Y_i = [y_i^0, \ldots, y_i^{k-1}]$ and the result $\hat{l}_i^{s-1}$ of algorithm $A^{s-1}([x_i, x_{i+1}], m, Y_i)$ on $[x_i, x_{i+1}]$, where m is defined in Step 4. The function $\hat{l}_i^{s-1}$ is an approximation of the solution $\bar{u}_i$ of the local problem on interval $[x_i, x_{i+1}]$ of form (20). Before providing the formula for Y_{i+1}, we give some properties of solutions of the local problem and some necessary definitions.

It is easy to see that for $j = 0, \ldots, k-1$ the solution $\bar{u}_i$ satisfies

$$\begin{aligned} \bar{u}_i^{(j)}(x_{i+1}) = & \sum_{p=0}^{k-1-j} y_i^{j+p} \frac{h^p}{p!} + \\ & + \int_{x_i}^{x_{i+1}} \int_{x_i}^{t_{k-j-1}} \cdots \int_{x_i}^{t_1} g\left(t, \bar{u}_i(t), \bar{u}_i'(t), \ldots, \bar{u}_i^{(q)}(t)\right) dt\, dt_1 \ldots dt_{k-j-1} \end{aligned} \tag{24}$$

(it is sufficient to integrate equation (20) $k-j$ times). Integrating by parts, we may transform the multiple integral into an integral with a weight. We get

$$\bar{u}_i^{(j)}(x_{i+1}) = \sum_{p=0}^{k-1-j} y_i^{j+p}\frac{h^p}{p!} + \\ + \frac{1}{(k-1-j)!}\int_{x_i}^{x_{i+1}} (x_{i+1}-t)^{k-j-1}\, g\left(t, \bar{u}_i(t), \bar{u}_i'(t), \ldots, \bar{u}_i^{(q)}(t)\right)dt. \tag{25}$$

Let us define the number $l = n^{2^s-4}$ in the randomized setting or $l = n^{s-2}$ in the quantum one. Take the uniform partition of $[x_i, x_{i+1}]$ with points $x_{i,p} = x_i + p\bar{h}$, where $\bar{h} = h/(ml)$ and $p = 0, \ldots, ml-1$. Then

$$\bar{u}_i^{(j)}(x_{i+1}) = \sum_{p=0}^{k-1-j} y_i^{j+p}\frac{h^p}{p!} + \\ + \frac{1}{(k-1-j)!}\sum_{p=0}^{ml-1}\int_{x_{i,p}}^{x_{i,p+1}} (x_{i+1}-t)^{k-j-1} g\left(t, \bar{u}_i(t), \bar{u}_i'(t), \ldots, \bar{u}_i^{(q)}(t)\right)dt. \tag{26}$$

In the right-hand side of (26), we now add and subtract the integrals of functions $(x_{i+1}-t)^{k-j-1}\, g\left(t, \hat{l}_i^{s-1}(t), \ldots, \hat{l}_i^{s-1,(q)}(t)\right)$.

For $p = 0, \ldots, ml-1$ and $i = 0, \ldots, n-1$, let

$$w_{i,p}(t) = \sum_{\alpha=0}^{r-1}\frac{1}{\alpha!} g^{(\alpha)}\left(x_{i,p}, \hat{l}_i^{s-1}(x_{i,p}), \ldots, \hat{l}_i^{s-1,(q)}(x_{i,p})\right)(t-x_{i,p})^{\alpha}, \tag{27}$$

be a polynomial approximating the function $g\left(t, \hat{l}_i^{s-1}(t), \ldots, \hat{l}_i^{s-1,(q)}(t)\right)$. Let functions $g_{i,p}^j(\theta)$ for $j = 0, \ldots, k-1$ defined on $[0,1]$ be given by

$$g_{i,p}^j(\theta) = \frac{\left(h-(p+\theta)\bar{h}\right)^{k-j-1}\left(g\left(x_{i,p}+\theta\bar{h}, \ldots, \hat{l}_i^{s-1,(q)}(x_{i,p}+\theta\bar{h})\right) - w_{i,p}\left(x_{i,p}+\theta\bar{h}\right)\right)}{h^{k-j-1}\bar{h}^r(k-1-j)!}. \tag{28}$$

The notation $g^{(\alpha)}\left(x_{i,p}, \hat{l}_i^{s-1}(x_{i,p}), \ldots, \hat{l}_i^{s-1,(q)}(x_{i,p})\right)$ denotes here the α-th derivative of function $g\left(t, \hat{l}_i^{s-1}(t), \ldots, \hat{l}_i^{s-1,(q)}(t)\right)$ with respect to t at point $t = x_{i,p}$.

After few transformations we get a formula equivalent to (26)

$$\begin{aligned}\bar{u}_i^{(j)}(x_{i+1}) = &\sum_{p=0}^{k-1-j} y_i^{j+p}\frac{h^p}{p!} + \frac{1}{(k-1-j)!}\sum_{p=0}^{ml-1}\int\limits_{x_{i,p}}^{x_{i,p+1}}(x_{i+1}-t)^{k-j-1}\,w_{i,p}(t)dt+\\ &+\bar{h}^{r+1}h^{k-j-1}ml\frac{1}{ml}\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta+\\ &+\frac{1}{(k-1-j)!}\sum_{p=0}^{ml-1}\int\limits_{x_{i,p}}^{x_{i,p+1}}(x_{i+1}-t)^{k-j-1}\left(g\left(t,\bar{u}_i(t),\ldots,\bar{u}_i^{(q)}(t)\right)-\right.\\ &\left.-\,g\left(t,\hat{l}_i^{s-1}(t),\ldots,\hat{l}_i^{s-1,(q)}(t)\right)\right)dt.\end{aligned} \tag{29}$$

Neglecting the last term in (29) and approximating the mean value $1/ml\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta$ by (random or quantum) approximation $\mathrm{AP}_i^j(g)$, we get the final formula for $Y_{i+1}=[y_{i+1}^0,\ldots,y_{i+1}^{k-1}]$, given by

$$\begin{aligned}y_{i+1}^j = &\sum_{p=0}^{k-1-j} y_i^{j+p}\frac{h^p}{p!}+\\ &+\frac{1}{(k-1-j)!}\sum_{p=0}^{ml-1}\int\limits_{x_{i,p}}^{x_{i,p+1}}(x_{i+1}-t)^{k-j-1}\,w_{i,p}(t)dt+\bar{h}^r h^{k-j}\mathrm{AP}_i^j(g).\end{aligned} \tag{30}$$

This formula describes the way of getting Y_{i+1} in Step 5.

It remains to show the way to compute the (random or quantum) numbers $\mathrm{AP}_i^j(g)$, $j=0,\ldots,k-1$. To each integral $\int_0^1 g_{i,p}^j(\theta)d\theta$, we apply the mid-point rule with N knots u_α, i.e., $1/N\sum_{\alpha=0}^{N-1} g_{i,p}^j(u_\alpha)$. Next, by using optimal randomized or quantum algorithms with repetitions, ([2, 7]), we compute the mean

$$\frac{1}{mlN}\sum_{p=0}^{ml-1}\sum_{\alpha=0}^{N-1} g_{i,p}^j(u_\alpha) \tag{31}$$

with cost equal to $N^2\log(1/\delta_1)$ in the randomized or $N\log(1/\delta_1)$ in quantum setting, where δ_1 is specified below. It follows from [7] and [2] that there exists a constant C_1 (different in each setting) that gives us the approximation

$$\max_{j=0\ldots k-1}\left|\mathrm{AP}_i^j(g)-\frac{1}{mlN}\sum_{p=0}^{ml-1}\sum_{\alpha=0}^{N-1} g_{i,p}^j(u_\alpha)\right| \le C_1\frac{1}{N}\left(\max_{i,j,p}\|g_{i,p}^j\|_{[0,1]}\right)^\beta, \tag{32}$$

with probability at least $1-\delta_1$. At the right side of the inequality, the maximum is taken over $i=0,\ldots,n-1$, $j=0,\ldots,k-1$ and $p=0,\ldots,ml-1$. The exponent β may be set to 1 in the randomized setting, and to 3 in the quantum one. Parameter N is chosen as $N=n^{2^{s-1}-1}$ in the randomized setting or $N=n^{s-1}$ in the quantum setting. The parameter δ_1 depends on index K and basic parameter n of the final algorithm A^K, and is independent of s. For a given failure parameter δ, we set $\delta_1=1-(1-\delta)^{1/n^{2^K-1}}$ in the randomized setting, and $\delta_1=1-(1-\delta)^{1/n^K}$ in the quantum setting.

5. UPPER BOUNDS ON ERRORS OF THE ALGORITHMS

Upper bounds on the error of randomized and quantum algorithms defined in the previous section are presented in the following theorem.

Theorem 5.1. *Let K be the index of the final algorithm. Let $\alpha_K=r(2^K-1)+2^{K-1}-1$, $\beta_K=2^K-1$ in the randomized setting and $\alpha_K=rK+K-1$, $\beta_K=K$ in the quantum setting. Let $M=M(n)$ be the cost of the final algorithm $A^K([a,b],n,Y)$ (randomized or quantum) for solving problem* (1)*. Then*

$$e^{\mathrm{rand}}_{M(n)}\left(A^K,\mathcal{G}^r\right)=O\left(n^{-\alpha_K}\right),\quad \textit{and}\quad M(n)=O\left(n^{\beta_K}\log n\right). \tag{33}$$

For a given failure probability $\delta\in(0,1/2)$

$$e^{\mathrm{quant}}_{M(n)}\left(A^K,\mathcal{G}^r,\delta\right)=O\left(n^{-\alpha_K}\right),\quad \textit{and}\quad M(n)=O\left(n^{\beta_K}\left(\log n+\log\frac{1}{\delta}\right)\right). \tag{34}$$

The constants in big-O notations depend on the class parameters, k, a, b, K and initial values $u_a^1,\ldots,u_a^{k-1}$.

Before proving Theorem 5.1, we show some auxiliary results. We start with the following lemma showing bounds on the approximation function and initial values obtained in the algorithm at each step. We use the following notation. For the algorithm $A^s([c,d],n,Y)$ with initial values $Y=[y^0,y^1,\ldots,y^{k-1}]$, by z we mean the maximum $\max\limits_{j=1,\ldots,k-1}|y^j|$ for $k>1$ and 0 otherwise. Similarly, by z_i we mean $\max\limits_{j=1,\ldots,k-1}|y_i^j|$ for $k>1$ and 0 for $k=1$, where $Y_i=[y_i^0,y_i^1,\ldots,y_i^{k-1}]$ are initial conditions constructed in the algorithm A^s. With this notation, $z_0=z$.

Lemma 5.2. *Let $[c,d]\subset[a,b]$. Let $\hat{l}^s$ be an approximation given by $A^s([c,d],n,Y)$. There exist: an increasing sequence of positive constants $\{F^s\}$ and a constant B depending exclusively on class parameters, k, a and b, such that if $Q=ze^{d-c}+rF^s\left(e^{d-c}-1\right)$ and n satisfies $(d-c)Q/n<1$, then for*

$$Q_i=\begin{cases} z_i & s=1,\\ z_ie^h+rF^{s-1}\left(e^h-1\right) & s>1,\end{cases} \tag{35}$$

there is

$$Q_i \le Q, \ \textit{for} \quad i = 0, 1, \ldots, n-1. \tag{36}$$

Moreover,

$$\|\hat{l}^{s(j)}(\cdot)\|_{[c,d]} \le \begin{cases} B(1+Q) & j = 1, \ldots, k-1, \\ B(1+Q+\ldots+Q^{j-k}) & j = k, \ldots, r+k-1. \end{cases} \tag{37}$$

The proof of this lemma is outlined in the Appendix.

Using Lemma 5.2, we now prove the following fact concerning $g_{i,p}^j$. We show that each function $g_{i,p}^j$ is bounded and satisfies the Lipschitz condition with a constant independent of partition points and the length of interval.

Fact 5.3. *Consider algorithm $A^s([c,d], n, Y)$. Let the assumptions of Lemma 5.2 be satisfied. Then there exists a positive constant E depending exclusively on the parameters of the class $\mathcal{G}^r$, a and b (and independent of i, p, j, $x_{i,p}$, y_i^j) such that*

$$|g_{i,p}^j(\theta)| \le E(1+Q+\ldots+Q^r), \qquad \theta \in [0,1], \tag{38}$$

$$|g_{i,p}^j(\theta_1) - g_{i,p}^j(\theta_2)| \le 2E(1+Q+\ldots+Q^r)|\theta_1 - \theta_2|, \qquad \theta_1, \theta_2 \in [0,1]. \tag{39}$$

Proof. The proof is a consequence of the definition of $g_{i,p}^j(\theta)$, properties of class $\mathcal{G}^r$ and solutions $\hat{l}_i^{s-1}$. Since $w_{i,p}(t)$ is a Taylor approximation of function $g\left(t, \ldots, \hat{l}_i^{s-1,(q)}(t)\right)$, then

$$\sup_{\theta\in[0,1]} \left|g_{i,p}^j(\theta)\right| \le \sup_{\theta\in[0,1]} \left| \frac{\left(h-(p+\theta)\bar{h}\right)^{k-j-1}(\theta\bar{h})^r}{h^{k-j-1}\bar{h}^r(k-1-j)!} \right| \cdot \sup_{t\in[x_{i,p},x_{i,p+1}]} \left| g^{(r)}\left(t, \ldots, \hat{l}_i^{s-1,(q)}(t)\right)\right|.$$

The rth derivative of the function $g\left(t, \ldots, \hat{l}_i^{s-1,(q)}(t)\right)$ with respect to t is a derivative of a compound function, where the external derivatives are the partial derivatives of $g \in \mathcal{G}^r$ (and are bounded by D_j) and the internal derivatives are the derivatives of $\hat{l}_i^{s-1}$. From Lemma 5.2, (36) holds. This guarantees that the assumptions $hQ_i/m < 1$ for algorithms $A^{s-1}([x_i, x_{i+1}], m, Y_i)$ are fulfilled for any m, so there exists a constant B such that for $i = 0, \ldots, n-1$, the derivatives of functions $\hat{l}_i^{s-1}$ are bounded by $B(1+Q_i)$ or by $B(1+\ldots+Q_i^{j-k})$ (see 37). Because $Q_i \le Q$, we may bound $\sup_{t\in[x_{i,p},x_{i,p+1}]} \left|g^{(r)}\left(t, \ldots, \hat{l}_i^{s-1,(q)}(t)\right)\right|$ by $E(1+\ldots+Q^r)$, where E is a constant depending exclusively on class parameters, a, b and k. Since $\sup_{\theta\in[0,1]} \left| \frac{\left(h-(p+\theta)\bar{h}\right)^{k-j-1}(\theta\bar{h})^r}{h^{k-j-1}\bar{h}^r(k-1-j)!} \right| \le 1$, then the statement of (38) holds true.

The proof of (39) is similar. We show the bound on the first derivative of $g_{i,p}^j$ and use the Lagrange theorem. □

The following theorem shows the bounds on the error, cost and on the probability of success of the algorithm from the previous section. We use the following notation:

— $\mathrm{cost}(A^s, n)$ – the total cost of recursive algorithm A^s with a basic parameter n (this cost is equal to the number of evaluations of the right-hand side function);
— $\mathrm{Prob}(A^s, n)$ – the probability of success of algorithm A^s for a basic parameter n;
— in the randomized setting:

$$C(Q) = \begin{cases} \dfrac{Q^{r+1}-1}{Q-1} & \text{if} \quad Q \neq 1, \\ r+1 & \text{if} \quad Q = 1, \end{cases} \tag{40}$$

$$\alpha_s = r(2^s - 1) + 2^{s-1} - 1, \tag{41}$$

$$\beta_s = 2^s - 1, \tag{42}$$

$$\psi_s(n) = \sum_{i=1}^{s-1} n^{2^i - 1} \text{ for } s > 1; \quad \psi_1(n) = 0, \tag{43}$$

— in the quantum setting:

$$C(Q) = \begin{cases} \left(\dfrac{Q^{r+1}-1}{Q-1}\right)^3 & \text{if} \quad Q \neq 1, \\ (r+1)^3 & \text{if} \quad Q = 1, \end{cases} \tag{44}$$

$$\alpha_s = rs + s - 1, \tag{45}$$

$$\beta_s = s, \tag{46}$$

$$\psi_s(n) = \frac{n^s - n}{n-1} \text{ for } s > 1; \quad \psi_1(n) = 0. \tag{47}$$

The following result gives error and cost bounds for algorithm A^s.

Theorem 5.4. *Let $[c,d] \subset [a,b]$. Let $\hat{l}^s$ be an approximation given by $A^s([c,d], n, Y)$ and let $\delta_1 \in (0, 1/2)$. Let Q be given as in Lemma 5.2. There exist positive constants C_1^s, C_2^s depending on the class parameters and a, b, k, s, such that if n satisfies $(d-c)Q/n < 1$, then*

$$\sum_{j=0}^{q} \sup_{x \in [c,d]} \left| u^{(j)}(x) - \hat{l}^{s\,(j)}(x) \right| \leq C_1^s \, C(Q) \, (d-c)^{r+1} n^{-\alpha_s}, \tag{48}$$

with probability at least

$$\mathrm{Prob}(A^s, n) = (1 - \delta_1)^{\psi_s(n)}. \tag{49}$$

The total cost is bounded by

$$\mathrm{cost}(A^s, n) \leq C_2^s n^{\beta_s} \log 1/\delta_1. \tag{50}$$

(*In* (48)*, by the derivative of $\hat{l}^s$ at its discontinuity points we mean its right-hand side derivative.*)

Proof. We shall use l_1 vector and matrix norms. We prove the statement by induction with respect to s.

Let $s = 1$. We shall need bounds on $\sum_{j=0}^{q} \sup_{x\in[c,d]} |u^{(j)}(x) - \hat{l}^{1\,(j)}(x)|$, where $\hat{l}^1$ is a function obtained by the Taylor algorithm approximating the solution $u(x)$ on $[c, d]$. Simple modifications of the proof given in [9] do not give us a satisfactory bound. Hence, we give a new proof here.

Let

$$E_i^j = \sup_{x\in[x_i,x_{i+1}]} \left| u^{(j)}(x) - \bar{u}_i^{(j)}(x) \right|, \tag{51}$$

$$e_i^j = u^{(j)}(x_i) - \bar{u}_i^{(j)}(x_i) = u^{(j)}(x_i) - y_i^j, \tag{52}$$

for $j = 0, 1, \ldots, k-1$. As in expression (25) for $\bar{u}_i^{(j)}(x)$, the jth derivative of the solution $u(x)$ satisfies the equation

$$u^{(j)}(x) = \sum_{p=0}^{k-1-j} u^{(j+p)}(x_i) \frac{(x-x_i)^p}{p!} + \\ + \frac{1}{(k-1-j)!} \int_{x_i}^{x} (x-t)^{k-1-j}\, g\left(t, u(t), u'(t), \ldots, u^{(q)}(t)\right) dt$$

in $[x_i, x_{i+1}]$. By subtracting $\bar{u}_i^{(j)}(x)$ from $u^{(j)}(x)$, we obtain

$$u^{(j)}(x) - \bar{u}_i^{(j)}(x) = \sum_{p=0}^{k-1-j} \left(u^{(j+p)}(x_i) - y_i^{j+p} \right) \frac{(x-x_i)^p}{p!} + \\ + \frac{1}{(k-1-j)!} \int_{x_i}^{x} (x-t)^{k-1-j} \left[g\left(t, u(t), \ldots, u^{(q)}(t)\right) - g\left(t, \bar{u}_i(t), \ldots, \bar{u}_i^{(q)}(t)\right) \right] dt. \tag{53}$$

The function g satisfies the Lipschitz condition with a constant D_1. Using this, and passing in (53) to supremum over $x \in [x_i, x_{i+1}]$, we get the bounds

$$E_i^j \le \sum_{p=0}^{k-1-j} \left| e_i^{j+p} \right| \frac{h^p}{p!} + D_1 \frac{h^{k-j}}{(k-j)!} \sum_{p=0}^{q} E_i^p, \qquad j = 0, 1, \ldots, k-1. \tag{54}$$

After summing up the expressions above for $j = 0, \ldots, q$ and after some simple calculation we get (for details see the proof of Theorem 5.1 in [9])

$$\sum_{j=0}^{q} E_i^j \le \frac{\sum\limits_{p=0}^{q} |e_i^p| \sum\limits_{j=0}^{p} \frac{h^j}{j!} + \sum\limits_{p=q+1}^{k-1} |e_i^p| \sum\limits_{j=p-q}^{p} \frac{h^j}{j!}}{1 - D_1 \sum\limits_{p=0}^{q} \frac{h^{k-p}}{(k-p)!}}. \tag{55}$$

Then for sufficiently small h, we arrive at the following bound

$$\sum_{j=0}^{q} E_i^j \leq 2 \sum_{p=0}^{k-1} |e_i^p|. \tag{56}$$

We now derive a bound on $\sum_{p=0}^{k-1} |e_i^p|$. According to the definition of the algorithm, the initial conditions y_i^j of $\bar{u}_i(x)$ satisfy $y_{i+1}^j = \hat{l}_i^0(x_{i+1})$ for $i = 0, \ldots, n-1$, and $y_0^j := y^j$, where y^j are initial conditions of the problem. We split e_{i+1}^j into two parts

$$e_{i+1}^j = u^{(j)}(x_{i+1}) - y_{i+1}^j = u^{(j)}(x_{i+1}) - \bar{u}_i^{(j)}(x_{i+1}) + \bar{u}_i^{(j)}(x_{i+1}) - \hat{l}_i^{0\,(j)}(x_{i+1}). \tag{57}$$

The polynomial $\hat{l}_i^0$ is a Taylor approximation of $\bar{u}_i$. It may be shown that there exists a constant $C_T > 0$ depending exclusively on class parameters, k, such that for $j = 0, \ldots, k-1$

$$\left| \bar{u}_i^{(j)}(x_{i+1}) - \hat{l}_i^{0\,(j)}(x_{i+1}) \right| \leq C_T(1 + Q_i + \ldots + Q_i^r) h^{r+k-j}. \tag{58}$$

One can get this by bounding the $(r+k)$th derivative of function $\bar{u}_i$ (i.e., the rth derivative of compound function $g(\cdot, \bar{u}_i(\cdot), \ldots, \bar{u}_i^{(q)}(\cdot))$). From Lemma 5.2 there follows that $Q_i \leq Q$. Hence, using the definition of $C(Q)$, we get

$$\left| \bar{u}_i^{(j)}(x_{i+1}) - \hat{l}_i^{0\,(j)}(x_{i+1}) \right| \leq C_T\, C(Q)\, h^{r+k-j}. \tag{59}$$

Taking $x = x_{i+1}$ in (53), then using (54) and (56), we get

$$\left| u^{(j)}(x_{i+1}) - \bar{u}_i^{(j)}(x_{i+1}) \right| \leq \sum_{p=0}^{k-1-j} |e_i^{j+p}| \frac{h^p}{p!} + 2D_1 \frac{h^{k-j}}{(k-j)!} \sum_{p=0}^{k-1} |e_i^p|. \tag{60}$$

Using (57), the triangle inequality and the above approximations, we obtain

$$|e_{i+1}^j| \leq \sum_{p=0}^{k-1-j} |e_i^{j+p}| \frac{h^p}{p!} + 2D_1 \frac{h^{k-j}}{(k-j)!} \sum_{p=0}^{k-1} |e_i^p| + C_T\, C(Q)\, h^{r+k-j}, \tag{61}$$

so that

$$|e_{i+1}^j| \leq |e_i^j| + (2D_1 + 1)h \sum_{p=0}^{k-1} |e_i^p| + C_T\, C(Q)\, h^{r+k-j}. \tag{62}$$

It is easy to see that

$$\left| e_i^j \right| \leq P_i^j, \qquad j = 0, 1, \ldots, k-1, \tag{63}$$

where the numbers P_i^j satisfy the following system of difference equations

$$P_{i+1}^j = P_i^j + (2D_1 + 1)h \sum_{p=0}^{k-1} P_i^p + C_T\, C(Q)\, h^{r+k-j}, \tag{64}$$

with $P_0^j = 0$ (since $\left|u^{(j)}(c) - y_0^j\right| = 0$). System (64) may be written in a matrix form as follows. Let $P_i = [P_i^0, \ldots, P_i^{k-1}]^T$, $B = C_T\, C(Q)\, h^{r+1}[h^{k-1}, \ldots, 1]^T$ and $A = I + (2D_1 + 1)hM$, where I is the identity matrix, and all elements of matrix M are equal to 1. Then

$$P_{i+1} = AP_i + B = \left(I + A + \cdots + A^i\right) B, \tag{65}$$

where $P_0 = [0, \ldots, 0]^T$. Hence,

$$\|P_{i+1}\| \le \left(1 + \|A\| + \cdots + \|A\|^i\right) \|B\|, \tag{66}$$

where for sufficiently small h, $\|B\| \le 2C_T\, C(Q)\, h^{r+1}$. The matrices A and M are of size $k \times k$, so $\|M\| = k$, and hence

$$\|A\| = \|I + (2D_1 + 1)hM\| \le 1 + (2D_1 + 1)hk. \tag{67}$$

From this, for $i = 0, \ldots, n-1$,

$$\begin{aligned} 1 + \|A\| + \cdots + \|A\|^i &\le \sum_{p=0}^{i} (1 + (2D_1 + 1)hk)^p = \\ &= \frac{(1 + (2D_1 + 1)\, hk)^{i+1} - 1}{(2D_1 + 1)hk} \le \\ &\le \frac{C_A(d-c)}{h}, \end{aligned} \tag{68}$$

where the constant C_A depends on the class parameters and interval $[a, b]$, and is independent of n, i, c, d. Thus,

$$\|P_{i+1}\| \le 2C_T C_A\, C(Q)\, (d-c)h^r. \tag{69}$$

Using the definition of P_i and the obtained bound, from (56) we get

$$\sum_{j=0}^{q} E_i^j \le 2\sum_{p=0}^{k-1} |e_i^p| \le 2\|P_i\| \le 4C_T C_A\, C(Q)\, (d-c)h^r. \tag{70}$$

From the properties of Taylor's expansion of the function $\bar{u}_i$ (see (59)), we get

$$\sum_{j=0}^{q} \sup_{x \in [x_i, x_{i+1}]} \left|\bar{u}_i^{(j)}(x) - \hat{l}_i^{0\,(j)}(x)\right| \le 2C_T\, C(Q)\, h^{r+k-q}. \tag{71}$$

This bound, valid for a sufficiently small h, is independent of i.

We are now ready to prove the bound on $\sum_{j=0}^{q} \sup_{x \in [c,d]} |u^{(j)}(x) - \hat{l}^{1\,(j)}(x)|$, where $\hat{l}^1(x) := \hat{l}_i^0(x)$ for $x \in [x_i, x_{i+1}]$. From the triangle inequality, the bound on $\sum_{j=0}^{q} E_i^j$ and (71), we get (for all i and small h)

$$\begin{aligned} \sum_{j=0}^{q} \sup_{x \in [x_i, x_{i+1}]} \left|u^{(j)}(x) - \hat{l}^{1\,(j)}(x)\right| &\le \sum_{j=0}^{q} E_i^j + \sum_{j=0}^{q} \sup_{x \in [x_i, x_{i+1}]} \left|\bar{u}_i^{(j)}(x) - \hat{l}_i^{0\,(j)}(x)\right| \le \\ &\le 4C_T C_A\, C(Q)\, (d-c)h^r + 2C_T\, C(Q)\, h^{r+k-q} \le \\ &\le C_T\, C(Q)\, (4C_A + 1)(d-c)^{r+1} n^{-r}. \end{aligned}$$

Hence, the algorithm $A^1([c,d],n,Y)$ (both the randomized and quantum one) satisfies

$$\sum_{j=0}^{q} \sup_{x\in[c,d]} \left| u^{(j)}(x) - \hat{l}^{1(j)}(x) \right| \leq C_1^1 \, C(Q) \, (d-c)^{r+1} n^{-r}, \tag{72}$$

where $C_1^1 := (q+1)C_T(4C_A+1)$. This holds with certainty ($\mathrm{Prob}(A^1,n)=1$). The constant C_1^1 depends on class parameters, a, b, k (and $q<k$). The total cost is

$$\mathrm{cost}(A^1,n) = C_2^1 n, \tag{73}$$

where the constant $C_2^1 > 0$ depends on r and k only. This ends the proof for the case of $s=1$.

Let us now assume that the statement of the theorem holds for $s-1$. We recall that $x_i = c+ih$ for $h=(d-c)/n$ define a uniform partition of interval $[c,d]$, $\hat{l}^s$ denotes the approximation of the solution on $[c,d]$, which we get by the algorithm $A^s([c,d],n,Y)$, and $\hat{l}_i^{s-1}(x)$ is the approximation of the solution on $[x_i,x_{i+1}]$ obtained by the algorithm $A^{s-1}([x_i,x_{i+1}],m,Y_i)$ with $\mathrm{cost}(A^{s-1},m)$ and the probability of success $\mathrm{Prob}(A^{s-1},m)$. The numbers m, l and N are given in the definition of the algorithm. Let E_i^j and e_i^j be defined by (51) and (52). Similarly as in the case of $s=1$, we show that (56) holds. We now derive a bound on $\sum_{j=0}^{k-1}|e_i^j|$. By similar arguments as used to show (29), we can write

$$\begin{aligned} u^{(j)}(x_{i+1}) = &\sum_{p=0}^{k-1-j} u^{(j+p)}(x_i)\frac{h^p}{p!} + \frac{1}{(k-1-j)!}\sum_{p=0}^{ml-1}\int_{x_{i,p}}^{x_{i,p+1}} (x_{i+1}-t)^{k-j-1} w_{i,p}(t)dt + \\ &+ \bar{h}^{r+1}h^{k-j-1}ml\frac{1}{ml}\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta + \\ &+ \frac{1}{(k-1-j)!}\int_{x_i}^{x_{i+1}} (x_{i+1}-t)^{k-j-1}\Big[g\Big(t,u(t),\ldots,u^{(q)}(t)\Big) - \\ &\qquad - g\left(t,\hat{l}_i^{s-1}(t),\ldots,\hat{l}_i^{s-1,(q)}(t)\right)\Big]dt, \end{aligned} \tag{74}$$

where $j=0,\ldots,k-1$. After subtracting expression (30) for the initial point y_{i+1}^j and remembering that $h=\bar{h}ml$, we get

$$\begin{aligned} u^{(j)}(x_{i+1}) - y_{i+1}^j = &\sum_{p=0}^{k-1-j}\left(u^{(j+p)}(x_i) - y_i^{j+p}\right)\frac{h^p}{p!} + \\ &+ \frac{1}{(k-1-j)!}\int_{x_i}^{x_{i+1}} (x_{i+1}-t)^{k-j-1}\Big[g\Big(t,u(t),\ldots,u^{(q)}(t)\Big) - \\ &- g\left(t,\bar{u}_i(t),\ldots,\bar{u}_i^{(q)}(t)\right) + \\ &+ g\left(t,\bar{u}_i(t),\ldots,\bar{u}_i^{(q)}(t)\right) - g\left(t,\hat{l}_i^{s-1}(t),\ldots,\hat{l}_i^{s-1,(q)}(t)\right)\Big]dt + \\ &+ \bar{h}^r h^{k-j}\left(\frac{1}{ml}\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta - \mathrm{AP}_i^j(g)\right). \end{aligned} \tag{75}$$

The numbers $\mathrm{AP}_i^j(g)$ approximate the mean value of integrals $\frac{1}{ml}\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta$. The error of this approximation is a sum of two components

$$\begin{aligned}\Big|\frac{1}{ml}\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta - \mathrm{AP}_i^j(g)\Big| \le \Big|\frac{1}{ml}\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta - \frac{1}{mlN}\sum_{p=0}^{ml-1}\sum_{\alpha=0}^{N-1} g_{i,p}^j(u_\alpha)\Big| + \\ + \Big|\frac{1}{mlN}\sum_{p=0}^{ml-1}\sum_{\alpha=0}^{N-1} g_{i,p}^j(u_\alpha) - \mathrm{AP}_i^j(g)\Big|.\end{aligned} \tag{76}$$

Bounding the error of the N point mid-point rule and using Fact 5.3 we get

$$\left|\frac{1}{ml}\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta - \frac{1}{mlN}\sum_{p=0}^{ml-1}\sum_{\alpha=0}^{N-1} g_{i,p}^j(u_\alpha)\right| \le \frac{2E}{N}C(Q). \tag{77}$$

Using (32), we get the following bound:

$$\left|\frac{1}{ml}\sum_{p=0}^{ml-1}\int_0^1 g_{i,p}^j(\theta)d\theta - \mathrm{AP}_i^j(g)\right| \le \frac{C_c}{N}C(Q), \tag{78}$$

for some constant C_c dependent on class parameters, a, b, k and independent of initial values.

Coming back to (75), to bound the first integral in (75) we use the Lipschitz condition for g and (56). Applying bound (78), we next arrive at

$$\begin{aligned}|e_{i+1}^j| \le \sum_{p=0}^{k-1-j}|e_i^{j+p}|\frac{h^p}{p!} + 2D_1\frac{h^{k-j}}{(k-j)!}\sum_{p=0}^{k-1}|e_i^p| + \\ + D_1\frac{h^{k-j}}{(k-j)!}\sum_{j=0}^{k-1}\sup_{x\in[x_i,x_{i+1}]}\left|\bar{u}_i^{(j)}(x) - \hat{l}_i^{s-1,(j)}(x)\right| + \bar{h}^r h^{k-j}\frac{C_c}{N}C(Q).\end{aligned} \tag{79}$$

From (79) and the inductive assumption for $\sum_{p=0}^{k-1}\sup_{x\in[x_i,x_{i+1}]}\left|\bar{u}_i^{(j)}(x) - \hat{l}_i^{s-1,(j)}(x)\right|$, for $i = 0,\ldots,n-1$ and $j = 0,\ldots,k-1$, we get

$$\begin{aligned}|e_{i+1}^j| \le |e_i^j| + (2D_1+1)h\sum_{p=0}^{k-1}|e_i^p| + \\ + D_1\frac{h^{k-j}}{(k-j)!}C_1^{s-1}C(Q_i)h^{r+1}m^{-\alpha_{s-1}} + \bar{h}^r h^{k-j}\frac{C_c\,C(Q)}{N}.\end{aligned} \tag{80}$$

Since the assumptions of the Lemma 5.2 hold, there is $Q_i \le Q$, and hence $C(Q_i) \le C(Q)$. As in the case of $s = 1$, we define the system of difference equations with the same matrix A

$$P_{i+1} = AP_i + B, \tag{81}$$

where $B = [b^0, \ldots, b^{k-1}]^T$ with $b^j = D_1 h^{k-j}/(k-j)! C_1^{s-1} C(Q)\, h^{r+1} m^{-\alpha_{s-1}} + \bar{h}^r h^{k-j} C_c\, C(Q)\, /N$ and $P_i = [P_i^0, \ldots, P_i^{k-1}]^T$. Then $|e_i^j| \leq P_i^j$.

Let us now bound the norm of vector B. We start with the randomized setting. We remind that $\bar{h} = h/(ml)$ and $m = n^2$, $l = n^{2^s-4}$, $N = n^{2^{s-1}-1}$, $\alpha_s = r(2^s-1) + 2^{s-1} - 1$. Thus, there is

$$\begin{aligned} b^j &= D_1 \frac{h^{k-j}}{(k-j)!} C_1^{s-1}\, C(Q)\, h^{r+1} m^{-\alpha_{s-1}} + \bar{h}^r h^{k-j} \frac{C_c}{N} C(Q) = \\ &= C(Q)\, h^{k-j} (d-c)^r \Big[D_1 C_1^{s-1} \frac{d-c}{(k-j)!} n^{-r-1} n^{-2\left(r(2^{s-1}-1)+2^{s-2}-1\right)} + \\ &\qquad + C_c n^{-r} \left(n^{-2} n^{-2^s+4} \right)^r n^{-2^{s-1}+1} \Big] = \\ &= C(Q)\, h^{k-j} (d-c)^r \left(D_1 C_1^{s-1} \frac{d-c}{(k-j)!} + C_c \right) n^{-r(2^s-1)-2^{s-1}+1}. \end{aligned}$$

Hence, for a sufficiently small h

$$\|B\| \leq 2h(d-c)^r\, C(Q)\, \left(D_1 C_1^{s-1}(d-c) + C_c \right) n^{-\alpha_s}. \tag{82}$$

The same bound we get in the quantum setting, with $m = n$, $l = n^{s-2}$, $N = n^{s-1}$ and $\alpha_s = rs + s - 1$.

In both settings, $P_0 = [0, \ldots, 0]^T$, and $\|A\| \leq 1 + (2D_1 + 1)hk$. Therefore, using (68) we get

$$\begin{aligned} \|P_{i+1}\| &\leq \left(1 + \|A\| + \cdots + \|A\|^i \right) \|B\| \leq \\ &\leq C_A \frac{d-c}{h} \|B\| \leq \\ &\leq 2C_A\, C(Q)\, (d-c)^{r+1} \left(D_1 C_1^{s-1}(d-c) + C_c \right) n^{-\alpha_s}. \end{aligned}$$

From the inductive assumption and the bound on $\|P_i\|$ above, for $i = 0, \ldots, n-1$, we finally get

$$\begin{aligned} \sum_{j=0}^{q} \sup_{x \in [x_i, x_{i+1}]} \left| u^{(j)}(x) - \hat{l}^{s(j)}(x) \right| &\leq \sum_{j=0}^{q} E_i^j + \sum_{j=0}^{q} \sup_{x \in [x_i, x_{i+1}]} \left| \bar{u}_i^{(j)}(x) - \hat{l}_i^{s-1,(j)}(x) \right| \leq \\ &\leq 2\|P_i\| + C_1^{s-1}\, C(Q)\, h^{r+1} m^{-\alpha_{s-1}} \leq \\ &\leq \tilde{C}_1^s\, C(Q)\, (d-c)^{r+1} n^{-\alpha_s}, \end{aligned}$$

where $\tilde{C}_1^s := 4C_A \left(D_1 C_1^{s-1}(b-a) + C_c \right) + C_1^{s-1}$.

Hence

$$\sum_{j=0}^{q} \sup_{x \in [c,d]} \left| u^{(j)}(x) - \hat{l}^{s(j)}(x) \right| \quad \leq \quad C_1^s\, C(Q)\, (d-c)^{r+1} n^{-\alpha_s}, \tag{83}$$

for $C_1^s := (q+1)\tilde{C}_1^s$.

Let us now compute the total cost of the algorithm A^s. The cost consists of:

— the cost of computing $w_{i,p}$, which is equal to $C_w nml$, where C_w is a positive constant depending on r and k only;
— the cost of computing $\hat{l}_i^{s-1}$, which is equal to $\mathrm{cost}(A^{s-1},m)\cdot n \leq C_2^{s-1} n m^{\beta_{s-1}}(\log 1/\delta_1)$; and
— the cost of computing $\mathrm{AP}_i(g)$, which is equal to $N^\kappa \log 1/\delta_1$, where $\kappa = 2$ in the randomized setting, and $\kappa = 1$ in the quantum setting.

The parameters m, l and N are chosen so that the exponents of n in the above three components are the same and are equal to $\beta_s = 2^s - 1$ in the randomized setting or $\beta_s = s$ in the quantum setting. The total cost is then bounded by

$$\mathrm{cost}(A^s,n) \leq C_2^s n^{\beta_s} \log 1/\delta_1 \tag{84}$$

for

$$C_2^s = C_w + C_2^{s-1} + 1. \tag{85}$$

Bound (83) holds true with the probability of success at least

$$\mathrm{Prob}(A^s,n) = (1-\delta_1)^n \mathrm{Prob}(A^{s-1},m)^n. \tag{86}$$

Simple calculations yield that $\mathrm{Prob}(A^s,n) = (1-\delta_1)^{\psi_s(n)}$ for $\psi_s(n)$ given by (43) and (47) in the randomized and the quantum settings, respectively. This ends the proof of the theorem. □

We now give the proof of Theorem 5.1.

Proof of Theorem 5.1. Consider the randomized setting. Let $A^K([a,b],n,Y)$ be the final algorithm. From Theorem 5.4, for $Q = ze^{b-a} + rF^K(e^{b-a}-1)$, with $z = \max\{|u_a'|,\ldots,|u_a^{k-1}|\}$ for $k>1$ and $z=0$ for $k=1$, we get

$$\sup_{x\in[a,b]} \left|u(x) - \hat{l}^K(x)\right| \leq \sum_{j=0}^{q} \sup_{x\in[a,b]} \left|u^{(j)}(x) - \hat{l}^{K\,(j)}(x)\right| \leq C_1^K\, C(Q)\,(b-a)^{r+1} n^{-\alpha_K}, \tag{87}$$

with probability at least $(1-\delta_1)^{\psi_K(n)}$. Let $\delta \in (0,1/2)$. From the inequality $\psi_s(n) \leq n^{2^s-1}$ it follows that to get probability of success at least $1-\delta$ it is sufficient to take δ_1 equal to $1-(1-\delta)^{1/n^{2^K-1}}$. (For A^s with $s<K$, the parameter δ_1 is fixed.)

We now pass from the probabilistic error to the error defined by (7). Since the result of the randomized algorithm A^K is random, the error $e^{\omega}_{M(n)}(A^K,g) = \sup_{x\in[a,b]} |u(x) - \hat{l}^{\omega\,K}(x)|$ is a random variable (taking a finite number of values). This error is bounded by $L\,C(Q) n^{-r(2^K-1)}$, for some deterministic positive constant L depending on the parameter of class $\mathcal{G}^r$ and interval $[a,b]$ only. To see that such a constant exists, note that the random variable $\mathrm{AP}_i^j(g)$ satisfies $\|\mathrm{AP}_i^j(g)\| \leq \max\limits_{i,j,p} \|g_{i,p}^j\|_{[0,1]}$.

Hence, instead of (32) we can proceed with the bound

$$\left| \frac{1}{ml} \sum_{p=0}^{ml-1} \int_0^1 g_{i,p}^j(\theta) d\theta - \mathrm{AP}_i^j(g) \right| \leq 3E\, C(Q), \tag{88}$$

which holds with certainty.

The error of A^K can be bounded by

$$\left(e_{M(n)}^{\mathrm{rand}}(A^K, \mathcal{G}^r)\right)^2 \leq L^2 C^2(Q) n^{-2r(2^K-1)} \delta + (C_1^K\, C(Q)\, (b-a)^{r+1} n^{-\alpha_K})^2. \tag{89}$$

Indeed, for every function $g \in \mathcal{G}^r$ and $\lambda = C_1^K\, C(Q)\, (b-a)^{r+1} n^{-\alpha_K}$, there holds

$$\begin{aligned} \mathbf{E}\left(e_{M(n)}^{\omega}(A^K, g)\right)^2 = &\int_{e_{M(n)}^{\omega}(A^K,g)>\lambda} \left(e_{M(n)}^{\omega}(A^K, g)\right)^2 d\mathbf{P}(\omega) + \\ &+ \int_{e_{M(n)}^{\omega}(A^K,g)\leq\lambda} \left(e_{M(n)}^{\omega}(A^K, g)\right)^2 d\mathbf{P}(\omega). \end{aligned} \tag{90}$$

The estimate $e_{M(n)}^{\omega}(A^K, g) \leq \lambda$ holds true with probability at least $1-\delta$. For $e_{M(n)}^{\omega}(A^K, g) > \lambda$, we use the fact that the error in the supremum norm is bounded.

Hence,

$$\mathbf{E}\left(e_{M(n)}^{\omega}(A^K, g)\right)^2 \leq \left(L\, C(Q) n^{-r(2^K-1)}\right)^2 \delta + \lambda^2. \tag{91}$$

Taking $\delta = \left(C_1^K (b-a)^{r+1} n^{-2^{K-1}+1} / L\right)^2$, we get

$$e_{M(n)}^{\mathrm{rand}}(A^K, \mathcal{G}^r) = O\left(n^{-\alpha_K}\right). \tag{92}$$

The constant in the big-O notation depends on class parameters, a, b, k, K and initial values $u_a', \ldots, u_a^{k-1}$.

This error is achieved with the cost

$$M(n) = \mathrm{cost}(A^K, n) = O\left(n^{\beta_K} \log \frac{1}{\delta_1}\right). \tag{93}$$

Since

$$\log\left(\frac{1}{\delta_1}\right) = O\left(\log n + \log \frac{1}{\delta}\right) = O\left(\log n\right), \tag{94}$$

then

$$M(n) = \mathrm{cost}(A^K, n) = O\left(n^{\beta_K} \log n\right), \tag{95}$$

where the constant is dependent on K. The proof in the randomized setting is completed.

In the quantum setting, we take $\delta_1 = 1 - (1-\delta)^{1/n^K}$. The bound

$$e_{M(n)}^{\mathrm{quant}}(A^K, \mathcal{G}^r, \delta) = O\left(n^{-\alpha_K}\right) \tag{96}$$

is then achieved with the cost

$$M(n) = \mathrm{cost}(A^K, n) = O\left(n^{\beta_K} \log \frac{1}{\delta_1}\right) = O\left(n^{\beta_K} (\log n + \log \frac{1}{\delta})\right). \tag{97}$$

This immediately follows from Theorem 5.4 and the definition of the quantum error. Here, the constant in the big-O notation also depends on class parameters, a, b, k, K and initial values $u'_a, \ldots, u_a^{k-1}$. □

We end this section with the theorem about the upper bounds on the complexity of problem (1), which is a consequence of results presented above.

Theorem 5.5. *For any $\gamma \in (0,1)$, there exist positive constants $C_1(\gamma)$, $\varepsilon_0(\gamma)$ (depending exclusively on γ, the parameters of the class $\mathcal{G}^r$, a, b and initial conditions at point a) such that for all $\varepsilon \in (0, \varepsilon_0(\gamma))$, the ε-complexity in the randomized and quantum settings satisfy*

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{G}^r, \varepsilon) \leq C_1(\gamma) \left(\frac{1}{\varepsilon}\right)^{1/(r+1/2-\gamma)} \tag{98}$$

and for $\delta \in (0, 1/2)$

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{G}^r, \varepsilon, \delta) \leq C_2(\gamma) \left(\frac{1}{\varepsilon}\right)^{1/(r+1-\gamma)} \log \frac{1}{\delta}. \tag{99}$$

Proof. We start with the randomized setting. For a fixed parameter γ, let K (the index of the final algorithm) be equal to $\lceil \log(1/\gamma + 1) \rceil$. Then

$$\frac{\beta_K}{\alpha_K} \leq 1/(r + 1/2 - \gamma/2). \tag{100}$$

From Theorem 5.1, the final algorithm has the cost

$$M(n) = \mathrm{cost}(A^K, n) = O\left(n^{\beta_K} \log n\right) \tag{101}$$

and the error

$$e^{\mathrm{rand}}_{M(n)}\left(A^K, \mathcal{G}^r\right) = O\left(n^{-\alpha_K}\right). \tag{102}$$

We now ask about the minimal cost $M(n)$ needed to achieve a given accuracy $\varepsilon > 0$, $e^{\mathrm{rand}}_{M(n)}\left(A^K, \mathcal{G}^r\right) \leq \varepsilon$. Comparing the bounds on error and cost, we get

$$\mathrm{cost}(A^K, \varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1/2-\gamma/2)} \log \frac{1}{\varepsilon}\right), \tag{103}$$

so that for a sufficiently small ε

$$\mathrm{cost}(A^K, \varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1/2-\gamma)}\right). \tag{104}$$

The constant in big-O notation depends on γ.

We now pass to the quantum setting. Let $K = \lceil 2/\gamma \rceil$. Then

$$\frac{\beta_K}{\alpha_K} \leq 1/(r+1-\gamma/2). \tag{105}$$

The inequality $e^{\mathrm{quant}}_{M(n)}\left(A^K, \mathcal{G}^r, \delta\right) \leq \varepsilon$ leads to the following bound on cost

$$\mathrm{cost}(A^K, \varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1-\gamma/2)} \left(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right). \tag{106}$$

For a small ε,

$$\mathrm{cost}(A^K, \varepsilon) = O\left(\left(\frac{1}{\varepsilon}\right)^{1/(r+1-\gamma)} \log\frac{1}{\delta}\right). \tag{107}$$

The constants in the big-O notation depend on γ. This completes the proof in the quantum setting. □

6. LOWER COMPLEXITY BOUNDS IN THE RANDOMIZED AND QUANTUM SETTINGS

In this section, we show lower bounds on randomized and quantum complexity of equations of order k with the right-hand side function belonging to class $\mathcal{G}^r$.

Theorem 6.1. *Let $r \geq 1$. For an arbitrary k, there exist positive constants C_1 and C_2 depending on the class $\mathcal{G}^r$ and k only, such that*

$$\mathrm{comp}^{\mathrm{rand}}(\mathcal{G}^r, \varepsilon) \geq C_1 \left(\frac{1}{\varepsilon}\right)^{1/(r+1/2)}, \tag{108}$$

and for all $\delta \in (0, 1/4)$

$$\mathrm{comp}^{\mathrm{quant}}(\mathcal{G}^r, \varepsilon, \delta) \geq \mathrm{comp}^{\mathrm{quant}}(\mathcal{G}^r, \varepsilon, 1/4) \geq C_2 \left(\frac{1}{\varepsilon}\right)^{1/(r+1)}. \tag{109}$$

Proof. We first prove the lower bound in the quantum setting. Consider the subclass $\mathcal{G}^r_1$ of $\mathcal{G}^r$ given by functions dependent on x only:

$$\mathcal{G}^r_1 = \{g : [a,b] \to \mathbb{R} \mid g \in C^{(r)}([a,b]) \sup_{x\in[a,b]} \left|g^{(j)}(x)\right| \leq D_j,\ j = 0, \ldots, r\}. \tag{110}$$

Let A be any quantum algorithm using M information values solving problem (1), such that $e^{\mathrm{quant}}_M(A, \mathcal{G}^r) \leq \varepsilon$ (this yields $e^{\mathrm{quant}}_M(A, \mathcal{G}^r_1) \leq \varepsilon$). We shall prove that $M = \mathrm{cost}^{\mathrm{quant}}(A) \geq (1/\varepsilon)^{1/(r+1)}$ by showing that the solution of problem (1) with a suitable right-hand side function leads to the solution of the integration problem.

Note that for any function $g \in \mathcal{G}_1^r$ problem (1) reduces to the computation of the k-fold integral

$$u(x) = \sum_{j=0}^{k-1} \frac{u_a^j}{j!}(x-a)^j + \int_a^x \int_a^{t_{k-1}} \dots \int_a^{t_1} g(t)\, dt\, dt_1 \dots dt_{k-1}, \tag{111}$$

or, equivalently, to the weighted integral

$$u(x) = \sum_{j=0}^{k-1} \frac{u_a^j}{j!}(x-a)^j + \frac{1}{(k-1)!}\int_a^x (x-t)^{k-1} g(t)\, dt, \tag{112}$$

where $x \in [a,b]$. Since the algorithm A gives the approximation of $u(x)$ for any x, the result for $x = b$ can be used to approximate the weighted integral $\int_a^b (b-t)^{k-1} g(t)\, dt$. We now use Novak's lower complexity bound for the integration problem, see [8]. He established the minimal cost of an algorithm for computing the integral for functions from a Hölder class, but the results he obtained are also valid for the class of functions with bounded derivatives of up to r order. The minimal cost of computing the integral with the accuracy ε in this class is $\Omega\left((1/\varepsilon)^{1/(r+1)}\right)$. In our case the integrand vanishes at b. We can however restrict ourselves to a weight function bounded away from zero, by considering the functions $g \in \mathcal{G}_1^r$ with support $[a_1, b_1] \subset (a,b)$ ($a_1 < b_1$, a_1, b_1 fixed). In this case

$$\int_a^b (b-t)^{k-1} g(t)\, dt = \int_{a_1}^{b_1} (b-t)^{k-1} g(t)\, dt. \tag{113}$$

Hence, the algorithm A allows us to approximate the right-hand side integral in (113). We now apply Novak's lower bound to the right-hand side of (113), and get the desired lower bound for the initial-value problem considered.

In the randomized setting, we proceed in a similar way. The crucial point is the lower bound for randomized integration, which is $\Omega\left((1/\varepsilon)^{1/(r+1/2)}\right)$. This result follows from Bakhvalov, see [1]. □

Comparing Theorem 6.1 with the bounds obtained in Theorem 5.5 we conclude that the algorithms defined in Section 4 are almost optimal, up to an arbitrarily small positive parameter γ in the exponent, and a logarithmic factor.

7. NUMERICAL EXPERIMENTS

In this section we present simulation results confirming the bound from Theorem 5.1 in the randomized setting. Let us recall that an optimal error of a deterministic algorithm with cost M is of order M^{-r}. Theorem 5.5 tells that randomized algorithm with this cost yields the error of order $M^{-(r+1/2-\gamma)}$. The parameter γ can be arbitrarily small. For all experiments we have chosen $\gamma = 1/4$ (then $K = 3$). Firstly, we want to

confirm that the error of the algorithm is independent of k (the order of the equation considered). Secondly, we want to check if the exponent $r + 1/2 - \gamma$ can be achieved.

We study the relation between cost and error of the algorithm. According to the obtained results,

$$e_M^{\mathrm{rand}}(A^K, \mathcal{G}^r) = C\left(M^{-(r+1/2-\gamma)}\right), \tag{114}$$

so that

$$\log(1/e_M^{\mathrm{rand}}(A^K, \mathcal{G}^r)) = a \log M + b, \tag{115}$$

where $a = (r + 1/2 - \gamma)$, $b = -\log C$, and the constant C varies for the upper and lower bound. This is a linear dependence of logarithm of inverse of the error on logarithm of the cost. An approximation of $e_M^{\mathrm{rand}}(A^K, \mathcal{G}^r)$ is computed by repeating the algorithm a number of times, and taking the mean.

We started with simple linear right-hand sides and with different k. We consider the following four equations on $[0, 1]$ with initial values equal to 1, such that the solution in each case is $u(x) = \exp(x)$:

1) $u'(x) = u(x)$,
2) $u''(x) = u'(x)$,
3) $u'''(x) = u''(x)$,
4) $u^{(4)}(x) = u'''(x)$.

For each equation, we consider three cases, taking r equal to $1, 2$ or 3. We have performed numerical experiments for a basic parameter $n = 2, 3, \ldots, 8$. In each case we have calculated cost of the algorithm and the randomized error of the obtained approximation. To confirm dependence (115), we have determined the regression line.

The results are presented in Table 1. The plots in Figure 1 show the linear dependence of logarithm of inverse of the error on logarithm of the cost, as expected.

Due to (115), the slope of the regression line should be equal to $r + 0.25$. The results presented in Table 1 are close to this value. The slope is independent of the order k, as expected.

In the next experiments we have checked if the parameter q influences the error, that is the slope of the regression line. We have considered the following equations:

5) $u^{(4)}(x) = u'''(x)$,
6) $u^{(4)}(x) = u''(x)$,
7) $u^{(4)}(x) = u'(x)$,
8) $u^{(4)}(x) = u(x)$,

with initial conditions equal to 1. We took $r = 1, 2, 3$ and $n = 2, \ldots, 8$.

From the results presented in Table 2 and Figure 2, we conclude that q does not affect the rate of convergence.

We have also made an experiment for a linear problem with non-constant coefficients:

9) $u''(x) = u(x)/x^2 - u'(x)/x, \quad u(1) = 1,\ u'(1) = 0$, where $x \in [1, 2]$.

Table 1. The coefficients of regression lines for problems 1–4 and $r = 1, 2, 3$

Equation	k	q	r	b	a
1	1	0	1	−2.011 239 81	1.279 940
1	1	0	2	−3.193 753 09	2.261 855
1	1	0	3	−4.536 029 14	3.242 381
2	2	1	1	−1.075 061 68	1.290 987
2	2	1	2	−2.821 233 28	2.259 749
2	2	1	3	−5.226 761 93	3.233 043
3	3	2	1	−0.042 516 61	1.299 979
3	3	2	2	−1.989 264 32	2.242 745
3	3	2	3	−5.708 448 17	3.243 509
4	4	3	1	1.446 208 58	1.290 628
4	4	3	2	−0.885 133 74	2.235 348
4	4	3	3	−5.459 259 95	3.237 311

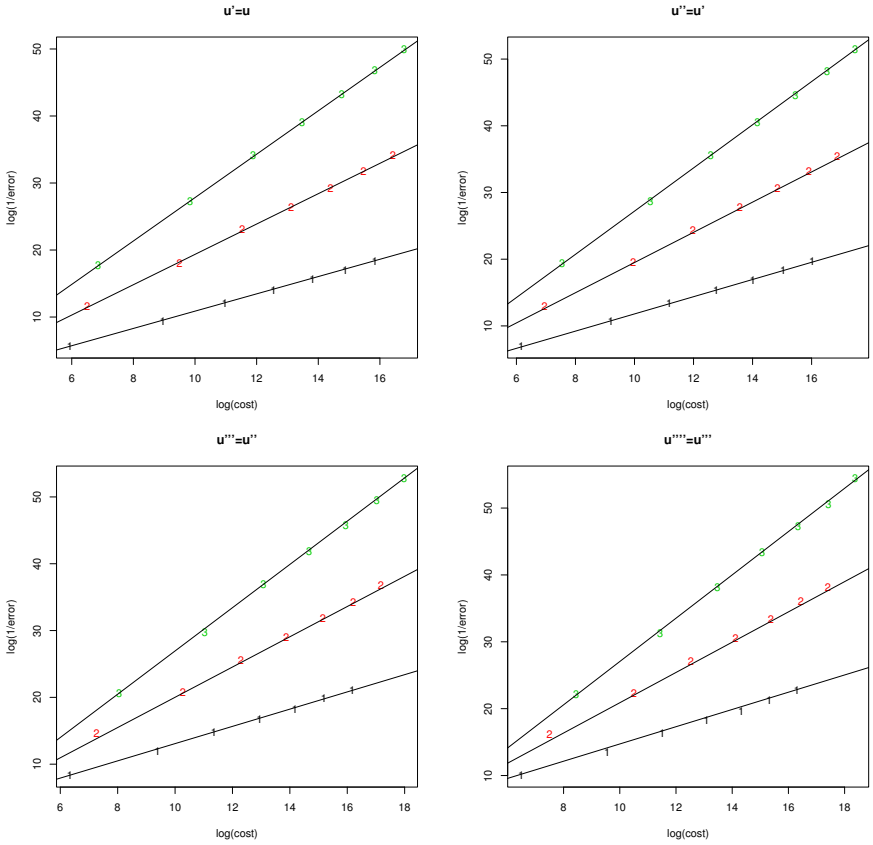

Fig. 1. Regression lines for problems 1–4 and $r = 1, 2, 3$

Table 2. The coefficients of regression lines for problems 5–8 and $r = 1, 2, 3$

Equation	k	q	r	b	a
5	4	3	1	1.446 208 58	1.290 628
5	4	3	2	−0.885 133 74	2.235 348
5	4	3	3	−5.459 259 95	3.237 311
6	4	2	1	2.018 836 57	1.276 988
6	4	2	2	−0.389 859 80	2.260 913
6	4	2	3	−3.915 468 74	3.248 108
7	4	1	1	2.018 118 77	1.282 098
7	4	1	2	0.098 357 80	2.258 445
7	4	1	3	−3.010 797 23	3.305 967
8	4	0	1	2.169 643 43	1.263 840
8	4	0	2	0.405 387 96	2.297 799
8	4	0	3	−0.744 591 91	3.245 933

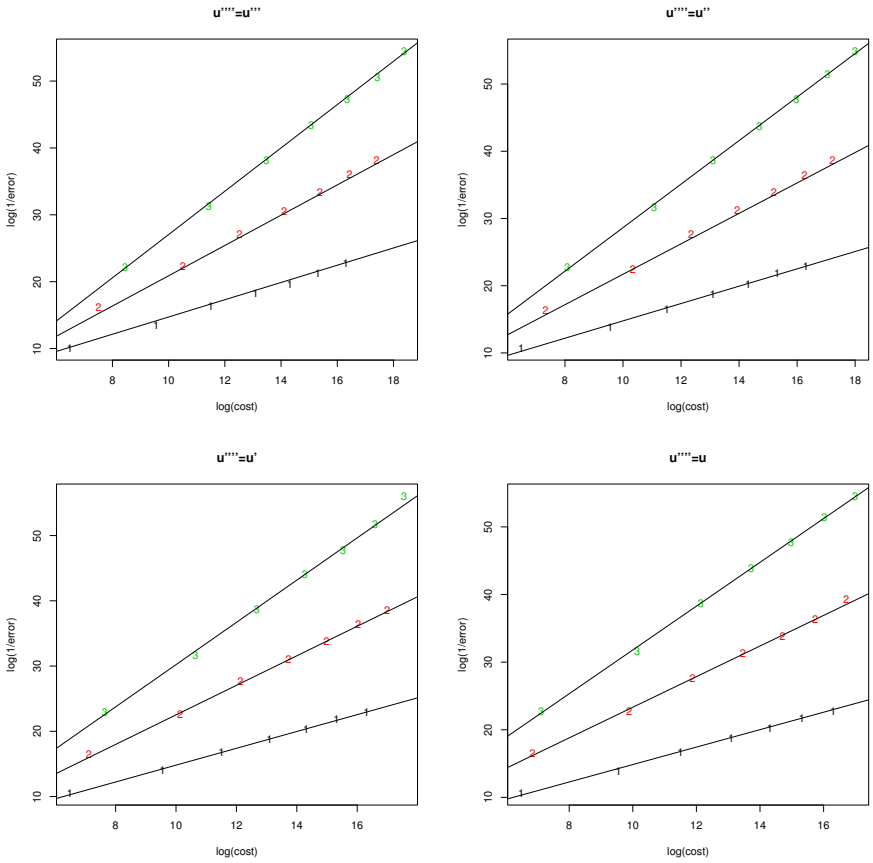

Fig. 2. Regression lines for problems 5–8 and $r = 1, 2, 3$

As previously, the results (presented in Tab. 3 and Fig. 3) confirm linear dependence of logarithm of inverse of the error on logarithm of the cost. The empirically determined slopes are close to theoretical ones.

Table 3. Regression lines for nonlinear problem 9 and $r = 1, 2, 3$

Equation	k	q	r	b	a
9	2	1	1	−0.713 175 7	1.353 947
9	2	1	2	−3.209 809 4	2.278 356
9	2	1	3	−7.647 463 7	3.263 805

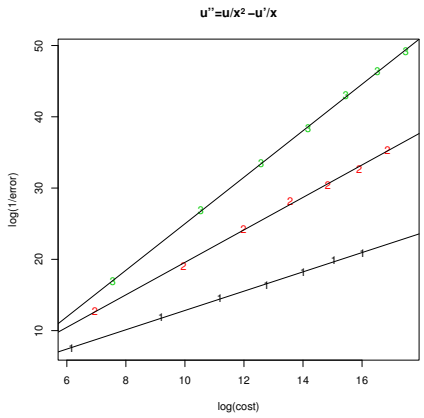

Fig. 3. Regression lines for nonlinear problem 9 and $r = 1, 2, 3$

8. SUMMARY AND COMMENTS

We have studied the complexity of scalar initial-value problems of higher order. We have considered two models of computations: randomized and quantum ones.

We presented algorithms solving problems of form (1) in both settings. These algorithms do not require passing to the system of the first order equations. The error analysis allowed us to establish upper complexity bounds for solving initial-value problems of order k. We also showed almost sharp lower bounds on ε-complexity for this problem. The bounds are independent of k, and are of the same order as in the case of the problem of system of first order equations. Comparing the bounds obtained with results known in the deterministic case, we see that randomized computation gives us a speed-up by $1/2$, and quantum computation by 1. We also included results of some numerical experiments. Even for small values of n, they confirm theoretical results.

APPENDIX

Sketch of the proof of Lemma 5.2. We only give main ideas of the proof. The proof is by induction on s. Let $s = 1$. From the simple calculations, we infer that there exists a constant D dependent on class parameters and k only, such that for $i = 0, \ldots, n-1$

$$\left|\bar{u}_i^{(j)}(x_i)\right| \leq \begin{cases} z_i & \text{for} \quad j = 1, \ldots, k-1, \\ D(1 + \ldots + z_i^{j-k}) & \text{for} \quad j = k, \ldots, k+r. \end{cases} \tag{116}$$

Hence and by the definition of y_{i+1}^j, we get that for $j = 1, \ldots, k-1$

$$|y_{i+1}^j| \leq z_i e^h + h e^h D(1 + \ldots + (hz_i)^{r-1}). \tag{117}$$

Then putting $c_i = hz_i$, we obtain the following recurrence inequality

$$c_{i+1} \leq c_i e^h + h^2 e^h D(1 + \ldots + c_i^{r-1}). \tag{118}$$

Let $F^1 := 3D$. It can be shown that if $hQ < 1$ then, for $i = 0, \ldots, n-1$,

1. $c_i < 1$, (119)
2. $c_i \leq c_0 e^{d-c} + h e^h D r \left(e^{d-c} - 1\right)$. (120)

From (120), for $h \leq 1$, we conclude that $Q_i := z_i < Q$, what ends the proof of the first statement of the lemma for $s = 1$. In order to prove the second one, let us note that for the approximation functions of local problems $\hat{l}_i^0$, the following bounds hold:

$$\|\hat{l}_i^{0(j)}(\cdot)\|_{[x_i, x_{i+1}]} \leq \begin{cases} z_i e^h + h^{k-j} e^h D \left(1 + \ldots + (z_i h)^{r-1}\right) & \text{for } j = 1, \ldots, k-1 \\ e^h D \Big(1 + \ldots + z_i^{j-k-1} & \\ \qquad + z_i^{j-k}\left(1 + \ldots + (z_i h)^{r+k-j-1}\right)\Big) & \text{for } j = k, \ldots, k+r-1, \end{cases} \tag{121}$$

with the same constant D as in (116). Due to (119), $z_i h < 1$. According to the definition, $\hat{l}^1(x) := \hat{l}_i^0(x)$ for $x \in [x_i, x_{i+1}]$. Hence for $h \leq 1$ there exists a constant B depending on class parameters, a, b and k only such that

$$\|\hat{l}^{1(j)}(\cdot)\|_{[c,d]} \leq \begin{cases} B(1+Q) & j = 1, \ldots, k-1 \\ B(1 + Q + \ldots + Q^{j-k}) & j = k, \ldots, r+k-1 \end{cases}, \tag{122}$$

which completes the proof for $s = 1$.

Let us now assume that the Lemma holds for degree $s-1$. We will prove it for s. From the inductive assumption for $A^{s-1}([x_i, x_{i+1}], m, Y_i)$, for each i there exist constants F^{s-1} and B such that if $Q_i = z_i e^h + r F^{s-1}\left(e^h - 1\right)$ and m satisfies $hQ_i/m < 1$, then

$$\|\hat{l}_i^{s-1(j)}(\cdot)\|_{[x_i, x_{i+1}]} \leq \begin{cases} B(1+Q_i) & j = 1, \ldots, k-1 \\ B(1 + \ldots + Q_i^{j-k}) & j = k, \ldots, r+k-1 \end{cases}, \tag{123}$$

Let us assume for the while that the assumption $hQ_i/m < 1$ holds for each i. We now derive the bounds on $eq : y_{i+1}^j$ defined by (30). An easy computation shows that there exists a constant E dependent on class parameters, a, b and k only, such that

$$\|g_{i,p}^j(\cdot)\|_{[0,1]} \leq E(1 + \ldots + Q_i^r) \qquad \text{for} \quad j = 0, \ldots, k-1, \tag{124}$$

and

$$\|w_{i,p}(\cdot)\|_{[x_{i,p}, x_{i,p+1}]} \leq E\left(1 + \ldots + (Q_i\bar{h})^r\right). \tag{125}$$

The bounds hold for $i = 0, \ldots, n-1$, $p = 0, \ldots, ml-1$. Let us recall that in the definition of the initial values y_{i+1}^j we use some randomized or quantum procedures giving the (random) numbers $AP_i^j(g)$. From [7] and [2], we can conclude the existence of a constant $\tilde{C}$ that there holds deterministically for any i and j

$$\left|AP_i^j(g)\right| \leq \tilde{C} \max_{i,j,p} \|g_{i,p}^j(\cdot)\|_{[0,1]}. \tag{126}$$

Using the inductive assumption and the above bounds, we get

$$\begin{aligned} \|y_{i+1}^j\| \leq z_i e^h + h^{k-j} \frac{1}{(k-1-j)!} E\left(1 + \ldots + (Q_i\bar{h})^{r-1}\right) + \\ + \bar{h} h^{k-j} \tilde{C} E\left(1 + \ldots + Q_i^r\right) \end{aligned} \tag{127}$$

This yields the following inequality (for $\bar{h} \leq 1$):

$$z_{i+1} \leq z_i e^h + (1 + \tilde{C}) hE\left(1 + \ldots + (Q_i\bar{h})^r\right). \tag{128}$$

Putting $c_i = Q_i\bar{h}$, we get the recurrence formula

$$c_{i+1} \leq c_i e^h + (1 + \tilde{C}) h\bar{h} E\left(1 + \ldots + c_i^r\right). \tag{129}$$

This inequality holds under the condition that the assumptions of inductive step for $s-1$ are satisfied. That means that the parameter m satisfies $hQ_i/m < 1$. Let us put $F^s = 3E + F^{s-1}$. It can be shown (by an induction on i) that if $hQ < 1$ (see the assumption of the Lemma), then for $i = 0, 1, \ldots, n-1$

1. $\forall m \in \mathbb{N} \quad hQ_i/m < 1,$ (130)
2. $c_{i+1} \leq c_i + (1 + \bar{C}) h\bar{h} e^h E(r+1),$ (131)
3. $Q_i \leq Q.$ (132)

From (130), we infer that the statements of Lemma hold true for $s-1$. Because $\hat{l}^s(x) := \hat{l}_i^{s-1}(x)$ for $x \in [x_i, x_{i+1})$, then applying (132) in (123) we obtain

$$\|\hat{l}^{s(j)}(\cdot)\|_{[c,d]} \leq \begin{cases} B(1+Q) & j = 1, \ldots, k-1 \\ B(1 + \ldots + Q^{j-k}) & j = k, \ldots, r+k-1 \end{cases}, \tag{133}$$

which is our claim. □

Acknowledgements
This research was partly supported by AGH grant No. 10.420.03. Part of calculations was carried out at ACK–CYFRONET-AGH. The machine time on IBM BladeCenter HS21 is financed by the Polish Ministry of Education and Science under grant No. MNiSW/IBM_BC_HS21/AGH/082/2007.

REFERENCES

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

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

[3] S. Heinrich, *Quantum approximation I. Embeddings of finite dimensional Lp spaces,* J. Complexity **20** (2004), 5–26; see also http://arxiv.org/abs/quant-ph/0305030.

[4] B. Kacewicz, *Optimal solution of ordinary differential equations,* J. Complexity **3** (1987), 451–465.

[5] B. Kacewicz, *Randomized and quantum algorithms yield a speed-up for initial-value problems,* J. Complexity **20** (2004), 821–834; see also http://arxiv.org/abs/quant-ph/0311148.

[6] B. Kacewicz, *Almost optimal solution of initial-value problems by randomized and quantum algorithms,* J. Complexity **22** (2006), 676–690; see also http://arxiv.org/abs/quant-ph/0510045.

[7] P. Mathé, *Random approximation of finite sums,* Institute for Applied Analysis and Stochastics, Berlin, Preprint 11, 1992.

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

[9] M. Szczęsny, *Complexity of initial-value problems for ordinary differential equations of order k,* J. Complexity **22** (2006), 821–834.

Maciej Goćwin
gocwin@uci.agh.edu.pl

AGH University of Science and Technology
Faculty of Applied Mathematics
al. Mickiewicza 30, 30-059 Cracow, Poland

Marek Szczęsny
szczesny@uci.agh.edu.pl

AGH University of Science and Technology
Faculty of Applied Mathematics
al. Mickiewicza 30, 30-059 Cracow, Poland

Received: July 4, 2007.
Revised: March 17, 2008.
Accepted: April 11, 2008.