STUDIA Z AUTOMATYKI I INFORMATYKI
TOM 38 – 2013

Dariusz Horla*

PERFORMANCE EVALUATION OF ITERATIVE METHODS TO UNCONSTRAINED SINGLE VARIABLE MINIMIZATION PROBLEMS

1. INTRODUCTION

Linear programming problems can be solved with high precision using reliable and fast IPM (interior-point methods) algorithms. There are optimization tasks, however, that do not meet linearity requirement, dominating in real-life. If the decision variables are numbers of unknown values and objective function is nonlinear, the problem falls in the category of unconstrained nonlinear programming.

The methods presented below concentrate on minimizing objective function which is not to stringent requirement, since all results can be obtained for maximization problem as well. Among general nonlinear programming problems one can identify dedicated and effective methods or special structures of the tasks.

There is a variety of approaches to solving nonlinear programs, thus there is no method to solve the problems in general. As in the case of unconstrained minimization problems, one can divide the methods available to classes, such as zero-, first- and second-order algorithms.

In some methods, necessary and sufficient optimality conditions are used, leading to obtaining the algorithms described in this paper.

The paper concentrates on presenting three classes of algorithms with information concerning efficiency of the algorithms given, defined as mean number of iterations necessary to reach the minimizer with a prescribed tolerance. The conclusions can be helpful in selecting the algorithm dedicated to the problem to be solved.

2. BASIC STEPS IN NUMERICAL METHODS TO SOLVE UNCONSTRAINED NONLINEAR PROGRAMS

Analytical methods based on solving necessary and sufficient conditions for nonlinear programs have limited applicability in practice. The possible reasons for the latter, and subsequently, possible reasons to resort to iterative methods are the cases when constraint or objective functions:

- are of complicated form, what may result in difficulties with solving the problems analytically,
- are not explicit functions of the decision variables,
- do not satisfy differentiability conditions or are discontinuous.

*Poznań University of Technology, Institute of Control and Information Engineering, Department of Control and Robotics, Piotrowo 3a Str., 60-965 Poznań, e-mail: Dariusz.Horla@put.poznan.pl

Numerical methods of solving optimization problems lead to iterative approaches, where an approximate solution is sought in consecutive steps, with improved solutions expected to be better than the current ones (objective function should decrease).

In the case of $\underline{x} \in \mathscr{R}^n$, one can formulate a general algorithm for such methods:

- for initial guess $\underline{x}^{(0)}$ put $k = 0$,
- find the direction $\underline{d}^{(k)}$ in which objective function value is improved,
- choose the step length α_k from the point $\underline{x}^{(k)}$ in the direction $\underline{d}^{(k)}$ to the point $\underline{x}^{(k+1)}$, i.e. an improved solution to the problem,
- compute the improved solution $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \alpha_k \underline{d}^{(k)}$ to the given problem,
- check stopping criterion, what should answer if the improved solution is satisfactory and whether it should be amended or not; if yes, put $k := k + 1$ and start the algorithm over.

3. STOPPING CRITERIA

And important question in numerical algorithms is if the improved solution is satisfactory, i.e. it is viable to set the stopping criterion to the algorithm. It is only in the limited number of cases that the solution to the optimization problem can be found in a limited number of steps. One has to choose the rules to verify if the approximated solution is acceptable.

Sample stopping criteria for arbitrarily chosen threshold levels include:

- theoretical criteria

$$|f(\underline{x}^{(k)}) - f(\underline{x}^*)| < \epsilon_1, \quad \|\underline{x}^{(k)} - \underline{x}^*\| < \epsilon_2 \,,$$

- approximate stationary solution to the problem

$$\|\nabla f(\underline{x}^{(k)})\| < \epsilon$$

- practical criteria

$$|x_i^{(k+1)} - x_i^{(k)}| \leqslant \epsilon_i \quad \text{or} \quad \|\underline{x}^{(k+1)} - \underline{x}^{(k)}\| \leqslant \epsilon$$

or

$$|f(\underline{x}^{(k+1)}) - f(\underline{x}^{(k)})| < \epsilon_1 \,.$$

4. CHARACTERISTICS OF THE NUMERICAL ALGORITHM

When using numerical algorithms one has to take the following features under consideration:

- convergence

  algorithm is convergent if there exists the limit to the sequence of approximate solutions obtained from its consecutive steps,

- initial solution

  effectiveness of the algorithm depends on the choice of initial solution, as, e.g., algorithm may stuck in a local minimum of a multimodal function and in such a case it should be started many times from different initial points to find the global minimum,
- stopping criterion

  in order to stop the iterative algorithm, one should check if the improved approximate solution changes from one iteration to another, as the speed of the algorithm, i.e. time that it takes to execute it, is an another factor, apart from the accuracy of the obtained solution.

5. CLASSIFICATION OF ITERATIVE ALGORITHMS OF SINGLE VARIABLE

The feature that allows one to differ the algorithms is the order of derivatives used by them, and as such we have:

- zero-order algorithms

  used to compute function values at specific points only, especially when objective function has complicated form or cannot be explicitly given,
- gradient algorithms (first-order)

  used when gradient of the objective function is available,
- second-order algorithms

  used when information about second derivatives are available.

6. ZERO-ORDER METHODS

6.1. HOW TO SEARCH FOR THE MINIMIZER OF THE UNIMODAL FUNCTION?

The zero-order algorithms are used to seek for the minimum of the function $f : \mathscr{R} \to \mathscr{R}$ in the interval $\left[x^{(0^-)},\, x^{(0^+)}\right]$ with $x^{(0^+)} > x^{(0^-)}$. The only property that objective function f must posses is unimodularity, i.e. it should have a single minimum in the given set. The examples of unimodal functions are shown in Figure 1.

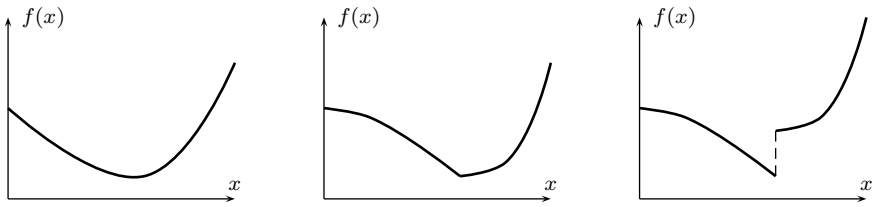

Fig. 1. Unimodal functions

The methods to be presented are based on computing values of the objective function in various points of the set $\left[x^{(0^-)},\, x^{(0^+)}\right]$ in such a way to enable one to stipulate the minimizer

of f in the smaller number of steps with successive reduction of the interval in which it is included in.

Let a unimodal function be given in the uncertainty range $\left[x^{(0^-)},\, x^{(0^+)}\right]$. If the value of f is computed at a single point only, there is no way to answer how to reduce its length to find the minimizer of f. The value of f should therefore be computed in two intermediate points, with symmetrical reduction, respecting the condition

$$x^{(1^-)} - x^{(0^-)} = x^{(1^+)} - x^{(0^+)} = \rho(x^{(0^+)} - x^{(0^-)})\,,$$

where $\rho < \frac{1}{2}$, as shown in Figure 2.

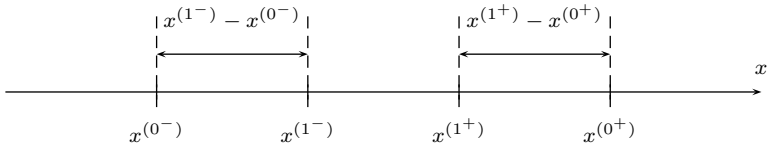

Fig. 2. Value of the objective function at two intermediate points

Having computed the values of f at intermediate points, the minimizer is in the interval $\left[x^{(0^-)},\, x^{(1^+)}\right]$ when it holds that $f(x^{(1^-)}) < f(x^{(1^+)})$, and for the opposite case, i.e. $f(x^{(1^-)}) \geqslant f(x^{(1^+)})$, it is in the range $\left[x^{(1^-)},\, x^{(0^+)}\right]$, as it is shown in the Figure 3.

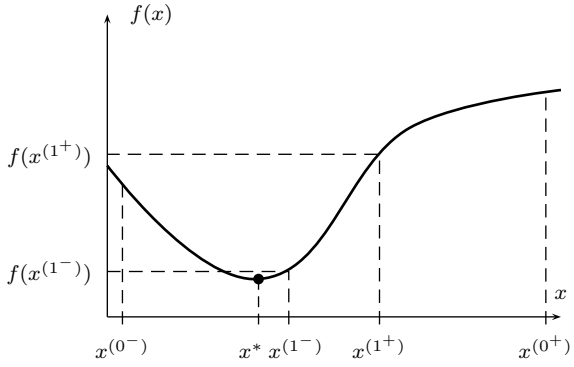

Fig. 3. The case of $f(x^{(1^-)}) < f(x^{(1^+)})$, with the minimizer $x^* \in \left[x^{(0^-)},\, x^{(1^+)}\right]$

The next iteration should be started from the reduced interval, and the procedure is repeated all over again by stipulating two new points, i.e. $x^{(2^-)}$, $x^{(2^+)}$ and comparing objective function values. From this point of view, the zero-order methods are obviously called branch and bound methods. Furthermore, the function f does not have to be neither differentiable, nor continuous in $\left[x^{(0^-)},\, x^{(0^+)}\right]$.

To summarize, for an initial set $x \in \mathscr{D}^{(0)} = \left[x^{(0^-)},\, x^{(0^+)}\right]$ a general branch and bound algorithms is as follows:

- compute a minimal number N of iterations so that the difference between exact x^* and approximate solutions $\hat{x}^*$ (assumed to lie in the center of $\mathscr{D}^{(N)}$) does not exceed a

given tolerance ϵ, i.e.

$$|x^* - \hat{x}^*| \leqslant \epsilon(x^{(0^+)} - x^{(0^-)}),$$

- for $k = 1, \ldots, N$:
  - set two new intermediate points $\hat{x}^{(k^-)}$, $\hat{x}^{(k^+)}$ ($\hat{x}^{(k^-)} < \hat{x}^{(k^+)}$, $\left\{\hat{x}^{(k^-)}, \hat{x}^{(k^+)}\right\} \in \mathscr{D}^{(k-1)}$) in the interval $\mathscr{D}^{(k-1)}$,
  - compute the next uncertainty range $\mathscr{D}^{(k)}$ with the previously proposed points:
    a) if it holds that $f(\hat{x}^{(k^-)}) < f(\hat{x}^{(k^+)})$, set $x^{(k+1)} \in \mathscr{D}^{(k)} = \left[x^{(k-1^-)}, \hat{x}^{(k^+)}\right]$,
    b) otherwise, if $f(\hat{x}^{(k^-)}) \geqslant f(\hat{x}^{(k^+)})$, then $x^{(k+1)} \in \mathscr{D}^{(k)} = \left[\hat{x}^{(k^-)}, x^{(k-1^+)}\right]$,
  - put $k := k + 1$,
- assume that $\hat{x}^* = \frac{1}{2}(x^{(N^+)} + x^{(N^-)})$ is the approximate solutions to the given problem.

6.2. EQUAL DIVISION METHOD

If the range $\mathscr{D}^{(0)}$ is divided into smaller and equal parts with N interior points of intermediate distance $\epsilon(x^{(0^+)} - x^{(0^-)})$, where ϵ is the preferred tolerance of the approximate solution, the minimal number of iterations satisfies $N \geqslant \frac{1}{\epsilon} - 1$ with:

$$\begin{aligned}\hat{x}^{(k^-)} &= x^{(k-1^-)} + \epsilon(x^{(0^+)} - x^{(0^-)}),\\ \hat{x}^{(k^+)} &= x^{(k-1^+)} - \epsilon(x^{(0^+)} - x^{(0^-)}),\end{aligned}$$

where $\frac{1}{\epsilon}$ should be a natural number, as shown in Figure 4.

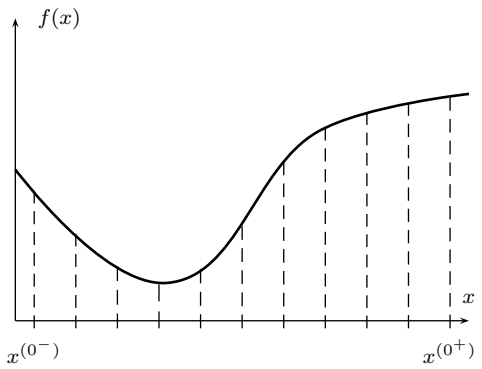

Fig. 4. Division of the initial set with tolerance of 10% ($N = 9$)

6.3. GOLDEN-SEARCH METHOD

Let for the given uncertainty interval $\mathscr{D}^{(0)}$ it hold that $f(x^{(0^-)}) < f(x^{(0^+)})$. It is evident that in such a case $x^* \in \mathscr{D}^{(1)} = [x^{(0^-)}, x^{(1^+)}]$. Since $x^{(1^-)}$ is in the new range $\mathscr{D}^{(1)}$, and the value $f(x^{(1^-)})$ is known (computed in the previous iteration), it is important to choose $x^{(1^-)}$

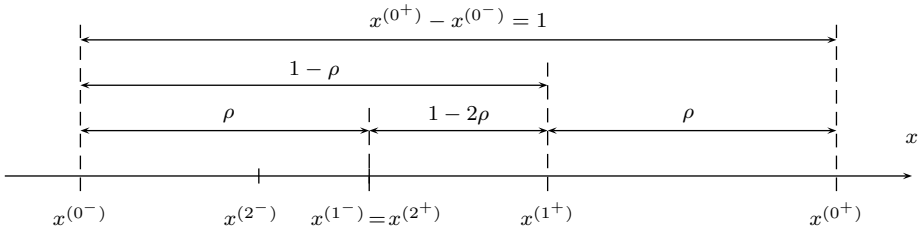

Fig. 5. How to choose ρ in order to compute value of f at a single intermediate point only

to lie at $x^{(2^+)}$. In such a case, only a single intermediate value of the function f should be computed at $x^{(2^-)}$.

In order to choose ρ that should satisfy the latter property, one should examine Figure 5.

We can assume that an initial set has a unit length, and in such a case ρ should be chosen to satisty

$$\rho\left(x^{(1^+)} - x^{(0^-)}\right) = \left(x^{(1^+)} - x^{(2^+)}\right) .$$

Since it holds hat $x^{(1^+)} - x^{(0^-)} = 1 - \rho$ and $x^{(1^+)} - x^{(2^+)} = 1 - 2\rho$, thus

$$\rho(1-\rho) = 1 - 2\rho\,,$$

and finally,

$$\rho^2 - 3\rho + 1 = 0\,.$$

The solutions to this quadratic equations are:

$$\rho_1 = \frac{3+\sqrt{5}}{2}\,, \quad \rho_2 = \frac{3-\sqrt{5}}{2}$$

and since it should hold that $\rho < \frac{1}{2}$, obviously we have $\rho = \frac{3-\sqrt{5}}{2} \cong 0.382$.

On this basis, we have a golden division of the line segment:

$$1-\rho = \frac{\sqrt{5}-1}{2} \cong 0.618\,,$$
$$\frac{\rho}{1-\rho} = \frac{1-\rho}{1}\,.$$

The length of the range is reduced by the factor of $1-\rho$ at each iteration, so after N iterations the initial uncertainty interval is reduced by factor $(1-\rho)^N$, and the minimal number of iterations satisfies $(1-\rho)^N \leqslant \epsilon$. The step of the algorithm can be described as:

$$\hat{x}^{(k^-)} = x^{(k-1^-)} + \rho(x^{(k-1^+)} - x^{(k-1^-)})\,,$$
$$\hat{x}^{(k^+)} = x^{(k-1^-)} + (1-\rho)(x^{(k-1^+)} - x^{(k-1^-)})\,.$$

6.4. GOLDEN-SEARCH METHOD (EXAMPLE)

Find the minimizer of $f(x) = 2x^5 + 3x^4 - 0.5x^3 + x^2 - 10x$ in the range $[0,\, 2]$ with tolerance $\epsilon = 0.2$.

Minimal number of iterations $N = 4$ satisfies $(1-\rho)^4 = 0.6180^4 = 0.1459 \leqslant 0.2$, and the initial interval is $\mathscr{D}^{(0)} = [0,\, 2]$.

Iteration I

$$\hat{x}^{(1^-)} = 0 + \frac{3-\sqrt{5}}{2}(2-0) = 0.7639\,,$$
$$\hat{x}^{(1^+)} = 0 + \left(1 - \frac{3-\sqrt{5}}{2}\right)(2-0) = 1.2361\,,$$
$$f(\hat{x}^{(1^-)}) = 2\cdot 0.7639^5 + 3\cdot 0.7639^4 - 0.5\cdot 0.7639^3 + 0.7639^2 - 10\cdot 0.7639 = -5.7365\,,$$
$$f(\hat{x}^{(1^+)}) = 2\cdot 1.2361^5 + 3\cdot 1.2361^4 - 0.5\cdot 1.2361^3 + 1.2361^2 - 10\cdot 1.2361 = 0.9981\,,$$
$f(\hat{x}^{(1^-)}) < f(\hat{x}^{(1^+)})$, thus $\mathscr{D}^{(1)} = [0,\, 1.2361]$, $|1.2361 - 0| = 1.2361 > |2-0|\epsilon = 0.4$.

Iteration II

$$\hat{x}^{(2^-)} = 0 + \frac{3-\sqrt{5}}{2}(1.2361-0) = 0.4721\,,$$
$$\hat{x}^{(2^+)} = 0 + \left(1 - \frac{3-\sqrt{5}}{2}\right)(1.2361-0) = 0.7640\,,$$
$$f(\hat{x}^{(2^-)}) = -4.3548\,,$$
$$f(\hat{x}^{(2^+)}) = -5.7366\,,$$
$f(\hat{x}^{(2^-)}) > f(\hat{x}^{(2^+)})$, thus $\mathscr{D}^{(2)} = [0.4721,\, 1.2361]$, $|1.2361 - 0.4721| = 0.7640 > 0.4$.

Iteration III

$$\hat{x}^{(3^-)} = 0.4721 + \frac{3-\sqrt{5}}{2}(1.2361-0.4721) = 0.7639\,,$$
$$\hat{x}^{(3^+)} = 0.4721 + \left(1 - \frac{3-\sqrt{5}}{2}\right)(1.2361-0.4721) = 0.9443\,,$$
$$f(\hat{x}^{(3^-)}) = -5.7365\,,$$
$$f(\hat{x}^{(3^+)}) = -5.0852\,,$$
$f(\hat{x}^{(3^-)}) < f(\hat{x}^{(3^+)})$, thus $\mathscr{D}^{(3)} = [0.4721,\, 0.9443]$, $|0.9443 - 0.4721| = 0.4722 > 0.4$.

Iteration IV

$$\hat{x}^{(4^-)} = 0.4721 + \frac{3-\sqrt{5}}{2}(0.9443-0.4721) = 0.6525\,,$$
$$\hat{x}^{(4^+)} = 0.4721 + \left(1 - \frac{3-\sqrt{5}}{2}\right)(0.9443-0.4721) = 0.7639\,,$$

$$f(\hat{x}^{(4^-)}) = -5.4578\,,$$
$$f(\hat{x}^{(4^+)}) = -5.7365\,,$$
$$f(\hat{x}^{(4^-)}) > f(\hat{x}^{(4^+)}),\ \text{thus}\ \mathscr{D}^{(4)} = [0.6525, 0.9443],\ |0.9443 - 0.6525| = 0.2918 < 0.4.$$

The stopping criterion is satisfied, and the approximate solution is $\hat{x}^* = \frac{0.6525+0.9443}{2} = 0.7984$, $f(\hat{x}^*) = -5.7332$ (an exact solution to the problem is $x^* = 0.7789$, $f(x^*) = -5.7410$).

6.5. DICHOTOMY METHOD

In this case, the two intermediate points are chosen symmetrically in the proximity of the middle of the current interval. The minimal number of iterations satisfies $\frac{1}{2^N} + \delta\left(1 - \frac{1}{2^N}\right) \leqslant \epsilon$, where $0 < \delta \ll 1$ is a small number defining the placement of intermediate points:

$$\hat{x}^{(k^-)} = x^{(k-1^-)} + \frac{1-\delta}{2}\left(x^{(k-1^+)} - x^{(k-1^-)}\right),$$
$$\hat{x}^{(k^+)} = x^{(k-1^-)} + \frac{1+\delta}{2}\left(x^{(k-1^+)} - x^{(k-1^-)}\right)$$

as close as possible to the center of the current range $\mathscr{D}^{(k)}$, as in the Figure 6.

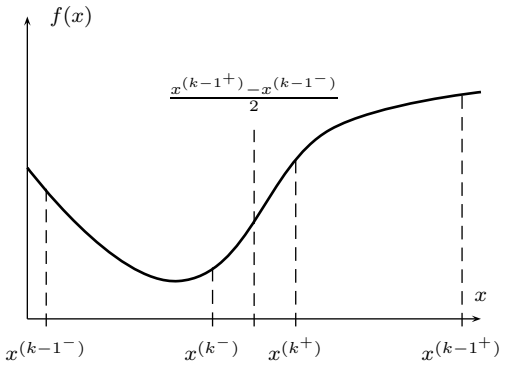

Fig. 6. Dichotomy method

6.6. FIBONACCI METHOD

In a golden-search method the reduction factor is constant, as ρ does not vary among all iterations. Let us assume that it can be changed with run of the algorithm so that at kth iteration the value ρ_k is used, and ρ_{k+1} in the next iteration. Similarly to the golden-search method, the factor $0 < \rho_k \leqslant \frac{1}{2}$ is chosen to compute the value of the objective function at a single point only at each iteration.

According to the Figure 7, the reduction factor ρ_k is sought such that

$$\rho_{k+1}(1 - \rho_k) = 1 - 2\rho_k\,,$$

with

$$\rho_{k+1} = 1 - \frac{\rho_k}{1-\rho_k}\,.$$

There are many choices of $\rho_1, \rho_2, \ldots$ that satisfy the above requirement and the inequality $0 < \rho_k \leqslant \frac{1}{2}$, e.g. $\rho_1 = \rho_2 = \ldots = \frac{3-\sqrt{5}}{2}$ (golden-search method).

We can assume that the rule of choosing the reduction factors exists, and after N iterations the initial range reduces by

$$(1-\rho_1)(1-\rho_2)\cdots(1-\rho_N).$$

The appropriate choice of reduction rates results from solving the nonlinear programming problem

$$\begin{aligned} \min_{\rho_1,\,\rho_2,\,\ldots,\,\rho_N} \quad & (1-\rho_1)(1-\rho_2)\cdots(1-\rho_N) \\ \text{s.t.} \quad & \rho_{k+1} = 1 - \frac{\rho_k}{1-\rho_k}\,, \quad k = 1,\,2,\,\ldots,\,N-1\,, \\ & 0 < \rho_k \leqslant \frac{1}{2}\,, \quad k = 1,\,2,\,\ldots,\,N-1\,, \end{aligned}$$

that is:

$$\begin{aligned} \rho_1 &= \frac{F_{N-1}}{F_{N+1}}\,, \\ \rho_2 &= \frac{F_{N-2}}{F_N}\,, \\ &\vdots \\ \rho_k &= \frac{F_{N-k}}{F_{N-k+2}}\,, \\ &\vdots \\ \rho_N &= \frac{F_0}{F_2} = \frac{1}{2}\,, \end{aligned}$$

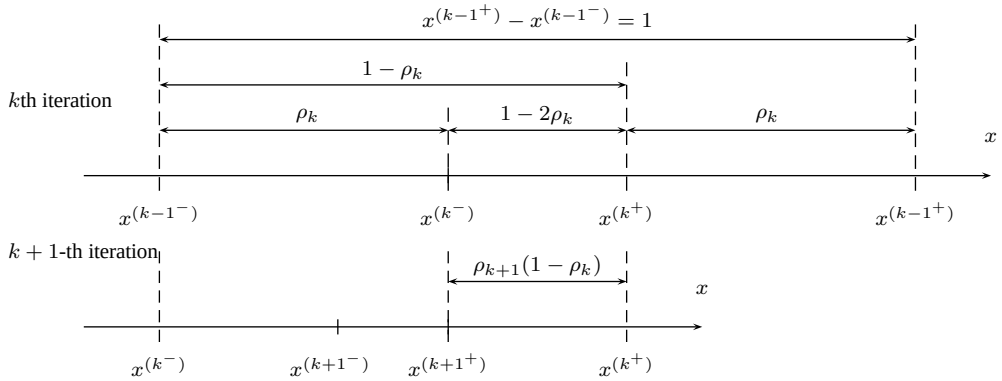

Fig. 7. The method of choosing intermediate points

where the numbers:

$$\begin{aligned} F_0 &= 1\,, \\ F_1 &= 1\,, \\ F_2 &= 2\,, \\ F_3 &= 3\,, \\ F_4 &= 5\,, \\ F_5 &= 8\,, \\ F_6 &= 13\,, \\ F_7 &= 21\,, \\ &\vdots \end{aligned}$$

form Fibonacci sequence, with $F_k = F_{k-1} + F_{k-2}$.

At the Nth iteration, it holds that $\rho_N = \frac{1}{2}$, thus the reduction factor is modified to be $\rho_N = \frac{F_0}{F_2} - \delta = \frac{1}{2} - \delta$, with $0 < \delta \ll 1$ to assure that the both proposed points are close to the middle to the last uncertainty interval.

Minimal number of iterations satisfies $F_{N+1} \geqslant \frac{1}{\epsilon}$, and intermediate points are chosen according to the formulas:

$$\begin{aligned} \hat{x}^{(k^-)} &= x^{(k-1^-)} + \rho_k(x^{(k-1^+)} - x^{(k-1^-)})\,, \\ \hat{x}^{(k^+)} &= x^{(k-1^-)} + (1-\rho_k)(x^{(k-1^+)} - x^{(k-1^-)})\,, \end{aligned}$$

with $\rho_k = \frac{F_{N-k}}{F_{N-k+2}}$.

7. SECOND-ORDER METHOD (NEWTON METHOD)

7.1. BASIC STEP

Let us go back to the assumption that the minimizer of f with respect to $x \in \mathscr{R}$ is sought. In addition let it be possible to compute $f(x^{(k)})$, $\dot{f}(x^{(k)})$ and $\ddot{f}(x^{(k)})$ at each point $x^{(k)}$.

One can fit a quadratic function $q(x)$ at each $x^{(k)}$, which first and second derivatives at $x^{(k)}$ are equal to $\dot{f}(x^{(k)})$ and $\ddot{f}(x^{(k)})$, respectively. Such a quadratic approximation has the form

$$q(x) = f(x^{(k)}) + \dot{f}(x^{(k)})(x - x^{(k)}) + \frac{1}{2}\,\ddot{f}(x^{(k)})(x - x^{(k)})^2\,,$$

where $q(x^{(k)}) = f(x^{(k)})$, $\dot{q}(x^{(k)}) = \dot{f}(x^{(k)})$, $\ddot{q}(x^{(k)}) = \ddot{f}(x^{(k)})$, according to the Figure 8.

Now, instead of minimizing f, one can find the minimizer of its approximation q, and on the basis of first-order necessary condition obtaining

$$\dot{q}(x^{(k)}) = \dot{f}(x^{(k)}) + \ddot{f}(x^{(k)})(x - x^{(k)}) = 0\,.$$

Having substituted $x = x^{(k+1)}$, we have

$$x^{(k+1)} = x^{(k)} - \frac{\dot{f}(x^{(k)})}{\ddot{f}(x^{(k)})}\,.$$

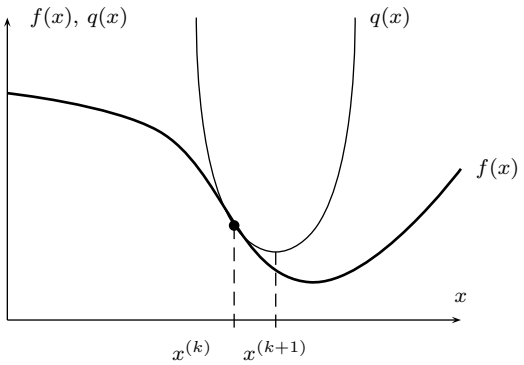

Fig. 8. Newton method for $\ddot{f}(x^{(k)}) > 0$

Since the approximation q is quadratic, the sufficient condition for the existence of its minimum is $\ddot{f}(x^{(k)}) > 0$.

At each iteration we have to assume that $\ddot{f}(x^{(k)}) > 0$. In the opposite case, i.e. $\ddot{f}(x^{(k)}) < 0$, the algorithm may stop avoiding the minimizer of f, as presented in Figure 9.

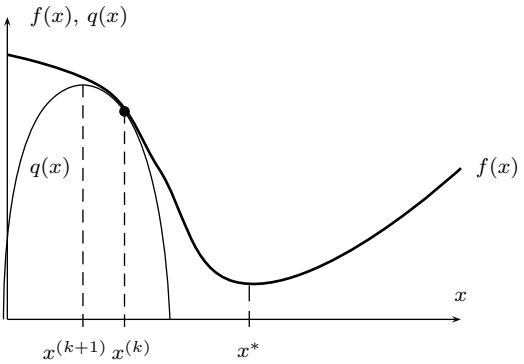

Fig. 9. Newton algorithm for $\ddot{f}(x^{(k)}) < 0$

7.2. EXAMPLE OF A NEWTON METHOD RUN

Find the minimizer of $f(x) = x^3 + 10x^2 + 20x + \mathrm{e}^{-x}$ with absolute tolerance $\epsilon = 0.01$ and $x^{(0)} = 1$.

Derivatives of $f(x)$ have the forms:

$$\begin{aligned} \dot{f}(x) &= 3x^2 + 20x + 20 - \mathrm{e}^{-x}\,, \\ \ddot{f}(x) &= 6x + 20 + \mathrm{e}^{-x}\,, \end{aligned}$$

and approximate solutions at consecutive iterations:

$$
\begin{aligned}
k &= 0 \\
\dot{f}(1) &= 42.6321\,, \\
\ddot{f}(1) &= 26.3679\,, \\
x^{(1)} &= 1 - \frac{42.6321}{26.3679} = -0.6168\,, \\
&\quad |-0.6168 - 1| = 1.6168 > \epsilon = 0.01\,; \\
k &= 1 \\
\dot{f}(-0.6168) &= 6.9523\,, \\
\ddot{f}(-0.6168) &= 18.1522\,, \\
x^{(2)} &= -0.6168 - \frac{6.9523}{18.1522} = -0.9998\,, \\
&\quad |-0.9998 + 0.6168| = 0.3830 > 0.01\,; \\
k &= 2 \\
\dot{f}(-0.9998) &= 0.2851\,, \\
\ddot{f}(-0.9998) &= 16.7189\,, \\
x^{(3)} &= -0.9998 - \frac{0.2851}{16.7189} = -1.0169\,, \\
&\quad |-1.0169 + 0.9998| = 0.0171 > 0.01\,; \\
k &= 3 \\
\dot{f}(-1.0169) &= -3.5434 \cdot 10^{-4}\,, \\
\ddot{f}(-1.0169) &= 16.6632\,, \\
x^{(4)} &= -1.0169 - \frac{-3.5434 \cdot 10^{-4}}{16.6632} = -1.0169\,, \\
&\quad |-1.0169 + 1.0169| \cong 0 < 0.01\,.
\end{aligned}
$$

Since the stopping criterion, i.e. $|x^{(4)} - x^{(3)}| \cong 0 \leqslant \epsilon = 0.01$, is satisfied thus $\hat{x}^* = -1.0169$ and $f(\hat{x}^*) = -8.2841$.

The Newton method can be alternatively shown as a mean to finding the point at which the first derivative of f vanishes. If we assume that $g(x) = \dot{f}(x)$, the Newton method transforms to the algorithm enabling one to solve equations $g(x) = 0$ in the form

$$x^{(k+1)} = x^{(k)} - \frac{g(x^{(k)})}{\dot{g}(x^{(k)})}\,,$$

namely to the method of tangents. The name can be understood when examining Figure 10.

If the tangent line to $g(x)$ at $x^{(k)}$ is drawn, it intersects the x axis at the point $x^{(k+1)}$ that should be closer to x^* than $x^{(k)}$. It also holds that

$$\dot{g}(x^{(k)}) = \frac{g(x^{(k)})}{x^{(k)} - x^{(k+1)}}\,,$$

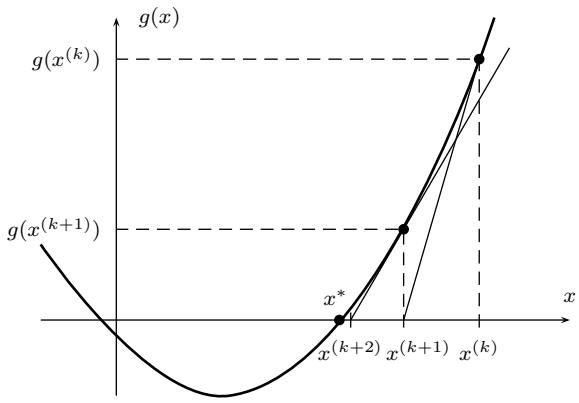

Fig. 10. Method of tangents

from which

$$x^{(k+1)} = x^{(k)} - \frac{g(x^{(k)})}{\dot{g}(x^{(k)})}\,.$$

The method of tangents may fail when $\frac{g(x^{(0)})}{\dot{g}(x^{(0)})}$ is not small enough, so any initial knowledge about the solution of $g(x) = 0$ is necessary.

A version of Newton method when derivative of $f(x)$ cannot be explicitly found uses the approximation:

$$\dot{f}(x^{(k)}) \cong \frac{f(x^{(k)} + \delta) - f(x^{(k)} - \delta)}{2\delta}\,,$$
$$\ddot{f}(x^{(k)}) \cong \frac{f(x^{(k)} + \delta) - 2f(x^{(k)}) + f(x^{(k)} - \delta)}{\delta^2}\,,$$

with a small number $\delta > 0$ (quasi-Newton method).

8. FIRST-ORDER METHOD (SECANT METHOD)

In the Newton method, the knowledge about derivative of f is used

$$x^{(k+1)} = x^{(k)} - \frac{\dot{f}(x^{(k)})}{\ddot{f}(x^{(k)})}\,.$$

When the second derivative of f is unavailable, it can be approximated as

$$\frac{\dot{f}(x^{(k)}) - \dot{f}(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}\,.$$

Having substituted this approximation to the general Newton method,

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{\dot{f}(x^{(k)}) - \dot{f}(x^{(k-1)})}\,\dot{f}(x^{(k)})\,,$$

one obtains secant method. It is necessary now to know $x^{(0)}$ and $x^{(-1)}$ to start the algorithm that can be summarized as:

$$x^{(k+1)} = \frac{\dot{f}(x^{(k)})x^{(k-1)} - \dot{f}(x^{(k-1)})x^{(k)}}{\dot{f}(x^{(k)}) - \dot{f}(x^{(k-1)})}.$$

Similarly to the Newton method, the value of f is not computed directly, but rather its derivative is driven to zero, and the method can be used to solve the equation $g(x) = 0$ with a basic step

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{g(x^{(k)}) - g(x^{(k-1)})}\, g(x^{(k)})$$

or

$$x^{(k+1)} = \frac{g(x^{(k)})x^{(k-1)} - g(x^{(k-1)})x^{(k)}}{g(x^{(k)}) - g(x^{(k-1)})}.$$

The idea of secant method is presented in Figure 11 – as it can be seen, the secant from $(k-1)$th iteration to kth iteration is drawn to compute $(k+1)$th iteration.

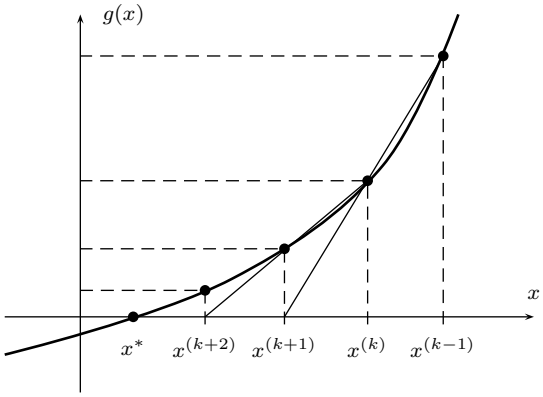

Fig. 11. Secant method

9. COMPARISON OF A SINGLE VARIABLE ALGORITHMS

A general comparison of the presented algorithms is difficult, and its effectiveness depends on:

- number of extremum points of f

  if the function is not unimodal, the computed local minimum may not in general be global thus multiple tests are needed to find the global minimizer changing the initial points; in the case of zero-order algorithms, the length and the location of the initial range must be varied; such a methodology should lead to obtaining the approximate solution for a function with multiple minima or a single extremum point, allowing one to choose the global minimum;

- first and second derivative norms;
- rounding errors.

In order to choose the appropriate algorithm for finding the minimizer we choose:

- Newton method, if explicit formulas to the function and its derivatives are available,
- Fibonacci method, if derivatives are unavailable, but the initial set is known|,
- quasi-Newton method, if derivatives, as well as initial set, are unavailable.

10. OPTIMAL CONTROL EXAMPLE

Find the controller that minimizes $J_t = (r_t - y_t)^2$ for the plant $y_t = 1.8y_{t-1} - 0.9y_{t-2} + u_t + 0.5u_{t-1}$ using quasi-Newton method in the unconstrained case, soft and hard constraints imposed on the control signal with zero initial conditions.

The performance index J_t is nonnegative, thus its minimal value, i.e.
zero, refers to the perfect tracking case. Using the formula for the tracking error

$$\begin{aligned} e_t &= r_t - 1.8y_{t-1} + 0.9y_{t-2} - u_t - 0.5u_{t-1} = \\ &= (r_t - 1.8y_{t-1} + 0.9y_{t-2} - 0.5u_{t-1}) - u_t = \varphi_t - u_t \end{aligned}$$

a one-step performance index takes the form

$$J_t = e_t^2 = (\varphi_t - u_t)^2 = u_t^2 - 2\varphi_t u_t + \varphi_t^2 \,,$$

i.e. the form of a quadratic function.

Introduction of soft constraints can be performed by assuming the performance index takes the form

$$J_t = e_t^2 + q_u u_t^2 = (1 + q_u)u_t^2 - 2\varphi_t u_t + \varphi_t^2 \,,$$

where the control cost coefficient $q_u \geqslant 0$ refers to the impact of u_t on the increase of the performance index at each step. It is to be stressed that the second term of J_t is also nonnegative, thus the control action is penalized for each $u_t \neq 0$.

Hard constraints can be implemented by transformation method. Let the constraint $-a \leqslant u_t \leqslant a$ ($a > 0$) be given. It holds that for $u_t = a\sin(x_t)$ the performance index can be transformed to (a function of x_t)

$$J_t = e_t^2 = a^2 \sin^2(x_t) - 2a\varphi_t \sin(x_t) + \varphi_t^2 = f(x_t) \,.$$

Having assumed that at each time instant $u_t^* = \arg\min J_t$ results from the optimization procedure with $\epsilon = 10^{-12}$, $\delta = 10^{-3}$, the tracking properties are presented in the Figure 12.

11. EFFICIENCY OF ITERATIVE MINIMIZATION METHODS OF FUNCTIONS OF SINGLE VARIABLE – A COMPARISON

11.1. ZERO-ORDER METHODS

The criterion related to the execution speed of the algorithm is the number N if iterations that it takes to find the approximate minimizer with the given tolerance ϵ. In the case of

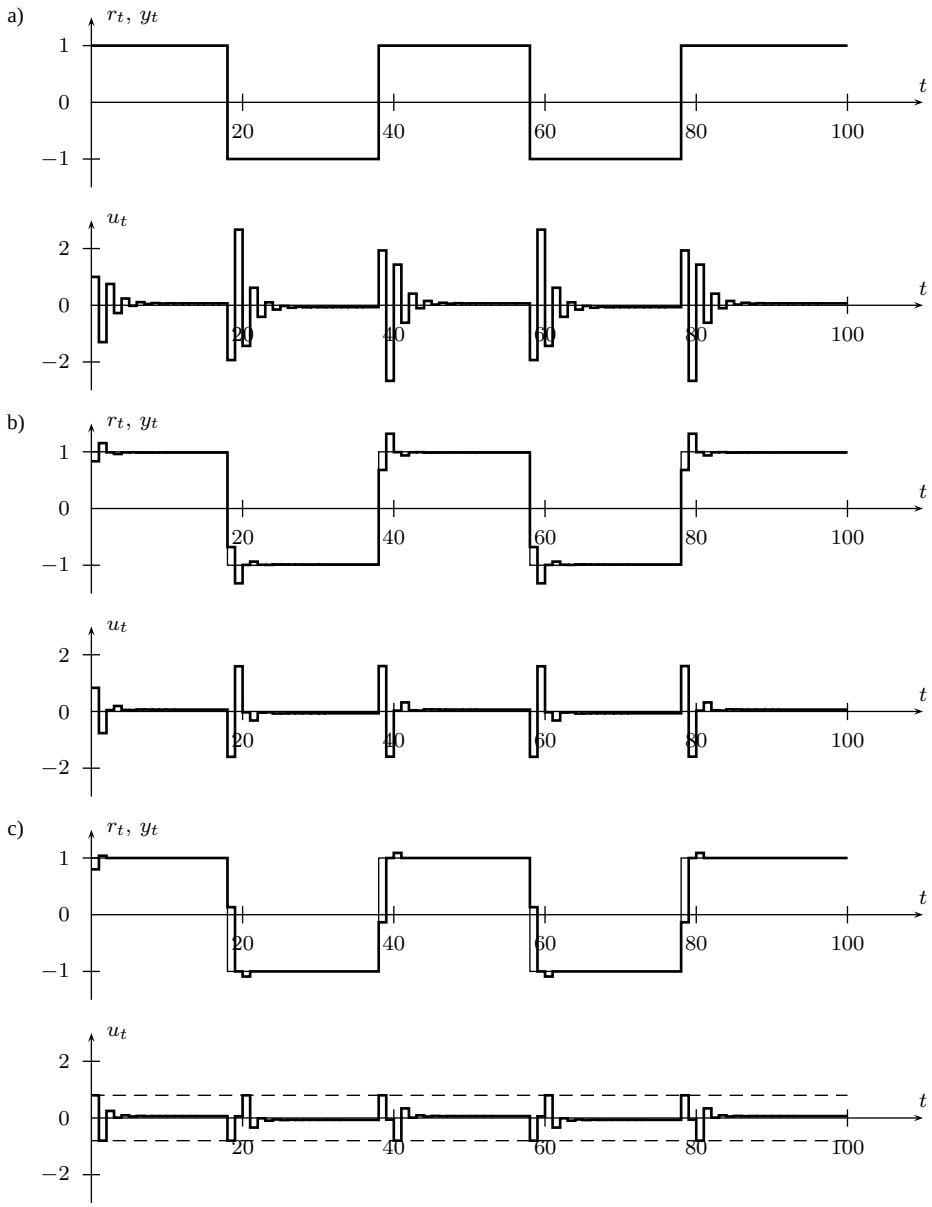

Fig. 12. a) unconstrained case, b) soft constraints ($q_u = 0.2$), c) hard constraints with the cut-off at level ± 0.8

zero-order methods, the formula $N(\epsilon)$ is a direct result of inference concerning the reduction factor at each iteration.

For the four previously considered methods, presented in Section 6, the comparison results are presented in the Figure 13 where the given tolerance is presented on the x axis in

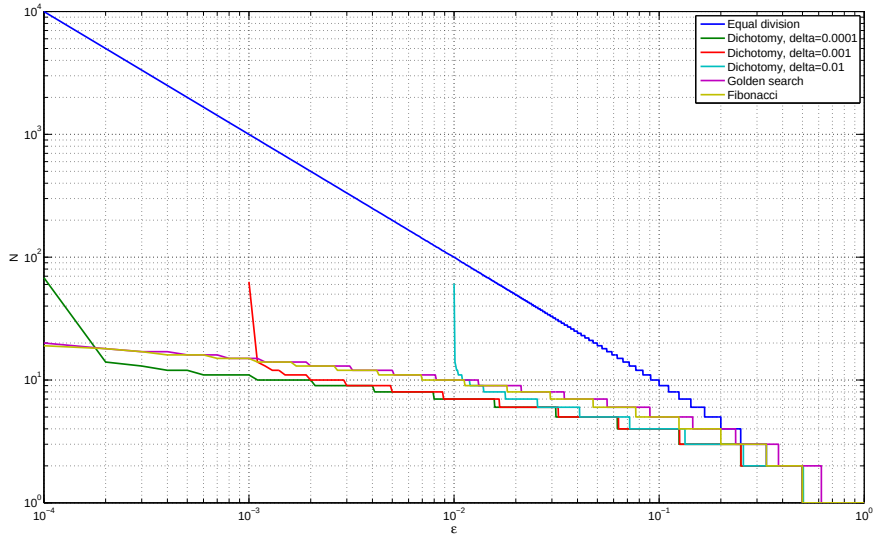

Fig. 13. Iteration number vs. tolerance or zero-order methods

range from 10^{-4} to 10^{0} in log scale. The y axis represents the number N of iterations, also in log scale. There are three different plots for $\delta = 10^{-4}$, $\delta = 10^{-3}$, and $\delta = 10^{-2}$ respectively and dichotomy method. One can verify from this Figure which method for the given tolerance ϵ requires the fewest number of iterations to obtain the approximate solution. The most effective method is the dichotomy method, the second one is the Fibonacci method, later the golden-search method, and finally, the equal division method.

11.2. FIRST-ORDER METHODS

11.2.1. PRELIMINARIES

In gradient methods, the iterative process stops when a stopping criterion is satisfied. On the contrary to zero- order methods, the number of iterations required to solve the problem does not depend on ϵ only, but also on the shape of the minimized function $f(x)$. In order to check the efficiency of first-order methods, the impact of ϵ on the iteration number taking aggregated stoping criterion into account, namely $N(\epsilon_1, \epsilon_2)$ for each of the four methods from the Section 8 and

$$|x^{(k+1)} - x^{(k)}| \leqslant \varepsilon_1 \,.$$
$$\|\nabla f(x^{(k)})\| \leqslant \varepsilon_2 \,.$$

The minimized function is a polynomial, i.e. a continuous function. The impact of steps in the plot of the minimized function (discontinuity points) has also been verified with recording the number of iterations. For this purpose, it has been checked what is the change in

$N(\epsilon_1,\, \epsilon_2)$ when the minimized function is continuous, has discontinuity points and includes step changes. It has also been verified what is the impact of the coefficient of the greatest power of the polynomial on the iteration number.

11.2.2. NUMBER OF ITERATIONS FOR CONTINUOUS FUNCTION

The function $N(\epsilon_1,\, \epsilon_2)$ has been computed for four different gradient methods, with the objective function

$$f(x) = \frac{1}{6}\, x^3 + 6x^2 + 3x - 1\,.$$

The plot of the function has been presented in Figure 14a. Its minimizers are $x^* = -0.2527$ and $x^* = -1.3776$. Two row vectors of 100 linearly spaced points between 10^{-4} and 10^0 have been generated, i.e. ϵ_1 and ϵ_2, and for each of the pairs $(\epsilon_1,\, \epsilon_2)$ each method has been run for a hundred initial points linearly spaced in the range $[-10,\, 10]$. The value of $N(\epsilon_1,\, \epsilon_2)$ is a mean value value computed from one hundred runs of the algorithm.

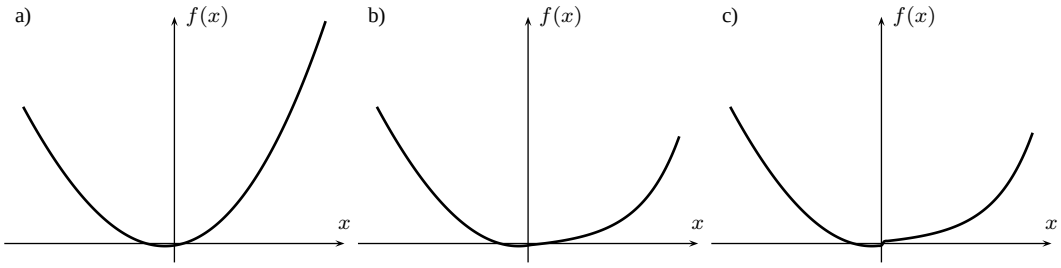

Fig. 14. Objective function: a) continuous, b) with a discontinuity point, c) with a step change

In the case of secant method, two initial points are required, thus for this case the point $x^{(-1)}$ is chosen as above, and for each choice of $x^{(-1)}$, the point $x^{(0)}$ lies in the middle of the range between $x^{(-1)}$ and x^*. In the case of quasi-Newton method, the values of $N(\epsilon_1,\, \epsilon_2)$ have been computed for $\delta = 10^{-4}$ and $\delta = 10^{-1}$ with results presented as mean numbers.

The results of the performed tests are presented in Figures 15–18. In the case of quasi-Newton method, the results for both values of δ are almost equal, what is why one can draw the conclusion, that this parameter has a minor impact on the number of iterations reached. In Table 1 a mean number of iterations has been presented for four pairs of $(\epsilon_1,\, \epsilon_2)$.

Tab. 1. Mean number of iterations for different pairs o $(\varepsilon_1,\, \varepsilon_1)$

ε_1	ε_2	secant	Newton	q.-Newton	tangents
10^{-4}	10^{-4}	3.83	4.52	3.52	3.77
10^{-4}	10^{-1}	3.14	3.95	2.92	3.77
10^{-1}	10^{-4}	3.83	4.52	3.52	3.52
10^{-1}	10^{-1}	1.59	2.88	1.88	2.35

Based on the performed tests, it can be seen that for not strict stopping criteria a secant method is mostly effective, and the least effective the Newton method is. From the Table 1 it

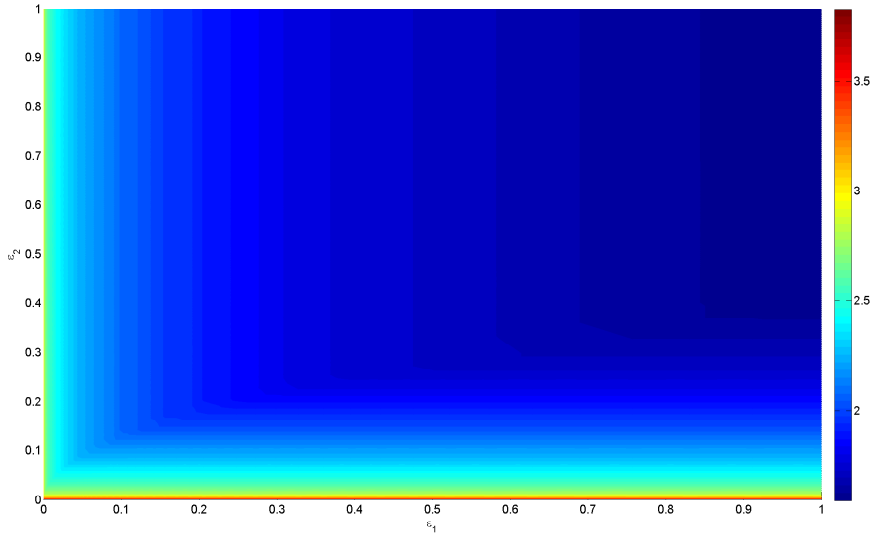

Fig. 15. Mean number of iterations N for secant method

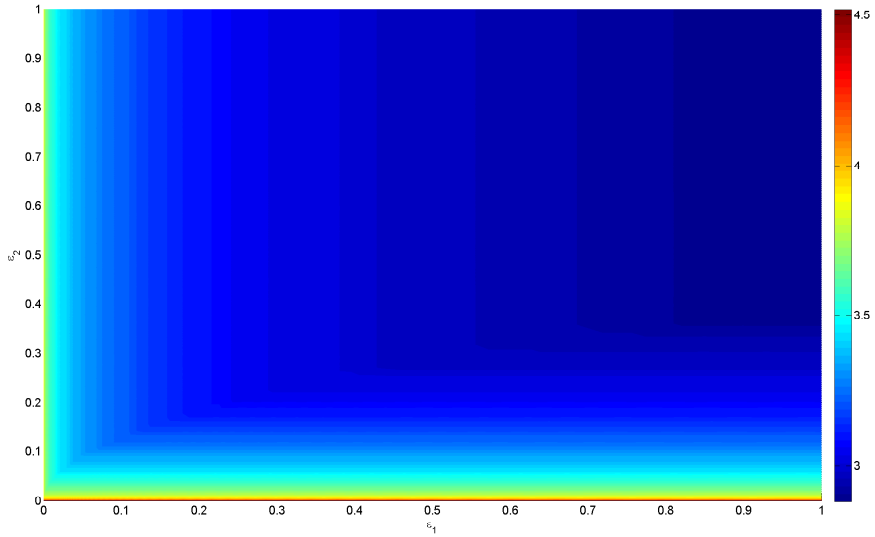

Fig. 16. Mean number of iterations N for Newton method

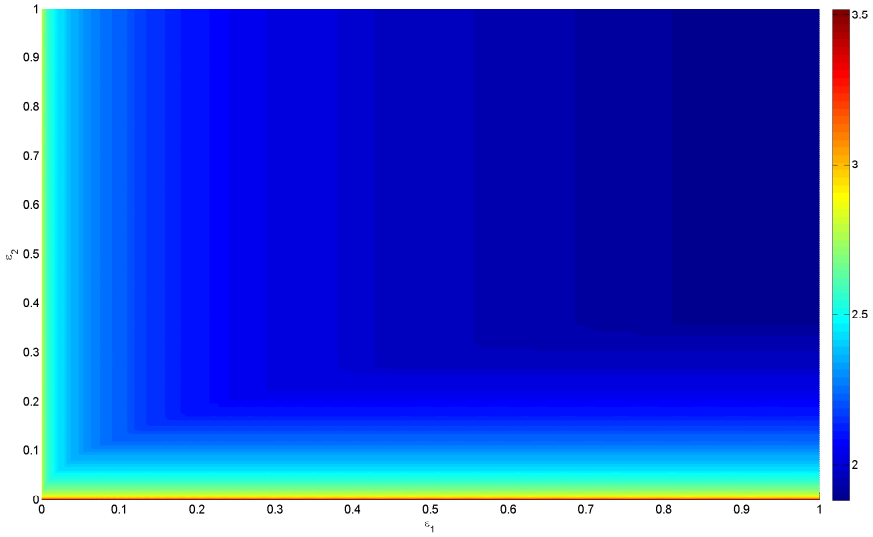

Fig. 17. Mean number of iterations N for quasi-Newton method

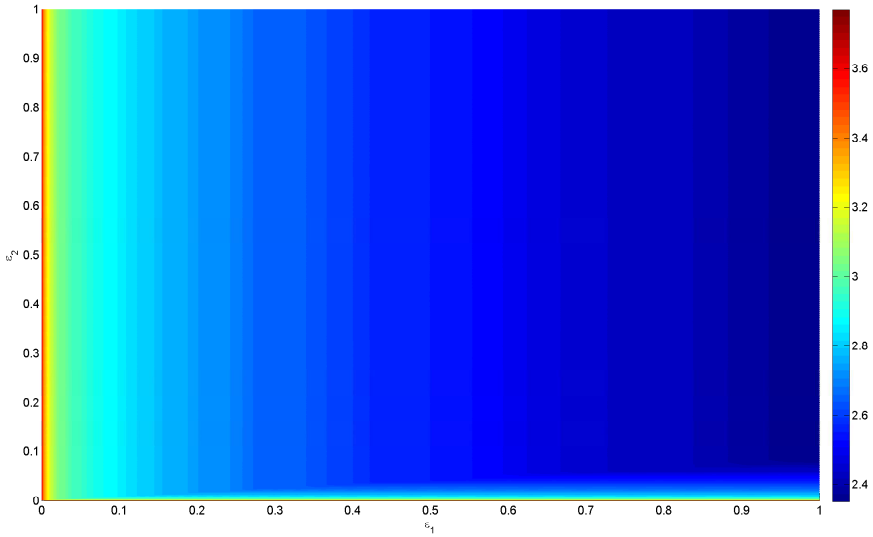

Fig. 18. Mean number of iterations N for method of tangents

can also be seen in the case of Newton method, quasi-Newton method and and secant method, there is smaller sensitivity of N to ε_1, than to ε_2. For the method of tangents, number N is more sensitive to ε_2, than to ε_1.

11.3. CONTINUITY IMPACT ON THE NUMBER OF ITERATIONS

Similar test, respecting the impact of continuity, points of discontinuity and steps in objective function have been carried out as in the previous Section. A function with discontinuity has a form

$$f(x) = \begin{cases} \frac{1}{6}x^3 + 6x^2 + 3x - 1 & \text{for } x < 0 \\ \mathrm{e}^x + x - 1.9016 & \text{for } x \geqslant 0 \end{cases},$$

with the plot presented in Figure 14b, and for function with step

$$f(x) = \begin{cases} \frac{1}{6}x^3 + 6x^2 + 3x - 1 & \text{for } x < 0 \\ \mathrm{e}^x + x & \text{for } x \geqslant 0 \end{cases}$$

the plot has been presented in Figure 14c.

Below, the plots of $N(\epsilon_1, \epsilon_2)$ have been presented in the Figures 19–24. In Table 2, number of iterations for selected points are presented for discontinuous function. Table 3 presents the function N for $\epsilon_1 = 0.0001$, $\epsilon_2 = 0.0001$ and continuous and discontinuous function. From Table 3 it results that secant method is mostly efficient for discontinuous function, enabling one to get the solution with the least number of iterations. From Table 3 it can be seen that the proposed discontinuity point causes severe increase in number of iterations. This is, however, with no major change in the shape of the surface describing $N(\varepsilon_1, \varepsilon_2)$ function.

Tab. 2. Mean number of iteration for different stopping criteria and discontinuous function

ε_1	ε_2	secant	q.-Newton	tangents
10^{-4}	10^{-4}	6.60	7.41	7.68
10^{-4}	10^{-1}	5.97	6.84	7.68
10^{-1}	10^{-4}	6.60	7.41	7.41
10^{-1}	10^{-1}	3.35	5.83	6.30

Tab. 3. Continuity/discontinuity impact on the mean number of iterations

function/method	secant	q.-Newton	tangents
continuous	3.83	3.52	3.77
discontinuous	6.60	7.41	7.68

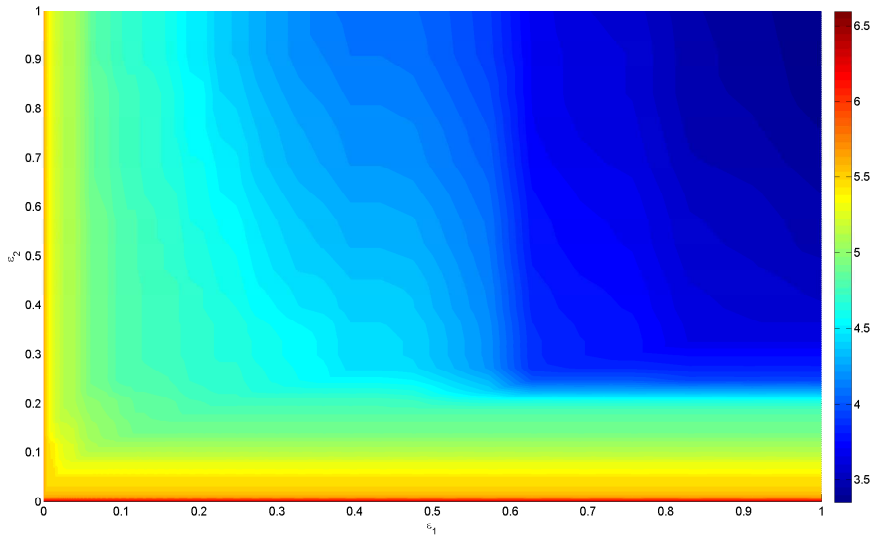

Fig. 19. Mean number of iterations N for secant method and discontinuous function

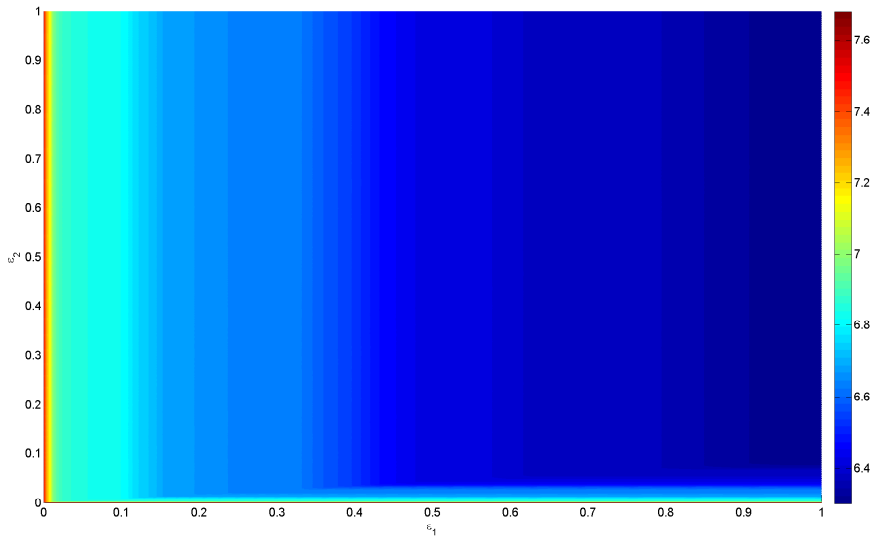

Fig. 20. Mean number of iterations N for method of tangents and discontinuous function

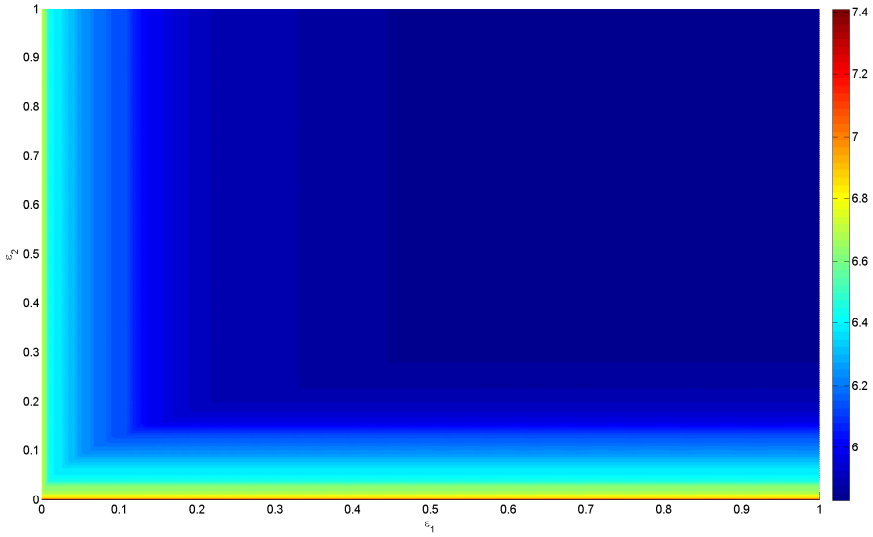

Fig. 21. Mean number of iterations N for quasi-Newton method and discontinuous function

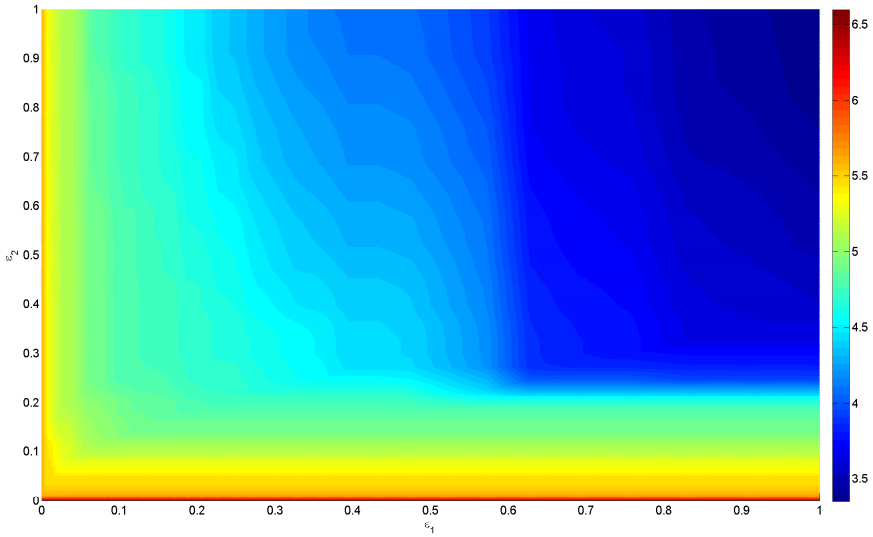

Fig. 22. Mean number of iterations N for secant method and function with a step change

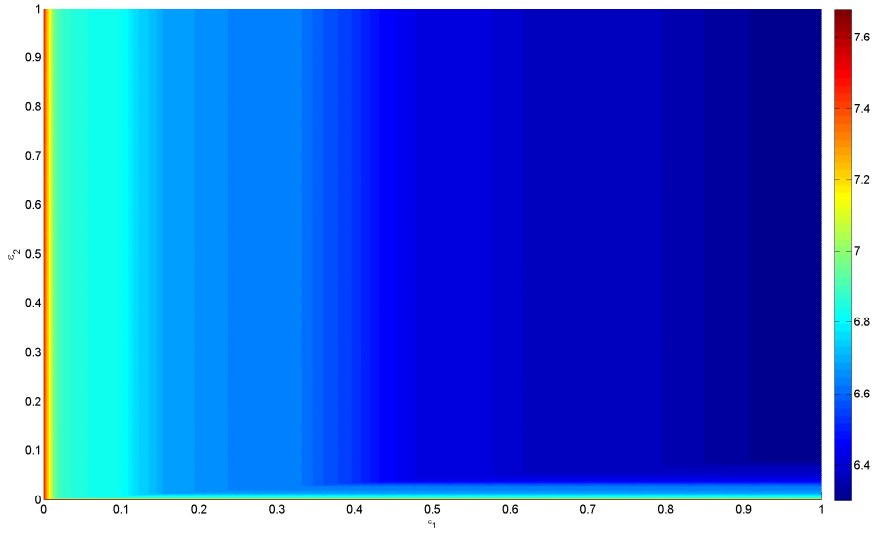

Fig. 23. Mean number of iterations N for method of tangents and function with a step change

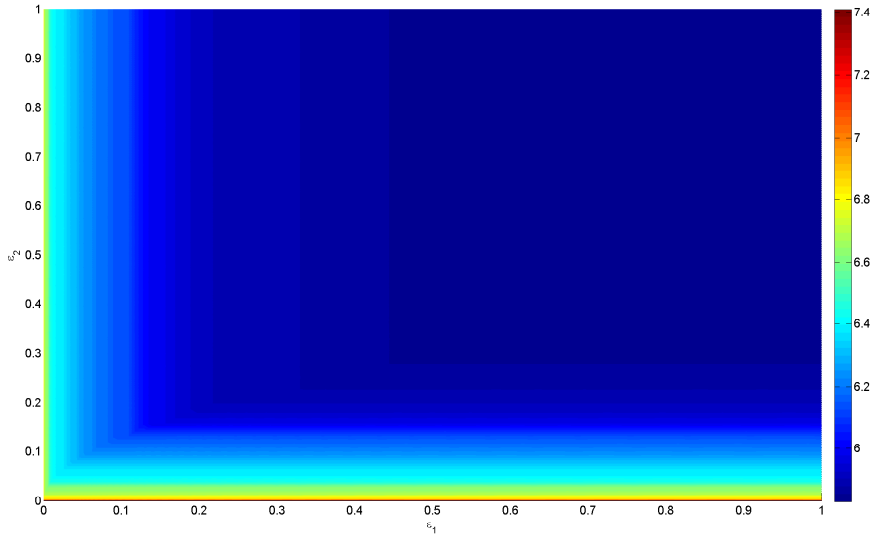

Fig. 24. Mean number of iterations N for quasi-Newton method and function with a step change

11.4. IMPACT OF STEEPNESS OF FUNCTION ON NUMBER OF ITERATIONS

In the current test, the impact of coefficient at the largest power of the decision variable in the polynomial defining objective function

$$f(x) = ax^3 + 6x^2 + 3x - 1$$

has been changed. For 20 linearly spaced values of a from the range $[0.01,\ 0.15]$ and constant stopping criteria, namely $\varepsilon_1 = 0.0001$, $\varepsilon_2 = 0.0001$ a number of iterations has been given allowing to obtain approximate solution to the minimization problem with the selected accuracy for each of the four methods.

This function has been presented in Figure 14c, numerical results have been presented in Table 4, and in the graphic form in Figure 25. As it can be seen, the function $N(a)$ is approximately linear, the linear regression coefficients $an + b$ computed and presented in the Table 5. The results indicate that the greater the coefficient a is, the greater the iteration number becomes. Since the directional coefficient for the presented methods are similar, one can say that for each of them, the impact of a is similar.

Tab. 4. Mean number of iterations vs. a

a/method	secant	Newton	q.-Newton	tangents
0.0100	2.57	3.09	2.09	2.51
0.0174	2.70	3.39	2.39	2.68
0.0248	2.76	3.54	2.54	2.76
0.0321	2.80	3.62	2.62	2.80
0.0395	2.87	3.67	2.67	2.83
0.0468	3.02	3.71	2.71	2.85
0.0542	3.11	3.74	2.74	2.91
0.0616	3.20	3.77	2.77	2.98
0.0689	3.26	3.82	2.82	3.08
0.0763	3.33	3.87	2.87	3.16
0.0837	3.37	3.92	2.92	3.21
0.0911	3.41	3.96	2.96	3.28
0.0984	3.45	4.01	3.01	3.32
0.1058	3.50	4.07	3.07	3.36
0.1132	3.54	4.12	3.12	3.40
0.1206	3.59	4.16	3.16	3.48
0.1279	3.64	4.22	3.22	3.52
0.1353	3.68	4.29	3.29	3.56
0.1426	3.71	4.33	3.33	3.62
0.1500	3.76	4.38	3.38	3.65

12. SUMMARY

Based on the performed tests it can be seen that for the approximate solutions obtained with low accuracy (tolerances $\geqslant 0.5$) and combined stopping criteria, and in the case of

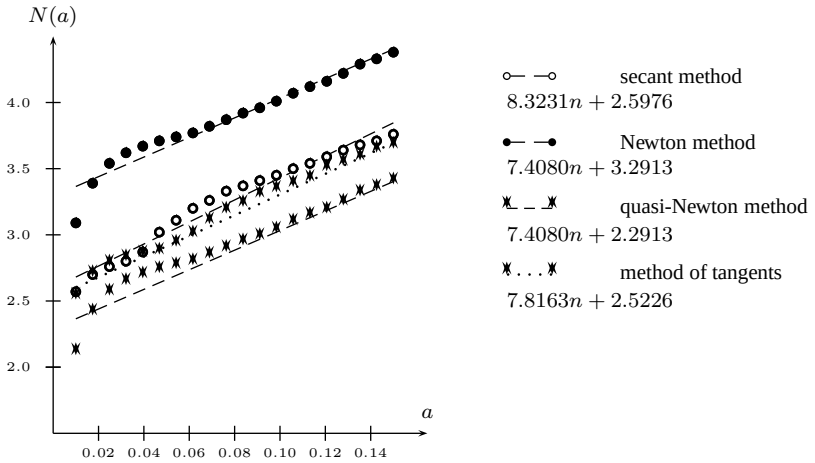

Fig. 25. A comparison of effectiveness of selected functions

Tab. 5. Linear regression coefficients

coeff.	secant	q.-Newton	Newton	tangents
a	8.3231	7.4080	7.4080	7.8163
b	2.5976	3.2913	2.2913	2.5226

continuous function, the performance of approx. 1.5 iterations per solution is achieved with secant method, 2.3 for method of tangents and, similarly, 5.8 or Newton methods. In the case of discontinuous function, as well as for the function with step change, 3.3 iterations per solution for secant method, 5.8 for quasi-Newton method and 6.3 for method of tangents.

When high accuracy is taken into consideration, as in Table 5 and Figure 25, the worst is, surprisingly, Newton method (with improving performance for $a \rightarrow 0$ – quadratic convergence and one-iteration solution for quadratic f), secondly, secant method, furthermore, method o tangents, and, finally, quasi-Newton method.

As far as Newton method is concerned, the low performance may be due to increase of the coefficient at the greatest power of x, as ax^3 increases for large values of initial points more rapidly than $6x^2$, and the minimizers in the sequence of quadratic approximations of this function require more steps to converge to the true minimum.

The results of this paper should aid the researchers in choosing the right method of optimization based not only on the information about the shape of objective function, but also on the graphical presentation of the interplay between stopping criteria.

REFERENCES

[1] Bazaraa M.S., Sherali H.D., Shetty C.M., *Nonlinear Programming: Theory and Algorithms*, 3rd ed., New Jersey, Wiley 2006.

[2] Bhatti M.A., *Practical Optimization Methods with Mathematica Applications*, New York, Springer Verlag 2000.

[3] Chong E.K.P., Żak S.H., *An Introduction to Optimization*, 2nd ed., John Wiley & Sons 2001.

[4] Horla D., *Metody obliczeniowe optymalizacji w zadaniach*, Poznań, Wydawnictwo Politechniki Poznańskiej, 2008 (in Polish).

[5] Stadnicki J., *Teoria i praktyka rozwiązywania zadań optymalizacji z przykładami zastosowań technicznych*, Warszawa, Wydawnictwa Naukowo-Techniczne 2006 (in Polish).

ABSTRACT

The paper considers performance issues of a class of iterative minimization methods of unconstrained single variable problems. Problem structures that assure superior performance of a specific method have been stipulated with appropriate conclusions drawn.

OCENA SZYBKOŚCI DZIAŁANIA ITERACYJNYCH METOD MINIMALIZACJI FUNKCJI JEDNEJ ZMIENNEJ BEZ OGRANICZEŃ

Dariusz Horla

W artykule poruszono zagadnienie szybkości działania metod minimalizacji funkcji jednej zmiennej dla zadań programowania nieliniowego bez ograniczeń. Wskazano przypadki, dla których konkretna metoda działa szybciej niż pozostałe oraz wyciągnięto wnioski odnośnie takiego stanu rzeczy.

ACKNOWLEDGEMENTS

The author wishes to thank to Mr Marcin Wawrzonkowski for valuable help in conducting tests presented in the paper.