Error analysis of numerical methods for optimization problems

Alexey V. Chernov, Alexandr G. Birjukov, Anna M. Lisachenko
Phystech-School of Applied Mathematics and Informatics
Moscow Institute of Physics and Technology
Moscow, Russian Federation
Federal Research Center "Computer Science and Control"
of the Russian Academy of Sciences
Moscow, Russian Federation
chernov.av@mipt.ru

Julia G. Chernova
Department of Mechanics and Mathematics
Moscow State University
Moscow, Russian Federation

Abstract

The article considers methods for constructing solution error estimates in optimization problems. Solution error estimates can be divided into two classes: theoretical and numerical. Estimates of the first class are based on a theoretical analysis of the convergence of the problem-solving method. These theoretical estimates are functions of the problem parameters and solution methods, whose values are often difficult to determine. Therefore, they are primarily useful for qualitative analysis. Numerical estimates, known for a limited number of optimization methods, provide explicit numerical values. In this paper, we propose two new methods for constructing numerical error estimates where the error estimate is a known function of values calculated during the optimization process. These new methods apply to a broad class of optimization problems and solution methods: the objective function is defined on a closed set in an n-dimensional Euclidean space and is continuous. The optimization method involves a monotonically decreasing sequence of objective function values.

The first method is based on a three-point scheme. From the decreasing sequence of function values, a group of three elements is selected where the ratio of consecutive deviations is less than unity. An exact error estimate formula is derived, which depends on the optimal value of the objective function. Using a function value calculated at a finite step, a sufficiently accurate error estimate can be obtained.

The second method, called the rounding method, assumes that the number of significant digits in the optimal solution increases with iterations. This condition allows for estimating the solution error at each iteration. The objective function value in this method corresponds to the rounding of the optimal solution.

This paper presents numerical experiment results for estimating the solution error based on the objective function values and the norm of the argument values as deviations from the optimal values.

1 Introduction

The article considers methods of constructing error estimates for numerical solutions of optimization problems of the form (1):

$$f(\boldsymbol{x}) \to \min_{\boldsymbol{x} \in \mathbb{G}}, \tag{1}$$

where objective function $f : \mathbb{R}^n \to \mathbb{R}^1$ is continuous function and the feasible set $\mathbb{G} \subset \mathbb{R}^n$ is connected.

Further we assume that:

- problem's (1) solution $\boldsymbol{x}^* \in \mathbb{G}$, $f(\boldsymbol{x}^*)$ exists;
- the problem is solved by some numerical method, which generates in the process of solving a sequence of points $\boldsymbol{x}^k{}_{k=0}^{\infty}$ and a corresponding sequence of values of the objective function $\{f_k\}_{k=0}^{\infty}$;
- $\boldsymbol{x}^* = \lim\limits_{k\to\infty} \boldsymbol{x}^k$;
- $f_k = f(\boldsymbol{x}^k) \to f^*$ where $k \to \infty$;
- the sequence $\{f_k\}_{k=0}^{\infty}$ is strictly decreasing: $f_k > f_{k+1}\ \forall k \geq 0$;
- the execution of the numerical method is stopped at the step N in the point $\boldsymbol{x}^N$ with the function's value $f_N = f(\boldsymbol{x}^N)$.

Let ε_1 and ε_2 be a small numbers.

Proposition 1 *The problem's (1) solution is calculated with ε_1 accuracy by coordinate, if*

$$\|\boldsymbol{x}^N - \boldsymbol{x}^*\| \leq \varepsilon_1;$$

The problem's (1) solution is calculated with ε_2 accuracy by function, if

$$f(\boldsymbol{x}^N) - f(\boldsymbol{x}^*) \leq \varepsilon_2.$$

In practice, it is not always possible to obtain the estimates indicated in the Proposition 1, so often some heuristic rules are used instead (Gill et al., 1981), for example:

- $\|\nabla f(\boldsymbol{x})\| \leq \varepsilon$ for the unconstrained minimization problem;
- $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\| \leq \varepsilon$ or $\left|f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k+1})\right| \leq \varepsilon$.

Currently in the study of optimization methods, estimates (2) that depend on the parameters of the problem are widely used.

$$f(\boldsymbol{x}^k) - f(\boldsymbol{x}^*) \leq \varphi(L, R, \mu, k) \leq \varepsilon. \tag{2}$$

In (2) L is the Lipschitz constant for $\nabla f(\boldsymbol{x})$ or $f(\boldsymbol{x})$, μ is the strong convexity constant, $R = \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|$ (Gasnikov, 2018) is the distance between initial point and problem's solution. Ideas for constructing the function φ were proposed in Nemirovski & Iudin (1979) and extended in Nesterov (2018),Bubeck (2015) and other works. However, the task of calculating such parameters often turns out to be more complicated than the original optimization problem, and it is difficult to use such estimates in practice.

In Biryukov & Grinevich (2012),Biryukov & Grinevich (2013), calculations with variable mantissa length were used to obtain estimates of the solution error according to the Proposition 1. Such approaches were used in numerical experiments, the analysis of which is given in this paper. In this paper, we build estimates of the error of solving the problem (1) of the form (3) and (4):

$$f(\boldsymbol{x}^k) - f(\boldsymbol{x}^*) \leq \varphi_1(\boldsymbol{x}^k, \boldsymbol{x}^N, \nabla f(\boldsymbol{x}^k), f(\boldsymbol{x}^k)) \leq \varepsilon; \tag{3}$$

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\| \leq \varphi_2(\boldsymbol{x}^k, \boldsymbol{x}^N, \nabla f(\boldsymbol{x}^k), f(\boldsymbol{x}^k)) \leq \varepsilon. \tag{4}$$

In (3), (4) $\boldsymbol{x}^N$, $f(\boldsymbol{x}^N)$ is some approximate solution of the (1) and N is the number of the iteration it was obtained.

Definition 1 *The control solution of task (1) is the point $\boldsymbol{x}^N$, $f(\boldsymbol{x}^N)$.*

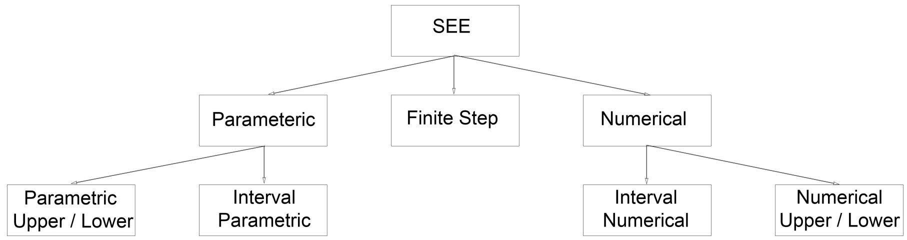

Figure 1: Classification of the solution error estimates

These types of solution error estimates (2), (3), (4) are not unique. Figure 1 suggests some classification of them also mentioned in Biryukov & Chernov (2021).

The classes of solution error estimates (SEE) presented in Figure 1 are the following:

- Parametric estimate of the solution error (an example of such an estimate is (2), which includes such parameters as the Lipschitz constant L, the strong convexity constant μ, etc.). Such estimates are often used to analyze the convergence rate of the method.
- Finite step – solution error estimate for finite-step numerical methods (namely, methods that in exact arithmetic find the exact solution of the problem in a finite number of steps).
- Numerical (or experimental) estimates of the solution error, e.g., (3) and (4), where the functions φ_1 and φ_2 are known, and the values of arguments at each step of the iterative process are also known.
- Interval solution error estimates;
- Upper / lower solution error estimates.

Various examples of estimates can be given for the most of these classes, but the upper numerical estimates of the solution error are unknown to the authors of this article. The purpose of this work is to obtain estimates of the error of the solution $f_i - f^*$, where f_i is the element of the sequence f_k, $k = 1, \ldots$ generated by numerical method.

Let us introduce the following notations for $i = 0, 1, 2, \ldots$:

$$\delta_i = f_i - f_{i+1}; \quad g_i = \delta_{i+1}/\delta_i \equiv \frac{f_{i+1} - f_{i+2}}{f_i - f_{i+1}} < 1.$$

Then the following lemma is proved in Appendix A:

Lemma 1 *Let the sequence* $\{f_i\}_{i=0}^{\infty}$ *satisfy the condition* $g_i < 1 \ \forall i \geq 0$*, the value* $\hat{g} = \max_{i \geq 0} g_i$ *exists and* $\hat{g} < 1$*, then the following solution error estimate takes place:*

$$f_i - f^* \leq \frac{\delta_i}{1 - \hat{g}}. \tag{5}$$

To obtain an estimate of the value $\hat{g}$ for the sequence $\{g_k\}_{k=i}^{\infty}$ requires a separate study, which is beyond the scope of this paper.

2 The basis for the three-point scheme

Consider one way of constructing an estimate of the solution error based on Lemma 1.

From the sequence $\{f_k\}_{k=i}^{\infty}$ where $f_k = f(\boldsymbol{x}^k)$ and $f_k > f_{k+1}$, select the elements f_i, f_j, f_l and f_N such that

$$f_N < f_l < f_j < f_i; \quad f_i + f_l - 2f_j > 0. \tag{6}$$

Let us denote the difference of function values at points with the indices i and j as δ_{ij}: $\delta_{ij} = f_i - f_j$, and the value g_{ijl} as the ratio of the corresponding values between points j and l and points i and j respectively: $g_{ijl} = \delta_{jl}/\delta_{ij}$. Therefore, by virtue of the assumption (6) we obtain $g_{ijl} < 1$.

Consequently, the estimate (5) with $\hat{g} = g_{ijl}$ are of the form:

$$f_i - f^* \leq \frac{\delta_{ij}}{1-\hat{g}} = \frac{f_i - f_j}{1 - \delta_{jl}/\delta_{ij}} = \frac{(f_i - f_j)^2}{f_i + f_l - 2f_j}. \tag{7}$$

Next, we investigate the conditions for the fulfillment of the formula given f_i, f_l, and varying f_j.

Theorem 1 *Let the condition (6) be satisfied. Then the estimate (7) is valid if and only if*

$$0 \leq f_j^2 - f_i f_l + f^*(f_i + f_l - 2f_j). \tag{8}$$

The proof is presented in Appendix B.

Remark 1 *The values f_i, f_{i+1}, f_{i+2} in the lemma 1 is identified by the only index i. The identification of the elements f_i, f_j, f_l are determined according to (6) and thus except the variation of the one index i, one can variate three indexes i, j, l. It is shown further that for the selected indexes i, l one can find the index j for which estimate (7) are exact.*

Remark 2 *The union of the numerical method used to solve the optimization problem (1) which builds the decreasing sequence $\{f_k\}_{k=0}^{\infty}$ and the method used to retrieve estimates $f_i - f^*$ from (6) is called the three-point scheme of the solution error estimation method (7) for the optimization problem (1).*

Remark 3 *The three-point scheme of the solution error estimation method is usefull for the wide class of the optimization problems (1) and numerical methods that are used to solve this problems. For example three-point scheme is applicable for the gradient descent method, conjugate gradient method, quasi-newtons methods in case of unconstrained optimization problems and for conditional gradient method, gradient projection method, modified Lagrange function method in case of mathematical programming problems where a decreasing sequence $\{f_k\}_{k=0}^{\infty}$ is generated.*

Further, we can state the following.

Theorem 2 *In the sequence $\{f_k\}$, $k = 1, 2, 3, \ldots$, which converges to f^*, there are points i and l such that $f^* < f_l < f_i$. And there exists $\tilde{j}$ where the inequality (6) is satisfied. Then there exists a point j such that $f^* < f_l < f_j < f_i$ and $f_l < f_0 \leq f_j < f_c$, $f_0 = f^* + \sqrt{(f_i - f^*)(f_l - f^*)}$ and the condition (7) is satisfied. If $f_j = f_0$, i.e.*

$$f_j = f^* + \sqrt{(f_i - f^*)(f_l - f^*)}. \tag{9}$$

then the estimate (7) is exact.

The proof is presented in Appendix C.

Remark 4 *For the practical implementation of the following proposed three-point scheme for constructing an estimate of the solution error it is necessary to carry out the lower and upper estimates of the value $f_j = f_0$ in (9), which are done further in Section 4.*

Remark 5 *The scheme studied above for estimating the error of the solution considered the variation of f_j when f_i and f_l are known. However, when f_i, f_j are known, it is possible to find the value of f_l at which the estimate of (7) is accurate. It follows from (9) that*

$$f_l = f^* + \frac{(f_j - f^*)^2}{f_i - f^*}.$$

3 The method of rounding function values in solution error estimation

It is difficult to use the condition (9) because f^* is unknown. To get around this, we consider two estimates for the value. Let $\overline{f^*}$ and $\underline{f^*}$ be the upper and lower estimates of f^*, respectively. Then the estimate for $f_i - f^*$ take the form: $f_i - f^* \leq f_i - \overline{f^*}$, which is incorrect. For example, the upper estimate is $f_N = f(\boldsymbol{x}^N)$. Also we have the inequality $f_i - f^* \leq f_i - \underline{f^*}$, which is already the required estimate of the solution error. Finding a lower estimate $\underline{f^*} \leq f^*$ can sometimes be easy, but it can be rough.

Consider one method that can give fairly accurate estimates of both the lower and upper estimate of the solution error (Biryukov & Chernov, 2021). Let the decimal representation (10) of the value of function f_s at some step s with mantissa length $m + 1$.

$$f_s = \pm a_0^s, a_1^s a_2^s \dots a_m^s \cdot 10^{t_s} \tag{10}$$

In (10) t_s is the order of the number, $1 \leq a_0^s \leq 9$, $0 \leq a_i^s \leq 9$, $i = \overline{1, m}$. Since the sequence $\{f_i\}$ is convergent, for large enough N the value of f_N is "close enough" to f^*, i.e. the numbers f_N and f_i have the same first few digits of a_r: $a_r = a_r^N = a_r^i$, $r = \overline{0, m_i}$. And the number of m_j matching digits increases monotonically as the value of j changes from i to $N - 1$. We assume that f_i and f^* also have m_i matching digits. Under such conditions it is possible to specify solution error estimate. Indeed, if we have $m_i + 1$ of matching digits, then for $f^* > 0$

$$\underline{f_i^*} = a_0, a_1 a_2 \dots a_{m_i - 1} a_{m_i} \cdot 10^t; \quad \overline{f_i^*} = a_0, a_1 \dots a_{m_i - 1}(a_{m_i} + 1) \cdot 10^t. \tag{11}$$

Thus, the number f^* is rounded both downward and upward. Now let $f^* < 0$. To round a negative number down, we round the number $|f^*|$ up and vice versa. If the number of matching signs $m_i + 1$, then

$$\underline{f_i^*} = -a_0, a_1 a_2 \dots a_{m_i - 1}(a_{m_i} + 1) \cdot 10^t; \quad \overline{f_i^*} = -a_0, a_1 \dots a_{m_i - 1} a_{m_i} \cdot 10^t. \tag{12}$$

Hence the following estimate follows, which is proved in Appendix D.

Theorem 3 *Let the numbers f_s, $s = \overline{i, N - 1}$ and f_N have the same first $m_s + 1$ digits, and $m_{s+1} \geq m_s$. Then there is a solution error estimate*

$$f_s - f^* \leq f_s - \underline{f_s^*} \leq 10^{t - m_s}.$$

Remark 6 *The applicability of the rounding method for estimating solution error is based on the assumption that f_{N-1}, f_N, f^* have the same number of the matching first digits m_{N-1}.*

4 A THREE-POINT SCHEME FOR CONSTRUCTING AN ESTIMATE OF SOLUTION ERROR

The previously obtained estimates $\underline{f^*}$ and $\overline{f^*}$ (11), (12) are applicable to obtain the estimate (7). Note that in the strictly decreasing sequence $f^* < f_s$, $s = \overline{1, N}$ and f_s are upper estimates for f^*. If s is large enough, $f_s - f^*$ can be small enough, f_N is the best upper estimate of f^* (see Figure 7). Then $f_N \leq \overline{f_N^*}$, where $\overline{f_N^*}$ is defined in (11) and (12) at $i \equiv N$.

Theorem 4 *Let the Theorem 3 estimate the error of the solution and the lower and upper estimates (27) of the number $f_j = f_0$ in (9) be valid. Then for the elements of the sequence with numbers $s = \overline{i, N - 1}$, the following solution error estimate is valid.*

$$f_i - f^* = \frac{(f_i - f_0)^2}{f_i + f_l - 2f_0} \leq \frac{(f_i - \underline{f_j})^2}{f_i + f_l - 2\overline{f_j}}, \tag{13}$$

where

$$\underline{f_j} = \underline{f^*} + \sqrt{(f_i - \overline{f^*})(f_l - \overline{f^*})}; \quad \overline{f_j} = \overline{f^*} + \sqrt{(f_i - \underline{f^*})(f_l - \underline{f^*})}. \tag{14}$$

Proof 1 *The quation (13) is obtained from (7). Substituting the lower and upper estimates for f_j $\underline{f_j}$ and $\overline{f_j}$ in (9), we get an estimate of the error of the solution $f_i - f^*$. Here i is a number of $\overline{1, \ldots, N-1}$, and f_j, f_l are numbers satisfying the conditions (6) for the selected i. The most accurate lower estimates in (13) and (14) for f^* are determined by the formulas (11) and (12) for $m_i \equiv m_{N-1} \equiv m_N$, and for the upper estimate $\overline{f^*}$: $\overline{f^*} = f_N$.*

Using (9) and (14), we also get an estimate for $f_j - f^*$:

$$f_j - f^* \leq \sqrt{(f_i - \underline{f^*})(f_l - \underline{f^*})}.$$

The method of constructing an estimate of the error of the solution using (13) and (14) is called a three-point scheme, and the method of constructing an estimate of the error of the solution based on (28) is the rounding method.

5 On the accuracy of error estimation in the rounding method

The next step in analyzing the methods of constructing the estimation of the solution error is to compare the accuracy of the above methods.

The above methods of constructing the error estimate of the solution of the form $\hat{\delta}_i = f_i - f^*, i = \overline{1, N}$ for optimization problems are based on the assumption that $\{f_i\}$ is a monotonically convergent sequence. We can consider a class of sequences assuming that $\{f_i\}$ is only a convergent sequence, but not necessarily monotone.

Let us mention the following example. Suppose we are solving an extremal problem , where $\{f_i\}$ is a monotonically convergent sequence, and $\{\|\boldsymbol{x}_i\|\}$ is also a convergent sequence, but not necessarily monotone. Consider two sequences of points $\|\boldsymbol{x}_k\|$

$$\begin{array}{l} L_+ = \left\{k \in [1, N] : \|\boldsymbol{x}_k\| \geq \|\underline{\boldsymbol{x}_N}\|\right\}; \\ L_- = \left\{k \in [1, N] : \|\overline{\boldsymbol{x}_N}\| \geq \|\boldsymbol{x}_k\|\right\}. \end{array} \tag{15}$$

Obviously, $L_+ \cup L_- = [1, N]$. We assume that $\|\underline{\boldsymbol{x}^*}\| = \|\underline{\boldsymbol{x}_N}\|$ and $\|\overline{\boldsymbol{x}^*}\| = \|\overline{\boldsymbol{x}_N}\|$.

Similarly to the monotonically convergent sequence, combining L_+ and L_-, we obtain an estimate of the error of the solution

$$\begin{array}{ll} \text{For } \boldsymbol{x}_k \in L_+ : & \|\boldsymbol{x}_k\| - \|\boldsymbol{x}^*\| \leq \|\boldsymbol{x}_k\| - \|\underline{\boldsymbol{x}_N}\| \\ \text{For } \boldsymbol{x}_k \in L_- : & \|\boldsymbol{x}^*\| - \|\boldsymbol{x}_k\| \leq \|\overline{\boldsymbol{x}_N}\| - \|\boldsymbol{x}_k\|. \end{array} \tag{16}$$

The accuracy of the solution error estimate in Theorem 3, which depends obviously on m_s, may not be sufficient for small m_s. However, using the inequality $f_{s+1} < f_s$ (for a monotonically decreasing sequence) at $m_{s+1} > m_s$ and the corresponding inequalities for a monotonically increasing sequence, we construct upper and lower estimates of the solution error with varying accuracy; in doing so, it is possible, if the specifics of the problem allow it, to achieve the required accuracy.

The error estimates have the form:

$$f_i - \overline{f_r^*} \leq f_i - f^* \leq f_i - \underline{f_s^*}, \quad s = \overline{i, N}, \quad r = \overline{i, N}, \quad m_i \geq 0 \tag{17}$$

for a monotonically decreasing sequence;

$$\underline{f_r^*} - f_i \leq f^* - f_i \leq \overline{f_s^*} - f_i, \quad s = \overline{i, N}, \quad r = \overline{i, N}, \quad m_i \geq 0 \tag{18}$$

for a monotonically increasing sequence;

$$\begin{array}{c} \|\boldsymbol{x}_i\| - \|\overline{\boldsymbol{x}_r^*}\| \leq \|\boldsymbol{x}_i\| - \|\boldsymbol{x}^*\| \leq \|\boldsymbol{x}_i\| - \|\underline{\boldsymbol{x}_s^*}\|, \quad i \in L_+ \\ \|\underline{\boldsymbol{x}_r^*}\| - \|\boldsymbol{x}_i\| \leq \|\boldsymbol{x}^*\| - \|\boldsymbol{x}_i\| \leq \|\overline{\boldsymbol{x}_s^*}\| - \|\overline{\boldsymbol{x}_i}\|, \quad i \in L_- \\ s \in [l, N], r \in [i, N], \quad m_i \geq 0 \end{array} \tag{19}$$

for arguments from (16).

The relations (17), (18), (19) determine the range of solution error estimates for the above problems. The lowest accuracy of the solution error estimates is at $s = i$ and $r = i$, and the highest at $s = N$, $r = N$. The corresponding number of matching digits in the representation of $\underline{f_s^*}$, $\overline{f_s^*}$, $\|\underline{\boldsymbol{x}_s^*}\|$, $\|\overline{\boldsymbol{x}_s^*}\|$, $\underline{f_r^*}$, etc. is $m_s + 1$ and $m_r + 1$ numbers. The resulting values of the lower and upper estimates, if they are redundant in accuracy, can also be rounded.

Remark 7 *The inequalities (17), (18), (19) are estimates of the solution error, and their boundary values can be interpreted as the accuracy of the solution in terms of function and argument, as follows*

$$\begin{array}{ll} \varepsilon_{i,r} \equiv f_i - \overline{f_r^*}; & \varepsilon_{i,s} \equiv f_i - \underline{f_s^*} \\ \varepsilon_{i,r} \equiv \underline{f_r^*} - f_i; & \varepsilon_{i,s} \equiv \overline{f_s^*} - f_i \end{array} \quad - \quad \textit{accuracy by function;} \tag{20}$$

$$\begin{array}{ll} \delta_{i,r} \equiv \|\boldsymbol{x}_i\| - \|\overline{\boldsymbol{x}_r^*}\|; & \delta_{i,s} \equiv \|\boldsymbol{x}_i\| - \|\underline{\boldsymbol{x}_s^*}\| \\ \delta_{i,r} \equiv \|\underline{\boldsymbol{x}_r^*}\| - \|\boldsymbol{x}_i\|; & \delta_{i,s} \equiv \|\overline{\boldsymbol{x}_s^*}\| - \|\boldsymbol{x}_i\| \end{array} \quad - \quad \textit{accuracy by argument;} \tag{21}$$

However, a question arises. What is the limit point $(\boldsymbol{x}^*, f^*)$ for a monotonically convergent and simply convergent sequence? Is it a point of minimum or maximum, local or global, or perhaps a saddle point? To get the answer, we need to apply some sufficient extremum condition to the point $\boldsymbol{x}_N$. If it is satisfied for the neighborhood $\|\boldsymbol{x} - \boldsymbol{x}^*\| \le \|\boldsymbol{x}_N - \boldsymbol{x}_k\| \le \varepsilon$, then the error estimate found $\delta_k = f_k - f^*$, $\big|\|\boldsymbol{x}_k\| - \|\boldsymbol{x}^*\|\big|$ is correct.

6 PRACTICAL ASPECT

In practice, many optimization problems are solved. In this case, users (solvers) believe that the problem is solved (i.e., the sufficient extremum condition is satisfied for $\boldsymbol{x}_N, f(\boldsymbol{x}_N)$) with sufficient accuracy for their purposes. However, they often cannot guarantee that they can estimate the solution error $\hat{\delta_k} = f(\boldsymbol{x}_k) - f^*$, $\big|\|\boldsymbol{x}_k\| - \|\boldsymbol{x}^*\|\big|$. Their estimates are approximate. The proposed methods for estimating the solution error when the above requirements are met give guaranteed estimates $\forall k : k = \overline{1, N}$.

7 EXPERIMENTS

At the beginning of this section note, that experiments were completed on the PC with RAM 32 GB and CPU Intel Core I9 $2.40(3.10)$ GHz with OS Windows HE 10 (64) in PyCharm IDE (Python 3.8). In this article we consider two test optimization problems: unconstrained optimization problem and mathematical programming problem. Also we completed the next step in the analysis of the solution error estimates and decided to select the test problems analyzed in Biryukov & Chernov (2021). Each test problem is described in the corresponding subsection below.

In all tables in the subsections below we use the notions defined in the table 1.

7.1 SMOOTH CONVEX UNCONSTRAINED OPTIMIZATION PROBLEM

Let us consider the problem (1) where $\mathbb{G} = \mathbb{R}^n$ with the following strongly convex objective function:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i (x_i - \beta_i)^2 + \exp^{\boldsymbol{c}^T \boldsymbol{x}} + \gamma \exp^{-\boldsymbol{c}^T \boldsymbol{x}}, \tag{22}$$

where $\alpha_i = i^2$, $\beta_i = 1$, $c_i = 0.1$ for all $i = 1, \ldots n$, $\gamma = 2$, $n = 40$.

To solve the problem Polak-Ribiere-Polyak method (Polyak, 2021) with golden search as a one-dimensional search is used (initial point $\boldsymbol{x}^0 = 1_n$, the mantissa is 50, the required accuracy of the solution is 10^{-8} for the norm of the objective function gradient (stop condition of the iteration process), one dimensional search accuracy is 10^{-11}). The results are summarized in Tables 2 and 3. Tables with detailed results of this and subsequent experiments can be found in Appendix E.

Column Name	Description
#	Iteration index k
$f(\boldsymbol{x}^k)$	The value of the function f at the point $\boldsymbol{x}^k$
ε_1	The value of the ε_1 at the point $\boldsymbol{x}^k$, where $\varepsilon_1 = 10^{t-m_k}$, m_k is the amount of the first same digits for values $f(\boldsymbol{x}^k)$ and $f(\boldsymbol{x}^N)$
ε_2	The error estimate calculated according to (13) with $\overline{f^*} = f(\boldsymbol{x}^N)$
$\lVert\boldsymbol{x}^k\rVert$	The difference between 2-norms of the k-th point $\boldsymbol{x}^k$ and the last point $\boldsymbol{x}^N$.
ε_3	The value of the ε_3 at the point $\boldsymbol{x}^k$, where $\varepsilon_3 = 10^{t-m'_k}$, m'_k is the amount of the first same digits for values $\lVert\boldsymbol{x}^k\rVert$ and $\lVert\boldsymbol{x}^{\tilde{N}}\rVert$, where $\tilde{N}$ is the last number from the set I_+ or I_- respectively
ε_4	The error estimate calculated according to (13) with $f(\boldsymbol{x}) := \lVert\boldsymbol{x}\rVert$ and $\overline{f^*} = \lVert\boldsymbol{x}^{\tilde{N}}\rVert$, where $\tilde{N}$ is the last number from the set I_+ or I_-

Table 1: Notions in the experiment tables

#	$f(x^k)$	ε_1	ε_2
0	627387.11631838745289500864	$1 \cdot 10^{6}$	509894.77
8	**1**31996.90602183602538406045	$1 \cdot 10^{5}$	14504.56
18	**117**854.15903353102929779504	$1 \cdot 10^{3}$	361.81
28	**11749**6.62956020062317238274	$1 \cdot 10^{1}$	4.28
48	**117492.345**99695254306091291	$1 \cdot 10^{-3}$	0.00024
63	**117492.34575223**507416076358	$1 \cdot 10^{-8}$	$2.3 \cdot 10^{-9}$
83	**117492.345752232757789**716425	$1 \cdot 10^{-14}$	$3.1 \cdot 10^{-15}$
98	**117492.3457522327578940667**7	$1 \cdot 10^{-19}$	$2.6 \cdot 10^{-20}$
103	**117492.34575223275789406675**	$1 \cdot 10^{-20}$	$4.3 \cdot 10^{-22}$

Table 2: Smooth unconstrained convex optimization problem output (short). Function.

#	$\lVert x^k\rVert$	ε_3	ε_4
0	001.2728880402482	$1 \cdot 10^{3}$	227.031
13	**2**12.8510755796911	$1 \cdot 10^{2}$	15.45
23	**22**7.5905753548338	$1 \cdot 10^{1}$	0.71
43	**228**.2592315075667	$1 \cdot 10^{0}$	0.044
73	**228.3035643**821351	$1 \cdot 10^{-6}$	$2.9 \cdot 10^{-7}$
88	**228.30356409**41619	$1 \cdot 10^{-8}$	$7.0 \cdot 10^{-10}$
98	**228.3035640935**230	$1 \cdot 10^{-10}$	$2.9 \cdot 10^{-11}$
103	**228.303564093514**2	$1 \cdot 10^{-12}$	$2.0 \cdot 10^{-13}$

Table 3: Smooth unconstrained convex optimization problem output (short). Points.

7.2 NON-SMOOTH CONVEX UNCONSTRAINED OPTIMIZATION PROBLEM

In this section we provide results of the experiment for the problem (1) for $\mathbb{G} = \mathbb{R}^n$ with continuous non-smooth convex objective function:

$$f(\boldsymbol{x}) = \sum_{j=1}^{2} d_j \left|\boldsymbol{c}_j^T \boldsymbol{x} - b_j\right| + \sum_{i=1}^{n} \alpha_i (x_i - \beta_i)^2, \tag{23}$$

where $\alpha_i = i^2$, $\beta_i = 1/2$, $i = 1, \ldots, n$, $d_1 = 2$, $d_2 = 1$, $\boldsymbol{c}_1 = 1_n$, $c_{2,j} = (-1)^{j+1}$, $j = 1, \ldots, n$, $b_j = \boldsymbol{c}_j^T \beta$, $j = 1, 2$[1], $n = 20$.

7.3 SMOOTH CONSTRAINED CONVEX OPTIMIZATION PROBLEM

In this section we consider the following constrained convex optimization problem:

[1] Parameters of the objective function allows explicitly find problem solution $\boldsymbol{x}^* = \beta$, $f(\boldsymbol{x}^*) = 0$.

$$
\begin{aligned}
&f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i (x_i - \beta_i)^2 + \exp^{\boldsymbol{c}^T \boldsymbol{x}} + \gamma \exp^{-\boldsymbol{c}^T \boldsymbol{x}} \to \min_{\boldsymbol{x} \in \mathbb{G}} \\
&\mathbb{G} = \left\{ \boldsymbol{x} \in \mathbb{R}^n : x_i \geq 0, i = 1, \dots, n-1; \sum_{i=1}^{n-1} i x_i^2 + x_n \leq n \right\}
\end{aligned} \tag{24}
$$

In the problem (24) we use parameters $\alpha_i = i^2$, $\beta_i = 1$, $c_i = 0.1$ for all $i = 1, \dots n$, $\gamma = 2$, $n = 40$.

To solve the problem (24) we used the method of modified Lagrange function (Polyak, 2021) (MMLF) where Polak-Ribiere-Polyak method is used to solve unconstrained optimization problem with accuracy $\|\nabla M(\boldsymbol{x}, \lambda_k)\| \leq 10^{-10}$ where $M(\boldsymbol{x}, \lambda)$ is modified Lagrange function.

8 Conclusion

This paper is a continuation of Biryukov & Chernov (2021). In that paper, we considered methods for estimating errors in convex optimization problems, including methods based on the rounding of strictly decreasing sequences of values of objective functions and their arguments. In this paper, such a method is used to construct high-precision estimates within the framework of the three-point scheme for constructing solution error estimates proposed in the paper. This method is applicable to a wide class of problems, including those not mentioned in this paper.

Tables in Appendix E present the results of numerical experiments on the construction of solution error estimates using the considered methods for the smooth and nonsmooth convex unconstrained optimization problem, as well as for the mathematical programming problem. Estimates were obtained at all points of the iterative process.

We believe that the proposed methods for constructing solution error estimates will find practical application, and further research will allow to obtain and substantiate methods of solution error estimates for a wider class of problems.

Acknowledgments

The research was supported by Russian Science Foundation (project No. 21-71- 30005), https://rscf.ru/en/project/21-71-30005/.

This article is one of the last articles containing research results that the authors obtained in collaboration with Alexander G. Biryukov, who passed away in March 2023.

References

A. G. Biryukov and A. V. Chernov. On numerical estimates of errors in solving convex optimization problems. *Communications in Computer and Information Science*, 1514, 2021.

A. G. Biryukov and A. I. Grinevich. Multiprecision arithmetic as a guarantee of the required accuracy of numerical calculation results. *TRUDY MIPT*, 4(3):171–180, 2012.

A. G. Biryukov and A. I. Grinevich. Methods for rounding errors estimation in the solution of numerical problems using arbitrary precision floating point calculations. *TRUDY MIPT*, 5(2): 160–174, 2013.

S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015.

A. V. Gasnikov. *Modern Numerical Optimization Methods*. MIPT, Moscow, 2nd edn edition, 2018.

P. E. Gill, W. V. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 1981.

A. S. Nemirovski and D. B. Iudin. *Problem Complexity and Method Efficiency in Optimization*. Nauka, Moscow, 1979.

Yu. E. Nesterov. *Introduction to Convex Optimization*. MTsNMO, Moscow, 2018.

B. T. Polyak. *Introduction to Optimization*. Nauka, Moscow, 2021.

A Proof of Lemma 1

The error of the solution at the step i can be written in the following form:

$$\begin{aligned} f_i - f^* = &\ f_i - f_{i+1} + f_{i+1} - f_{i+2} + f_{i+2} - f_{i+3} + \ldots + f_{N-1} - f_N + f_N - f^* = \\ = &\ \delta_i + \delta_{i+1} + \ldots + \delta_{N-1} + \overline{\delta}_N = \\ = &\ \delta_i \cdot (1 + g_i + g_i g_{i+1} + \ldots + g_i g_{i+1} \ldots g_{N-2}) + \overline{\delta}_N \leq \\ \leq &\ \delta_i \cdot (1 + \hat{g} + ... + \hat{g}^{N-2}) + \overline{\delta}_N = \\ = &\ \delta_i \cdot \tfrac{1-\hat{g}^{N-1}}{1-\hat{g}} + \overline{\delta}_N. \end{aligned}$$

Similarly, for a point $\boldsymbol{x}^N$ you can get an estimate δ_N after m steps:

$$\overline{\delta}_N = \delta_N \cdot \frac{1-\hat{g}^{N+m-1}}{1-\hat{g}} + \overline{\delta}_{N+m} \leq \delta_N \cdot \frac{1}{1-\hat{g}} + \overline{\delta}_{N+m}.$$

Assuming that for $m \to \infty$ at $\delta_{n+m} \to 0$ we obtain

$$\overline{\delta}_N \leq \delta_N \cdot \frac{1}{1-\hat{g}} = \frac{f_N - f_{N+1}}{1-\hat{g}}.$$

Consequently

$$\overline{\delta}_N \leq \delta_N \cdot \frac{1}{1-\hat{g}} = \frac{\delta_i g_i g_{i+1} \ldots g_{N-1}}{1-\hat{g}} \leq \frac{\delta_i \hat{g}^{N-1}}{1-\hat{g}}$$

Thus

$$\hat{\delta}_i = f_i - f^* \leq \frac{\delta_i}{1-\hat{g}} = \frac{f_i - f_{i+1}}{1-\hat{g}}.$$

B Proof of Theorem 1

Let $\hat{\delta}_i = f_i - f^*$. It is obvious that for all i and j the equalities $f_i - f_j = \hat{\delta}_i - \hat{\delta}$ and $f_i + f_l - 2f_j = \hat{\delta}_i + \hat{\delta}_l - 2\hat{\delta}_j$ are valid. Thus (7) can be written in the form:

$$\hat{\delta}_i \leq \frac{(\hat{\delta}_i - \hat{\delta}_j)^2}{\hat{\delta}_i + \hat{\delta}_l - 2\hat{\delta}_j}$$

Using the assumption (6) we can rewrite the inequality above as: $\hat{\delta}_i(\hat{\delta}_i + \hat{\delta}_l - 2\hat{\delta}_j) \leq (\hat{\delta}_i - \hat{\delta}_j)^2$ and simplifying it we obtain the inequality: $\hat{\delta}_i \cdot \hat{\delta}_j \leq \hat{\delta}_j^2$. Thus, using $\hat{\delta}_i = f_i - f^*$, we obtain the inequality (8) from the inequality above. The proof in the other direction is obvious.

C Proof of Theorem 2

Consider the geometric interpretation of (7) in Figure 2.

Denote its right part as a function of the corresponding parameters: $\varphi(f_i, f_j, f_l) = \frac{(f_i - f_j)^2}{f_i + f_l - 2f_j}$. Let also $f_c = (f_i + f_l)/2$. Note that when $f_j = f_c$, the denominator of $f_i + f_l - 2f_j$ is 0, and the value of the function φ is ∞. Denote by f_0 the value f_j at which $\varphi(f_i, f_0, f_l) = f_i - f^*$, i.e. there is an exact estimate (7).

We can also conclude that the region of admissible values for f_j is the half-interval $[f_l, f_c)$. The sought upper estimate is observed at $f_j \in [f_0, f_c)$, and the lower estimate is obtained at $f_j \in [f_l; f_0]$.

Let us clarify the obtained conditions. Denote $F_1 = f_j^2 - f_i \cdot f_l$, $F_2 = f^* \cdot (f_i + f_l - 2f_j)$. Then (8) implies that the inequality $F_1 + F_2 \geq 0$ is satisfied, or (7) is satisfied for $F_1 + F_2 \geq 0$.

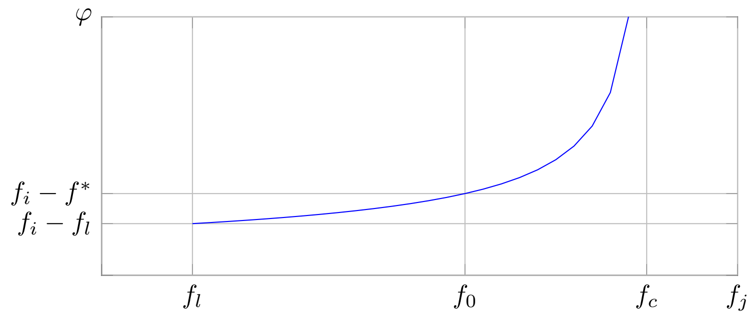

Figure 2: Geometric illustration for $\varphi(f_i, f_j, f_l)$.

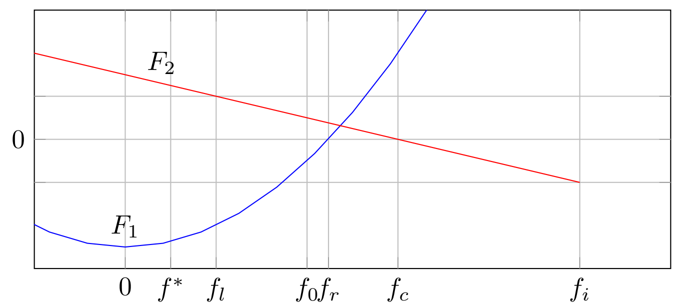

Figure 3: $f^* > 0$ case

Note also that $F_1 + F_2 < 0$ at $f_j \in [f_l, f_0)$ and $F_1 + F_2 \geq 0$ at $f_j \in [f_0, f_c)$.

In the inequality (8), the value of f^* can be equal to zero, greater than zero, or less than zero. If $f^* = 0$, then the error $f_i - f^*$ is the value of f_i itself, i.e., the inconsistency of the condition $f(\boldsymbol{x}) = 0$. Other variants need to be investigated separately.

Consider the case when $f^* > 0$. Consider f_i and f_l fixed. It is necessary to specify such f_j that $f_l < f_j < f_c$ and for which the condition (6) is satisfied. Figure 3 shows the graphs of the functions F_1 and F_2. Denote: $f_\Gamma = \sqrt{f_i f_l}$, $f_c = \frac{f_i + f_l}{2}$. (Obviously, $f_\Gamma \leq f_c$). The feasible region for f_j is the half-interval $[f_0, f_c)$, where f_0 is unknown. If $f_j = f_\Gamma$, then the estimate (7) has the form: $f_i - f^* \leq f_i$ and is always satisfied, being rough at close values of f_i and f^*. In this case, the search for the best value of $f_j \in [f_0, f_c)$ can be performed at $f_j \in [f_0, f_\Gamma]$.

Consider the case $f^* < 0$ and assume that $f_i < 0$. Figure 4 shows the graphs of the functions F_1 and F_2. In this case: $f^* < f_l < f_0 < f_c < f_\Gamma < f_i$, where $f_\Gamma = -\sqrt{f_i \cdot f_l}$. At the point $f_j = f_0$ the condition $F_1 + F_2 = 0$ is satisfied, and the point $f_j = f_\Gamma$ is invalid because in this case $f_i + f_l + 2\sqrt{f_i f_l} < 0$. The point $f_j = f_c$ is also invalid and $f_j \in [f_0, f_c)$.

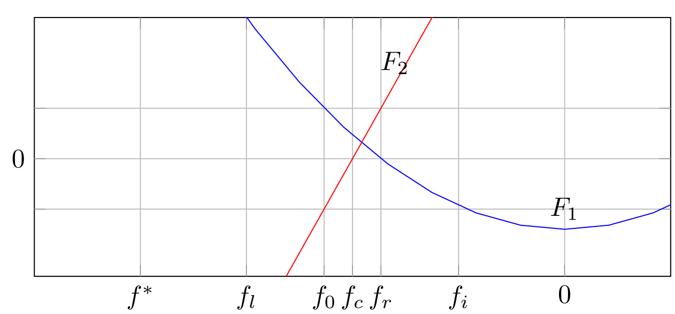

Figure 4: Geometric illustration. The case $f_i < 0$

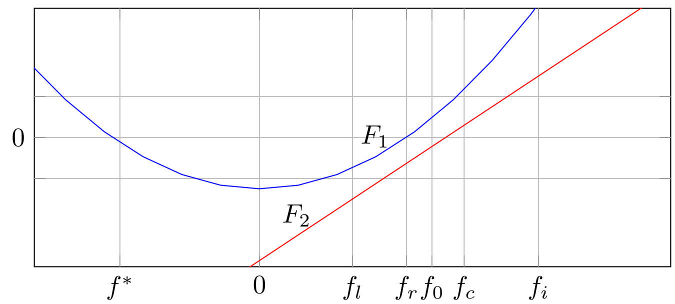

Figure 5: Geometric illustration. The case $f^* < 0$ and $f_i, f_l \geq 0$

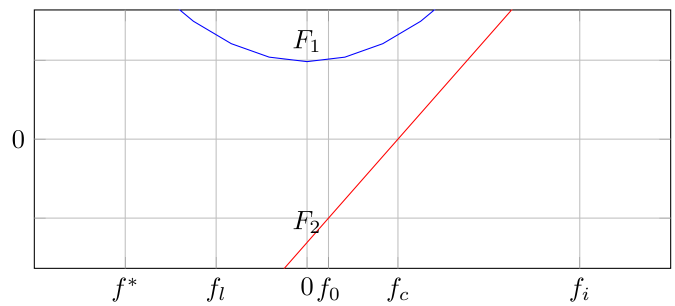

Figure 6: $f^* < 0$, $f_i > 0$, $f_l < 0$

Now consider the case $f^* < 0$ and $f_i, f_l \geq 0$ (see Figure 5). In this case $f_\Gamma = \sqrt{f_l f_i}$. Then the point $f_j = f_\Gamma$ corresponds to the case $f^* > 0$, but the estimate (6) is incorrect. The point f_c turns out to be invalid for (7). The value $f_j = f_0$ corresponds to the exact estimate (7).

Now let $f^* < 0$, $f_i > 0$, $f_l < 0$ (see Figure 6). Then $f_j \in [f_0, f_c)$, and the value $f_j = f_0$ corresponds to the exact estimate (7).

Consider now the estimate (8) equivalent to (7):

$$F_1(f_j) + F_2(f_j) = f^*(f_i + f_l - 2f_j) + f_j^2 - f_i f_l \geq 0.$$

The solution error estimates are accurate if $F_1(\cdot) + F_2(\cdot) = 0$. In this case: $f_j^2 - 2f_j f^* + f^*(f_i + f_l) - f_i f_l = 0$. Then:

$$f_j = f^* + \sqrt{(f^*)^2 - f^*(f_i + f_l) + f_i f_l} = f^* + \sqrt{(f_i - f^*)(f_l - f^*)}. \tag{25}$$

Equality (25) determines the value of the function $f_j = f_0$, at which there is an exact estimate of the solution error. The estimate (8) is equivalent to the inequality

$$(f_j - f^*)^2 - (f_i - f^*)(f_l - f^*) \geq 0,$$

and the equality (25) is equivalent to the equality

$$(f_j - f^*)^2 - (f_i - f^*)(f_l - f^*) = 0.$$

D Proof of Treorem 3

A geometric illustration of the solution error estimation with $\underline{f^*}$ and $\overline{f^*}$ is shown in Figure 7. Based on Figure 7 we can also get some lower estimates, for example: $f_i - \underline{f^*} > f_i - f_N > f_i - f_l > f_i - f_j$, etc.

Thus, the values $\underline{f^*}$ and $\overline{f^*}$ are estimates from below and above the values f^* and f_i, i.e.:

$$\underline{f^*} < f^* < \overline{f^*}, \quad \underline{f^*} < f_i < \overline{f^*}, \tag{26}$$

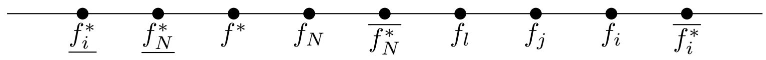

Figure 7: Geometric interpretation of the solution error estimate at the point $f = f_i$

where the numbers $\underline{f^*}$, f^*, f_i have equal $m_i + 1$ digits when $f^* > 0$ and m_i digits when $f^* < 0$, and the numbers $\overline{f^*}$, f^*, f_i have m_i digits when $f^* > 0$ and $m_i + 1$ digits when $f^* < 0$.

It follows from the inequalities (26) that

$$f^* - \underline{f^*} < 10^{t-m_i}, \quad \overline{f^*} - f^* < 10^{t-m_i}, \quad f_i - \underline{f^*} < 10^{t-m_i}, \\ \overline{f^*} - f_i < 10^{t-m_i}, \quad \overline{f^*} - \underline{f^*} < 10^{t-m_i}. \tag{27}$$

In addition, as stated above, $f_i - f^* \leq f_i - \underline{f^*}$. Given (27), we obtain the desired estimate of the solution error:

$$f_i - f^* < 10^{t-m_i}. \tag{28}$$

Assume that there exist matching digits in the representation of value f_s, $i \leq s \leq N-1$; f_N is the last value in the iterative process of solving the problem, with which numbers f_s and $m_s + 1$ are compared, where $m_s + 1$ is the number of matching digits in numbers f_s and f_N. Then we can construct a table of numbers f_s, $s = \overline{i, N-1}$ and select in it the digits coinciding with f_N. Then, using (28), we obtain an estimate of the solution error for values $s = \overline{i, N-1}$.

E EXPERIMENT RESULTS

#	$f(x^k)$	ε_1	ε_2
0	627387.11631838745289500864	$1 \cdot 10^6$	509894.77
1	478900.08308849168197638259	$1 \cdot 10^6$	361407.74
2	243348.13069838833605295973	$1 \cdot 10^6$	125855.78
3	216169.69227938405276040735	$1 \cdot 10^6$	98677.35
8	**1**31996.90602183602538406045	$1 \cdot 10^5$	14504.56
13	**1**20121.26164074089282338418	$1 \cdot 10^5$	2628.92
18	**117**854.15903353102929779504	$1 \cdot 10^3$	361.81
23	**117**530.92508592090817802585	$1 \cdot 10^3$	38.58
28	**11749**6.62956020062317238274	$1 \cdot 10^1$	4.28
33	**117492**.62475777944761005597	$1 \cdot 10^0$	0.28
38	**117492.3**7875336652483850275	$1 \cdot 10^{-1}$	0.033
43	**117492.3**5603159960543316724	$1 \cdot 10^{-1}$	0.01
48	**117492.345**99695254306091291	$1 \cdot 10^{-3}$	0.00024
53	**117492.3457**6246430599323281	$1 \cdot 10^{-4}$	$1.0 \cdot 10^{-5}$
58	**117492.345752**32570898953809	$1 \cdot 10^{-6}$	$9.3 \cdot 10^{-8}$
63	**117492.34575223**507416076358	$1 \cdot 10^{-8}$	$2.3 \cdot 10^{-9}$
68	**117492.345752232**85267005296	$1 \cdot 10^{-9}$	$9.5 \cdot 10^{-11}$
73	**117492.34575223275**870822855	$1 \cdot 10^{-11}$	$8.1 \cdot 10^{-13}$
78	**117492.345752232757**95044578	$1 \cdot 10^{-12}$	$5.6 \cdot 10^{-14}$
83	**117492.34575223275789**716425	$1 \cdot 10^{-14}$	$3.1 \cdot 10^{-15}$
88	**117492.345752232757894**10472	$1 \cdot 10^{-15}$	$3.8 \cdot 10^{-17}$
93	**117492.34575223275789406**717	$1 \cdot 10^{-17}$	$4.4 \cdot 10^{-19}$
98	**117492.3457522327578940667**7	$1 \cdot 10^{-19}$	$2.6 \cdot 10^{-20}$
99	**117492.3457522327578940667**6	$1 \cdot 10^{-19}$	$2.9 \cdot 10^{-20}$
100	**117492.3457522327578940667**6	$1 \cdot 10^{-19}$	$1.8 \cdot 10^{-21}$
103	**117492.34575223275789406675**	$1 \cdot 10^{-20}$	$4.3 \cdot 10^{-22}$

Table 4: Smooth unconstrained convex optimization problem output. Function.

#	$\|x^k\|$	ε_3	ε_4
0	001.2728880402482	$1 \cdot 10^3$	227.031
1	019.9048908572319	$1 \cdot 10^3$	208.4
2	109.9550314835497	$1 \cdot 10^3$	118.35
3	124.6780539085920	$1 \cdot 10^3$	103.63
8	184.1014031017751	$1 \cdot 10^3$	44.2
13	**2**12.8510755796911	$1 \cdot 10^2$	15.45
18	**22**4.8625629288455	$1 \cdot 10^1$	3.44
23	**22**7.5905753548338	$1 \cdot 10^1$	0.71
28	**228**.1115512836949	$1 \cdot 10^0$	0.19
33	**228**.1781371850296	$1 \cdot 10^0$	0.13
38	**228**.2245030510902	$1 \cdot 10^0$	0.079
43	**228**.2592315075667	$1 \cdot 10^0$	0.044
48	**228.30**15301041376	$1 \cdot 10^{-2}$	0.002
53	**228.303**2722587835	$1 \cdot 10^{-3}$	0.00029
58	**228.30356**07681773	$1 \cdot 10^{-5}$	$3.3 \cdot 10^{-6}$
63	**228.3035**764408335	$1 \cdot 10^{-4}$	$1.2 \cdot 10^{-5}$
68	**228.30356**60668867	$1 \cdot 10^{-5}$	$2.0 \cdot 10^{-6}$
73	**228.303564**3821351	$1 \cdot 10^{-6}$	$2.9 \cdot 10^{-7}$
78	**228.303564**1638766	$1 \cdot 10^{-6}$	$7.1 \cdot 10^{-8}$
83	**228.303564**1168132	$1 \cdot 10^{-6}$	$2.4 \cdot 10^{-8}$
88	**228.30356409**41619	$1 \cdot 10^{-8}$	$7.0 \cdot 10^{-10}$
93	**228.303564093**7173	$1 \cdot 10^{-9}$	$2.3 \cdot 10^{-10}$
98	**228.3035640935**230	$1 \cdot 10^{-10}$	$2.9 \cdot 10^{-11}$
99	**228.30356409351**50	$1 \cdot 10^{-11}$	$4.0 \cdot 10^{-13}$
100	**228.30356409351**33	$1 \cdot 10^{-11}$	$1.2 \cdot 10^{-12}$
103	**228.303564093514**2	$1 \cdot 10^{-12}$	$2.0 \cdot 10^{-13}$

Table 5: Smooth unconstrained convex optimization problem output. Points.

#	$f(x^k)$	ε_1	ε_2
1	123.2893499883321	$1 \cdot 10^3$	123.29
2	52.09959143810754	$1 \cdot 10^2$	52.1
3	30.72335415840225	$1 \cdot 10^2$	30.72
38	**0.**369051970412854	$1 \cdot 10^0$	0.37
73	**0.0**27874590126960	$1 \cdot 10^{-1}$	0.028
108	**0.00**4713454334457	$1 \cdot 10^{-2}$	0.0047
143	**0.000**894394670003	$1 \cdot 10^{-3}$	0.00089
178	**0.000**173439884020	$1 \cdot 10^{-3}$	0.00017
213	**0.0000**33762594077	$1 \cdot 10^{-4}$	$3.4 \cdot 10^{-5}$
248	**0.00000**6576774483	$1 \cdot 10^{-5}$	$6.6 \cdot 10^{-6}$
283	**0.00000**1281269557	$1 \cdot 10^{-5}$	$1.3 \cdot 10^{-6}$
318	**0.000000**249618530	$1 \cdot 10^{-6}$	$2.5 \cdot 10^{-7}$
353	**0.0000000**48631167	$1 \cdot 10^{-7}$	$4.9 \cdot 10^{-8}$
388	**0.00000000**9474428	$1 \cdot 10^{-8}$	$9.5 \cdot 10^{-9}$
423	**0.00000000**1845832	$1 \cdot 10^{-8}$	$1.9 \cdot 10^{-9}$
458	**0.000000000**359614	$1 \cdot 10^{-9}$	$3.7 \cdot 10^{-10}$
493	**0.0000000000**70065	$1 \cdot 10^{-10}$	$7.4 \cdot 10^{-11}$
528	**0.0000000000**13655	$1 \cdot 10^{-10}$	$1.5 \cdot 10^{-11}$
563	**0.00000000000**2665	$1 \cdot 10^{-11}$	$3.6 \cdot 10^{-12}$
598	**0.000000000000**524	$1 \cdot 10^{-12}$	$1.4 \cdot 10^{-12}$

Table 6: Non-smooth unconstrained convex optimization problem output.

#	$\|x^k\|$	ε_3	ε_4
1	3.50909086234578142	$1 \cdot 10^{1}$	1.27
2	3.25249619214075886	$1 \cdot 10^{1}$	1.016
3	3.05669444787201110	$1 \cdot 10^{1}$	0.82
38	**2.2**5555433276413277	$1 \cdot 10^{-1}$	0.019
73	**2.23**741141399673527	$1 \cdot 10^{-2}$	0.0013
108	**2.236**31507477184337	$1 \cdot 10^{-3}$	0.00025
143	**2.236**11497218344222	$1 \cdot 10^{-3}$	$4.7 \cdot 10^{-5}$
178	**2.2360**7721889738496	$1 \cdot 10^{-4}$	$9.2 \cdot 10^{-6}$
213	**2.23606**975667863855	$1 \cdot 10^{-5}$	$1.8 \cdot 10^{-6}$
248	**2.23606**832810608537	$1 \cdot 10^{-5}$	$3.5 \cdot 10^{-7}$
283	**2.23606**804502447861	$1 \cdot 10^{-5}$	$6.8 \cdot 10^{-8}$
318	**2.2360679**9080709652	$1 \cdot 10^{-7}$	$1.3 \cdot 10^{-8}$
353	**2.2360679**8006272692	$1 \cdot 10^{-7}$	$2.6 \cdot 10^{-9}$
388	**2.23606797**800487733	$1 \cdot 10^{-8}$	$5.1 \cdot 10^{-10}$
423	**2.236067977**59706814	$1 \cdot 10^{-9}$	$9.7 \cdot 10^{-11}$
458	**2.236067977**51896115	$1 \cdot 10^{-9}$	$1.9 \cdot 10^{-11}$
493	**2.236067977**50348249	$1 \cdot 10^{-9}$	$3.7 \cdot 10^{-12}$
528	**2.236067977**50051789	$1 \cdot 10^{-9}$	$7.4 \cdot 10^{-13}$
563	**2.236067977499**93039	$1 \cdot 10^{-12}$	$1.4 \cdot 10^{-13}$
598	**2.236067977499**81787	$1 \cdot 10^{-12}$	$2.9 \cdot 10^{-14}$
633	**2.23606797749979**557	$1 \cdot 10^{-14}$	$4.9 \cdot 10^{-15}$
650	**2.236067977499792**68	$1 \cdot 10^{-15}$	$2.7 \cdot 10^{-15}$
655	**2.236067977499792**15	$1 \cdot 10^{-15}$	$3.3 \cdot 10^{-17}$

Table 7: Non-smooth unconstrained convex optimization problem output.

#	$f(x^k)$	ε_1	ε_2
1	**5**75692.05137578458	$1 \cdot 10^5$	10278.17
2	**56**6816.38933322136	$1 \cdot 10^4$	1402.51
3	**565**802.33314266431	$1 \cdot 10^3$	388.45
17	**565413.8**8509552008	$1 \cdot 10^{-1}$	0.0064
44	**565413.8**8183280189	$1 \cdot 10^{-1}$	0.0032
74	**565413.8**8013226316	$1 \cdot 10^{-1}$	0.0015
104	**565413.87**930412149	$1 \cdot 10^{-2}$	0.00064
134	**565413.878**95278421	$1 \cdot 10^{-3}$	0.00028
164	**565413.878**80059818	$1 \cdot 10^{-3}$	0.00013
194	**565413.878**73276216	$1 \cdot 10^{-3}$	$6.4 \cdot 10^{-5}$
224	**565413.878**70137843	$1 \cdot 10^{-3}$	$3.3 \cdot 10^{-5}$
254	**565413.8786**8619214	$1 \cdot 10^{-4}$	$1.8 \cdot 10^{-5}$
284	**565413.8786**7846975	$1 \cdot 10^{-4}$	$9.9 \cdot 10^{-6}$
314	**565413.8786**7434236	$1 \cdot 10^{-4}$	$5.8 \cdot 10^{-6}$
344	**565413.8786**7203408	$1 \cdot 10^{-4}$	$3.5 \cdot 10^{-6}$
374	**565413.8786**7069341	$1 \cdot 10^{-4}$	$2.1 \cdot 10^{-6}$
404	**565413.87866**989163	$1 \cdot 10^{-5}$	$1.3 \cdot 10^{-6}$
434	**565413.87866**940176	$1 \cdot 10^{-5}$	$8.3 \cdot 10^{-7}$
464	**565413.87866**909791	$1 \cdot 10^{-5}$	$5.2 \cdot 10^{-7}$
494	**565413.878668**90749	$1 \cdot 10^{-6}$	$3.3 \cdot 10^{-7}$
524	**565413.878668**78732	$1 \cdot 10^{-6}$	$2.1 \cdot 10^{-7}$
554	**565413.878668**71111	$1 \cdot 10^{-6}$	$1.3 \cdot 10^{-7}$
584	**565413.878668**66262	$1 \cdot 10^{-6}$	$8.0 \cdot 10^{-8}$
614	**565413.878668**63168	$1 \cdot 10^{-6}$	$4.8 \cdot 10^{-8}$
644	**565413.878668**61191	$1 \cdot 10^{-6}$	$2.8 \cdot 10^{-8}$
674	**565413.8786685**9924	$1 \cdot 10^{-7}$	$1.5 \cdot 10^{-8}$
704	**565413.8786685**9112	$1 \cdot 10^{-7}$	$6.1 \cdot 10^{-9}$
726	**565413.878668858**706	$1 \cdot 10^{-8}$	$1.7 \cdot 10^{-9}$
730	**565413.8786685866**46	$1 \cdot 10^{-9}$	$1.1 \cdot 10^{-9}$
731	**565413.8786685866**31	$1 \cdot 10^{-9}$	$1.1 \cdot 10^{-9}$

Table 8: Smooth constrained convex optimization problem output. Function.

#	$\|x^k\|$	ε_3	ε_4
1	20.119061037035	$1 \cdot 10^{2}$	14.73
2	29.593519102641	$1 \cdot 10^{2}$	5.25
3	**3**2.124366408076	$1 \cdot 10^{1}$	2.71
17	**34.8**21131179517	$1 \cdot 10^{-1}$	0.011
44	**34.8318**35098335	$1 \cdot 10^{-4}$	$5.8 \cdot 10^{-6}$
74	**34.8318**33484930	$1 \cdot 10^{-4}$	$4.2 \cdot 10^{-6}$
104	**34.8318**32234587	$1 \cdot 10^{-4}$	$2.9 \cdot 10^{-6}$
134	**34.8318**31284485	$1 \cdot 10^{-4}$	$2.0 \cdot 10^{-6}$
164	**34.8318**30616449	$1 \cdot 10^{-4}$	$1.3 \cdot 10^{-6}$
194	**34.8318**30165123	$1 \cdot 10^{-4}$	$8.5 \cdot 10^{-7}$
224	**34.831829**866973	$1 \cdot 10^{-6}$	$5.5 \cdot 10^{-7}$
254	**34.831829**672599	$1 \cdot 10^{-6}$	$3.5 \cdot 10^{-7}$
284	**34.831829**546887	$1 \cdot 10^{-6}$	$2.3 \cdot 10^{-7}$
314	**34.831829**465977	$1 \cdot 10^{-6}$	$1.5 \cdot 10^{-7}$
344	**34.831829**414054	$1 \cdot 10^{-6}$	$9.3 \cdot 10^{-8}$
374	**34.8318293**80792	$1 \cdot 10^{-7}$	$6.0 \cdot 10^{-8}$
404	**34.8318293**59502	$1 \cdot 10^{-7}$	$3.8 \cdot 10^{-8}$
434	**34.8318293**45882	$1 \cdot 10^{-7}$	$2.5 \cdot 10^{-8}$
464	**34.8318293**37168	$1 \cdot 10^{-7}$	$1.6 \cdot 10^{-8}$
494	**34.8318293**31591	$1 \cdot 10^{-7}$	$1.0 \cdot 10^{-8}$
524	**34.83182932**8019	$1 \cdot 10^{-8}$	$6.5 \cdot 10^{-9}$
554	**34.83182932**5731	$1 \cdot 10^{-8}$	$4.1 \cdot 10^{-9}$
584	**34.83182932**4262	$1 \cdot 10^{-8}$	$2.6 \cdot 10^{-9}$
614	**34.83182932**3319	$1 \cdot 10^{-8}$	$1.6 \cdot 10^{-9}$
644	**34.83182932**2711	$1 \cdot 10^{-8}$	$9.5 \cdot 10^{-10}$
674	**34.83182932**2319	$1 \cdot 10^{-8}$	$5.3 \cdot 10^{-10}$
704	**34.83182932**2066	$1 \cdot 10^{-8}$	$2.5 \cdot 10^{-10}$
726	**34.8318293219**39	$1 \cdot 10^{-10}$	$2.7 \cdot 10^{-10}$
730	**34.83182932191**9	$1 \cdot 10^{-11}$	$2.9 \cdot 10^{-11}$
731	**34.83182932191**5	$1 \cdot 10^{-11}$	$1.0 \cdot 10^{-11}$
732	**34.83182932191**0	$1 \cdot 10^{-11}$	$2.2 \cdot 10^{-12}$

Table 9: Smooth constrained convex optimization problem output. Points.