DOI: 10.3842/umzh.v76i5.7395

UDC 519.853.6 : 519.613.2

Viktor Zadachyn[1] (Simon Kuznets Kharkiv National University of Economics, Kharkiv),
Maxim Bebiya (V. N. Karazin Kharkiv National University)

COMBINED METHODS FOR SOLVING DEGENERATE UNCONSTRAINED OPTIMIZATION PROBLEMS

КОМБІНОВАНІ МЕТОДИ РОЗВ'ЯЗУВАННЯ ВИРОДЖЕНИХ ЗАДАЧ БЕЗУМОВНОЇ ОПТИМІЗАЦІЇ

We present constructive second- and fourth-order methods for solving degenerate unconstrained optimization problems. The fourth-order method applied in the present work is a combination of the Newton method and the method that uses fourth-order derivatives. Our approach is based on the decomposition of $\mathbb{R}^n$ into the direct sum of the kernel of a Hessian matrix and its orthogonal complement. The fourth-order method is applied to the kernel of the Hessian matrix, whereas the Newton method is applied to its orthogonal complement. This method proves to be efficient in the case of a one-dimensional kernel of the Hessian matrix. In order to get the second-order method, Newton's method is combined with the steepest-descent method. We study the efficiency of these methods and analyze their convergence rates. We also propose a new adaptive combined quasi-Newton-type method (ACQNM) based on the use of the second- and fourth-order methods in the degenerate case. The efficiency of ACQNM is demonstrated by analyzing an example of some most common test functions.

Представлено конструктивні методи другого та четвертого порядку для розв'язування вироджених безумовних задач оптимізації. Метод четвертого порядку, який ми використовуємо, є комбінацією методу Ньютона та методу, що використовує похідні четвертого порядку. Наш підхід базується на предстасвленні $\mathbb{R}^n$ як прямої суми ядра матриці Гесса та її ортогонального доповнення. До ядра матриці Гесса застосовано метод четвертого порядку, а до ортогонального доповнення — метод Ньютона. Цей метод виявляється ефективним у випадку одновимірного ядра матриці Гесса. Для отримання методу другого порядку, метод Ньютона комбінується з методом найкрутішого спуску. Досліджено продуктивність цих методів та проаналізовано швидкість їх збіжності. Ми також пропонуємо новий адаптивний комбінований квазіньютонівський метод (ACQNM), що використовує методи другого та четвертого порядку для виродженого випадку. Ефективність ACQNM показано на прикладі деяких найбільш поширених тестових функцій.

1. Introduction. Unconstrained optimization has been extensively studied due to its particular importance. Being applied in many nonlinear, constrained, multivariable optimization problems and software design, unconstrained optimization is important from both theoretical and practical points of view [6, 14, 15, 18, 19].

Practical problems often deal with the optimality criterion in the case of a degenerate extremum point, which significantly complicates finding the solution. Most known numerical methods of unconstrained optimization, even when using the second-order derivatives, have slow convergence in the case of degenerate problems [2]. In order to improve the performance of a numerical method, it is often helpful to use the higher-order derivatives [2, 22]. However, using the third- and fourth-order derivatives makes the numerical method rather complicated.

Fairly effective Newton's and quasi-Newton methods of unconstrained optimization were described in [3] and still draw significant attention [4, 7, 8, 11 – 13, 16, 17, 20, 23, 27 – 30]. The significant attention given to Newton's and quasi-Newton methods stems from their efficiency, global convergence properties, and their applicability to a wide range of optimization problems. Quasi-Newton methods are used to overcome computational challenges associated with calculating the exact Hessian matrix.

[1] Corresponding author, e-mail: zadachinvm@gmail.com.

For solving degenerate unconstrained optimization problems there was often used an approach based on regularization of numerical methods [11, 20, 27 – 30].

The main goal of this paper is to present constructive second- and fourth-order numerical methods for solving degenerate unconstrained optimization problems. Our approach is based on the decomposition of the entire space as the direct sum of the kernel and the column space of the Hessian matrix, the idea of which was given in [25]. In contrast to [25], a numerically stable spectral decomposition of the Hessian matrix is used instead of the Cholesky factorization [26]. This is crucial, especially in the neighborhood of a degenerate minimum point, as even small computational errors can significantly disrupt the iterative process. The approach of decomposing the entire space into a sum of orthogonal subspaces is also interesting because it allows for the use of different methods on different subspaces. A Newton-like method is applied on one subspace, while another method is used on the other subspace. To analyze the convergence rate of these methods, the generalized necessary and sufficient conditions for a minimum of a degenerate unconstrained optimization problem proposed in [24] are used. These conditions are slightly different from the higher-order optimality conditions given in [9, 10], which makes them more convenient for analyzing the convergence rate of the optimization methods in the case of a degenerate minimum point.

This paper is organized as follows. In Section 2, the idea of a family of combined methods for solving degenerate unconstrained optimization problems, which is based on the spectral decomposition of the Hessian matrix, is described. Then, as an implementation of this idea, a second-order combined method is presented in Subsection 2.1. A fourth-order method variant is presented in Subsection 2.2. Subsection 2.3 proposes an algorithm for finding two step sizes for these methods. Subsection 2.4 analyzes the convergence rate of the presented methods in the neighborhood of the degenerate minimum point under sufficient fourth-order minimum point conditions from [24]. In Section 3, the possibility of using quasi-Newton's variants of the previously described theoretical methods is considered. In Section 4, a practical adaptive algorithm is proposed, which combines the well-known BFGS method with the methods described above. Such an algorithm allows adaptation to changes in the behavior of the objective function during the process of finding minimum. Section 5 presents the results of testing the proposed adaptive algorithm.

2. Combined methods. Consider the degenerate problem of unconstrained optimization

$$\min f(x), \quad x \in \mathbb{R}^n, \tag{1}$$

where $f(x)$ is a p, $p \geq 4$, times differentiable function.

Let $x^* \in \mathbb{R}^n$ be a local minimum point of the function $f(x)$. Assume that the Hessian matrix $f^{(2)}(x^*)$ is degenerate, but it does not vanish identically.

The following notation is used [24]: $R_1 = \operatorname{Ker}(f^{(2)}(x^*)) = \{x \in \mathbb{R}^n \mid f^{(2)}(x^*)x = 0\}$ is the kernel of the Hessian matrix $f^{(2)}(x^*)$; R_2 is an orthogonal complement of the subspace R_1 (that is, $\mathbb{R}^n = R_1 \oplus R_2$); P is an orthogonal projector onto the subspace R_1; $P^{\perp}$ is an orthogonal projector onto the subspace R_2; $f^{(l)}(x^*)$ is the lth derivative of $f(x)$ at the point x^*; $f^{(l)}(x^*)[u^i, v^{l-i}]$ is a multilinear form of l arguments $u, v \in \mathbb{R}^n$ (the superscripts i and $(l-i)$ indicate the multiplicity of the corresponding argument). Note that the value of a symmetric multilinear form is invariant under various permutations of the arguments.

In addition, let $\mathbb{R}^{(n)^{p/2}}$ denote the $(n \times n \times \ldots \times n)$-dimensional space of $\left(\frac{p}{2}\right)$-dimensional arrays. Therefore, $f^{(\frac{p}{2}+1)}(x^*)$ can be considered as a linear mapping from $\mathbb{R}^{(n)^{p/2}}$ to $\mathbb{R}^n$. Moreover, $\left(f^{(\frac{p}{2}+1)}(x^*)\right)^T$ is a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^{(n)^{p/2}}$. Note that $f^{(\frac{p}{2}+1)}(x^*)$ and $\left(f^{(\frac{p}{2}+1)}(x^*)\right)^T$ are conjugate. In addition, $f^{(p)}(x^*)$ can be looked upon as a linear mapping from $\mathbb{R}^{(n)^{p/2}}$ to $\mathbb{R}^{(n)^{p/2}}$, that is, the value of the multilinear form $f^{(p)}(x^*)[u^p] = U^T f^{(p)}(x^*)U$, where $U \in \mathbb{R}^{(n)^{p/2}}$, is a $\left(\frac{p}{2}\right)$-dimensional matrix with elements $U_{i,j,\ldots,k} = u_i u_j \ldots u_k$.

Now we develop some second and higher-order combined methods for solving the degenerate problem (1). Consider an infinite sequence of iterates $\{x^{(k)}\}$ defined as follows:

$$x^{(k+1)} = x^{(k)} + \alpha_{k1} u_1^{(k)} + \alpha_{k2} u_2^{(k)}, \quad k = 0, 1, 2, \ldots, \tag{2}$$

where $x^{(0)}$ is an initial approximation of the minimum point, $u_1^{(k)}$, $u_2^{(k)}$ are orthogonal vectors, and $\alpha_{k1} > 0$ and $\alpha_{k2} > 0$ are step sizes along the directions $u_1^{(k)}$, $u_2^{(k)}$. At each step the method involves taking $u_2^{(k)} \in \operatorname{Ker}(f^{(2)}(x^{(k)}))$ and the vector $u_1^{(k)}$ that belongs to the orthogonal complement of $\operatorname{Ker}(f^{(2)}(x^{(k)}))$.

To correctly determine the rank of the matrix $H_k = f^{(2)}(x^{(k)})$ and find the orthogonal projectors onto the corresponding subspaces, the following approach is considered.

Since the matrix H_k is symmetric, according to the spectral decomposition [21], it can be factorized as

$$H_k = Q_k \Lambda_k Q_k^T, \tag{3}$$

where Q_k is an orthogonal matrix, $\Lambda_k = \operatorname{diag}(\lambda_i^{(k)})$ is a diagonal matrix whose diagonal elements $\lambda_i^{(k)}$, $i = 1, \ldots, n$, are the eigenvalues of the matrix H_k, sorted in descending order by absolute value.

The eigenvalues of H_k can be rearranged to present Λ_k in the form

$$\Lambda_k = \begin{bmatrix} \Lambda_{k1} & 0 \\ 0 & \Lambda_{k2} \end{bmatrix}, \tag{4}$$

where $\Lambda_{k1} = \operatorname{diag}(\lambda_i^{(k)})$, $|\lambda_i^{(k)}| > \varepsilon$, $i = 1, \ldots, r_k$, $r_k \le n$, $\varepsilon > 0$ is a regularization parameter of the numerical method [3], $\Lambda_{k2} = \operatorname{diag}(\lambda_i^{(k)})$, $|\lambda_i^{(k)}| \le \varepsilon$, $i = (r_k + 1), \ldots, n$. Then the matrix Q_k is also partitioned into two blocks $Q_k = \begin{bmatrix} Q_{k1} & Q_{k2} \end{bmatrix}$, where Q_{k1} is an $(n \times r_k)$-matrix, Q_{k2} is an $((n - r_k) \times n)$-matrix.

By using (3) and (4), we obtain the following:

$$H_k = \begin{bmatrix} Q_{k1} & Q_{k2} \end{bmatrix} \begin{bmatrix} \Lambda_{k1} & 0 \\ 0 & \Lambda_{k2} \end{bmatrix} \begin{bmatrix} Q_{k1}^T \\ Q_{k2}^T \end{bmatrix} = Q_{k1} \Lambda_{k1} Q_{k1}^T + Q_{k2} \Lambda_{k2} Q_{k2}^T = H_{k\varepsilon} + E_{k\varepsilon},$$

where $H_{k\varepsilon} = Q_{k1} \Lambda_{k1} Q_{k1}^T, E_{k\varepsilon} = Q_{k2} \Lambda_{k2} Q_{k2}^T = H_k - H_{k\varepsilon}$.

Thus, we have constructed the orthogonal projectors $P_k = I - Q_{k1} Q_{k1}^T$ and $P_k^{\perp} = I - P_k = Q_{k1} Q_{k1}^T$ (where I is an identity matrix) onto the subspace $\operatorname{Ker}(H_{k\epsilon}) = \{x \in \mathbb{R}^n \mid H_{k\epsilon} x = 0\}$ and the orthogonal complement of it, respectively. Note that from the orthogonality of the matrix Q_k, it follows that $P_k = Q_{k2} Q_{k2}^T$.

2.1. Second-order combined method. Let us put $g^{(k)} = f^{(1)}(x^{(k)})$. The function $f(x)$, in a neighborhood of $x^{(k)}$, is approximated by the following function:

$$f_{k2}(x) = f_{k2}(u_1, u_2) = f(x^{(k)}) + (P_k^{\perp} g^{(k)}, u_1) + (P_k g^{(k)}, u_2) + \frac{1}{2} H_{k\varepsilon}\big[(u_1)^2\big] + \frac{1}{2}\varepsilon Q_{k2} Q_{k2}^T\big[(u_2)^2\big], \tag{5}$$

which is obtained from the second-order Taylor approximation after replacing the matrix $E_{k\varepsilon}$ with the matrix $\varepsilon Q_{k2} Q_{k2}^T = \varepsilon P_k$, and using $x - x^{(k)} = u_1 + u_2$, $u_1 = P_k^{\perp}(x - x^{(k)})$, $u_2 = P_k(x - x^{(k)})$, $H_{k\varepsilon} u_2 = 0$, $P_k u_1 = 0$.

Then the vectors $u_1^{(k)}$, $u_2^{(k)}$ from (2) are determined as the minimum point of the function $f_{k2}(u_1, u_2)$. Therefore, they satisfy the following system of equations:

$$\frac{\partial f_{k2}(u_1, u_2)}{\partial u_1} = P_k^{\perp} g^{(k)} + H_{k\varepsilon} u_1 = 0, \tag{6}$$

$$\frac{\partial f_{k2}(u_1, u_2)}{\partial u_2} = P_k g^{(k)} + \varepsilon, u_2 = 0. \tag{7}$$

From the equation (6), it is deduced that $u_1 = -H_{k\varepsilon}^{+} P_k^{\perp} g^{(k)}$, where $H_{k\varepsilon}^{+}$ is a pseudoinverse matrix [5] of the matrix $H_{k\varepsilon}$. Therefore, from (3) and the orthogonality of the matrix Q_k, it follows that $H_{k\varepsilon}^{+} = (Q_{k1}^T)^{+} \Lambda_{k1}^{-1} (Q_{k1})^{+} = Q_{k1} \Lambda_{k1}^{-1} Q_{k1}^T$. Thus,

$$u_1^{(k)} = -H_{k\varepsilon}^{+} P_k^{\perp} g^{(k)} = -Q_{k1} \Lambda_{k1}^{-1} Q_{k1}^T P_k^{\perp} g^{(k)}. \tag{8}$$

Finally, the vector u_2 is easily obtained from the equation (7):

$$u_2^{(k)} = -\varepsilon^{-1} P_k g^{(k)}. \tag{9}$$

Note that the method described by (2), (8), (9) for solving the degenerate problem (1) is a combination of Newton's method and the method of steepest descent. At each kth iteration, the entire space $\mathbb{R}^n$ is represented as the direct sum of two subspaces $\mathrm{Ker}\,(H_{k\varepsilon})$ and the orthogonal complement of $\mathrm{Ker}\,(H_{k\varepsilon})$. The method of steepest descent is applied to $\mathrm{Ker}\,(H_{k\varepsilon})$, and Newton's method is applied to the orthogonal complement of $\mathrm{Ker}\,(H_{k\varepsilon})$. Moreover, the regularization parameter of the numerical method ε is used to divide the entire space into two orthogonal subspaces. This leads us to the problem of determining two step sizes a_{k1} and a_{k2} involved in (2).

2.2. Fourth-order combined method. This subsection considers an optimization technique that requires the third- and fourth-order derivatives information. Generally, calculation of higher-order derivatives is time consuming. From practical point of view, such methods are often not very useful despite their theoretical significance. However, if the kernel of the Hessian matrix is one-dimensional (that is, $\mathrm{rank}\,(H_{k\varepsilon}) = n - 1$), this approach becomes quite effective.

First, $f(k)$ is approximated in a neighborhood of $x^{(k)}$ by the following function:

$$f_{k4}(x) = f_{k4}(u_1, u_2) = f(x^{(k)}) + (P_k^{\perp} g^{(k)}, u_1) + (P_k g^{(k)}, u_2) + \frac{1}{2} H_{k\varepsilon}[(u_1)^2] + \frac{1}{2} E_k\big[(u_2)^2\big] + \frac{1}{2} f^{(3)}(x^{(k)})[u_1, (u_2)^2] + \frac{1}{6} f^{(3)}(x^{(k)})\big[(u_2)^3\big] + \frac{1}{24} f^{(4)}(x^{(k)})[(u_2)^4]. \tag{10}$$

Algorithm 1. Second-order combined method

Input: initial point $x^{(0)}$; regularization parameter $\varepsilon > 0$; algorithm of one-dimensional minimization OneDimMin (which is given in Subsection 2.3); algorithm of spectral decomposition of a matrix eigen (which is a standard function in R); maximum number of iterations K; by the gradient $\mathrm{tool\_grad} > 0$; accuracy by the argument $\mathrm{tool\_arg} > 0$; accuracy by the function $\mathrm{tool\_fun} > 0$.

initialization: $k = 0$; $f_0 = f(x^{(0)})$; $g^{(0)} = f^{(1)}(x^{(0)})$; $H_0 = f^{(2)}(x^{(0)})$;
while $\|g^{(k)}\| > \mathrm{tool\_grad}$:
    $(\Lambda_k, Q_k) =$ eigen (H_k, symmetric = TRUE), $\Lambda_k = \mathrm{diag}\,(\lambda_1^{(k)}, \ldots, \lambda_n^{(k)})$, $\lambda_i^{(k)} \geq \lambda_{i+1}^{(k)} \forall i$;
initialization: $r_k = 0$;
for $i = 0, \ldots, n$:
    if $|\lambda_i^{(k)}| > \varepsilon$ **then** $r_k = r_k + 1$;
    if $\lambda_i^{(k)} < -\varepsilon$ **then** $\lambda_i^{(k)} = -\lambda_i^{(k)}$;
if $r_k = n$ **then**
    $u^{(k)} = -Q_k \Lambda_k^{-1} Q_k^T g^{(k)}$;
    α_k is determined as a minimum point of the function of one variable $\varphi_1(\alpha) = f(x^{(k)} + \alpha u^{(k)})$ with the initial approximation $\alpha_0 = 1$
        $[\alpha_k, x^{(k+1)}, f_{k+1}] = \mathrm{OneDimMin}(x^{(k)}, f_k, u^{(k)}, 1)$;
else
    $Q_k = \begin{bmatrix} Q_{k1} & Q_{k2} \end{bmatrix}, \Lambda_k = \begin{bmatrix} \Lambda_{k1} & 0 \\ 0 & \Lambda_{k2} \end{bmatrix}$
    $u_1^{(k)} = -Q_{k1} \Lambda_{k1}^{-1} Q_{k1}^T g^{(k)}$, $u_2^{(k)} = -\varepsilon^{-1} Q_{k2} Q_{k2}^T g^{(k)}$.
    α_{k1} is determined as a minimum point of the function of one variable $\varphi_1(\alpha) = f(x^{(k)} + \alpha u_1^{(k)})$ with the initial approximation $\alpha_0 = 1$
        $[\alpha_{k1}, \hat{x}^{(k+1)}, \hat{f}_{k+1}] = \mathrm{OneDimMin}(x^{(k)}, f_k, u_1^{(k)}, 1)$, where $\hat{x}^{(k+1)} = x^{(k)} + \alpha_{k1} u_1^{(k)}$, $\hat{f}_{k+1} = f(\hat{x}^{(k+1)})$;
        α_{k2} is determined as a minimum point of the function of one variable $\varphi_2(\alpha) = f(x^{(k)} + \alpha_{k1} u_1^{(k)} + \alpha u_2^{(k)})$ with the initial approximation $\alpha_0 = \max\,(1, a_{(k-1),2})$;
        $[\alpha_{k2}, x^{(k+1)}, f_{k+1}] = \mathrm{OneDimMin}(\hat{x}^{(k+1)}, \hat{f}_{k+1}, u_2^{(k)}, \max\,(1, \alpha_{(k-1),2}))$;
if $\|x^{(k+1)} - x^{(k)}\| / (1 + \|x^{(k+1)}\|) \leq \mathrm{tool\_arg}$ **then break** with code $= 1$;
if $|f_{k+1} - f_k| / (1 + |f_{k+1}|) \leq \mathrm{tool\_fun}$ **then break** with code $= 2$;
$x^{(k)} = x^{(k+1)}$: $f_k = f_{k+1}$; $k = k + 1$; $g^{(k)} = f^{(1)}(x^{(k)})$;
if $\|g^{(k)}\| \leq \mathrm{tool\_grad}$ **then break** with code=0;
if $k > K$ **then break** with code $= 3$;
$H_k = f^{(2)}(x^{(k)})$;

Output: last $x^{(k)}$, f_k, $g^{(k)}$, k.

Approximation (10) is obtained by the fourth-order Taylor expansion

$$f(x) = f(x^{(k)}) + (g^{(k)}, x - x^{(k)}) + \frac{1}{2} f^{(2)}(x^{(k)}) \big[(x - x^{(k)})^2\big] + \frac{1}{6} f^{(3)}(x^{(k)}) \big[(x - x^{(k)})^3\big] + \frac{1}{24} f^{(4)}(x^{(k)}) \big[(x - x^{(k)})^4\big] + O\big(\|x - x^{(k)}\|^5\big),$$

and by using the fact that $x - x^{(k)} = u_1 + u_2$, $u_1 = P_k^{\perp}(x - x^{(k)})$, $u_2 = P_k(x - x^{(k)})$, $H_{k\varepsilon}u_2 = 0$, and $E_{k\varepsilon}u_1 = 0$. Ignore the terms $\frac{1}{6}f^{(3)}(x^{(k)})[(u_1)^3]$, $\frac{1}{2}f^{(3)}(x^{(k)})[(u_1)^2, u_2]$, $\frac{1}{24}f^{(4)}(x^{(k)})[(u_1)^4]$ and the terms of higher order $O(\|u_1\|^2)$, $O(\|u_1\|\|u_2\|^2)$, $O(\|u_2\|^4)$ to get (10).

Second, the vectors $u_1^{(k)}$, $u_2^{(k)}$ are defined as the minimum point of the function $f_{k4}(u_1, u_2)$. Thus, $u_1^{(k)}$, $u_2^{(k)}$ satisfy the system of equations:

$$\frac{\partial f_{k4}(u_1, u_2)}{\partial u_1} = P_k^{\perp} g^{(k)} + H_{k\varepsilon}u_1 + \frac{1}{2}P_k^{\perp} f^{(3)}(x^{(k)})\big[(u_2)^2\big] = 0, \tag{11}$$

$$\frac{\partial f_{k4}(u_1, u_2)}{\partial u_2} = P_k g^{(k)} + E_{k\varepsilon}u_2 + f^{(3)}(x^{(k)})[u_1, u_2] + \frac{1}{2}f^{(3)}(x^{(k)})\big[(u_2)^2\big] + \frac{1}{6}f^{(4)}(x^{(k)})\big[(u_2)^3\big] = 0. \tag{12}$$

Note that (11) is linear in u_1 and (12) is cubic in u_2 and can be easily simplified. Using (11), u_1 is expressed in terms of u_2:

$$u_1 = -H_{k\varepsilon}^{+} P_k^{\perp} g^{(k)} - \frac{1}{2}H_{k\varepsilon}^{+} f^{(3)}(x^{(k)})\big[(u_2)^2\big]. \tag{13}$$

Substituting the expression for u_1 from (13) into (12), we get

$$P_{(k)}g^{(k)} + E_{k\varepsilon}u_2 + P_k f^{(3)}(x^{(k)})\Big[\big(-H_{k\varepsilon}^{+} P_k^{\perp} g^{(k)} - \frac{1}{2}H_{k\varepsilon}^{+} f^{(3)}(x^{(k)})[(u_2)^2]\big), u_2\Big) + \frac{1}{2}P_k f^{(3)}(x^{(k)})[(u_2)^2] + \frac{1}{6}P_k f^{(4)}(x^{(k)})\big[(u_2)^3\big] = 0.$$

Finally, u_2 is determined from

$$P_k g^{(k)} + E_{k\varepsilon}u_2 - P_k f^{(3)}(x^{(k)})\big[(H_{k\varepsilon}^{+} P_k^{\perp} g^{(k)}), u_2\big] + \frac{1}{2}P_k f^{(3)}(x^{(k)})[(u_2)^2] + \frac{1}{6}P_k\big(f^{(4)}(x^{(k)}) - 3\big(f^{(3)}(x^{(k)})\big)^T H_{k\varepsilon}^{+} f^{(3)}(x^{(k)})\big)\big[(u_2)^3\big] = 0. \tag{14}$$

System (14) can be solved numerically, for example, by Newton's method. Note that the matrix coefficient of $(u_2)^3$ corresponds to the sufficient condition for a minimum of the 4th order (2.20) in [24].

Practical implementation of this method proved to be difficult in view of necessity to calculate the derivatives of the function $f(x)$ up to the fourth-order and to solve the system (14).

Consider the case of the one-dimensional kernel of the Hessian matrix in a neighborhood of the minimum point, that is, $\operatorname{rank}(H_{k\varepsilon}) = r_k = n - 1$.

Lemma 2.1. *Suppose that* $\operatorname{rank}(H_{k\varepsilon}) = r_k = n - 1$. *Then the solution of the system* (14) *can be represented in the form*

$$u_2 = \mu \cdot s \cdot q_k^{(n)}, \tag{15}$$

where $q_k^{(n)}$ *is the last* n*th column of the matrix* Q_k *given by* (3), $s = \operatorname{sign}\big(\big(g^{(k)}\big)^T q_k^{(n)}\big)$, $\mu < 0$ *satisfies the following cubic equation*

$$\alpha + b\mu + \frac{1}{2}c\mu^2 + \frac{1}{6}d\mu^3 = 0 \tag{16}$$

with the coefficients $\alpha = \big|(g^{(k)})^T q_k^{(n)}\big|$, $b = \lambda_n^{(k)} - (Q_{k1}\Lambda_{k1}^{-1}Q_{k1}^T g^{(k)})^T y^{(k)}$, $c = s\dfrac{\partial^3\varphi(0)}{\partial\alpha^3}$, $d = \left(\dfrac{\partial^4\varphi(0)}{\partial\alpha^4} - 3(y^{(k)})^T Q_{k1}\Lambda_{k1}^{-1}Q_{k1}^T y^{(k)}\right)$, *where* $\varphi(\alpha) = f\big(x^{(k)} + \alpha q_k^{(n)}\big)$, $y^{(k)} = \dfrac{\partial^2\theta(0)}{\partial\alpha^2}$, $\theta(\alpha) = f^{(1)}(x^{(k)} + \alpha q_k^{(n)})$.

Proof. Since $\operatorname{rank}(H_{k\varepsilon}) = n-1$, the dimension of $\operatorname{Ker}(H_{k\varepsilon})$ is equal to 1, and $Q_{k2} = q_k^{(n)}$. Moreover, the orthogonal projector onto the subspace $\operatorname{Ker}(H_{k\varepsilon})$ is $P_k = q_k^{(n)}\big(q_k^{(n)}\big)^T$ and $E_{k\varepsilon} = q_k^{(n)}\lambda_n^{(k)}\big(q_k^{(n)}\big)^T$.Thus, any vector u_2 that belongs to $\operatorname{Ker}(H_{k\varepsilon})$ is collinear with the vector $q_k^{(n)}$, and therefore u_2 can be represented in the form (15). In addition, if the constant μ is negative, then the descent direction condition will hold for the vector u_2:

$$\begin{aligned}(g^{(k)})^T u_2 = (P_k g^{(k)})^T u_2 &= \mu\operatorname{sign}\big((g^{(k)})^T q_k^{(n)}\big)(P_k g^{(k)})^T q_k^{(n)}\\ &= \mu\operatorname{sign}\big((g^{(k)})^T q_k^{(n)}\big)(g^{(k)})^T q_k^{(n)}\big(q_k^{(n)}\big)^T q_k^{(n)}\\ &= \mu\operatorname{sign}\big((g^{(k)})^T q_k^{(n)}\big)(g^{(k)})^T q_k^{(n)} = \mu\big|(g^{(k)})^T q_k^{(n)}\big| < 0.\end{aligned}$$

Thus, it remains to determine the constant $\mu < 0$. Substitute u_2, given by (15) into (14) to get

$$\begin{aligned}P_k g^{(k)} + \mu s E_{k\varepsilon} q_k^{(n)} - \mu s P_k f^{(3)}(x^{(k)})\big[(H_{k\varepsilon}^+ P_k^\perp g^{(k)}), q_k^{(n)}\big] + \frac{1}{2}\mu^2 P_k f^{(3)}(x^{(k)})\big[\big(q_k^{(n)}\big)^2\big]\\ + \frac{1}{6}\mu^3 s P_k\Big(f^{(4)}(x^{(k)}) - 3\big(f^{(3)}(x^{(k)})\big)^T H_{k\varepsilon}^+ f^{(3)}(x^{(k)})\Big)\big[(q_k^{(n)})^3\big] = 0.\end{aligned} \tag{17}$$

Performing scalar product on both sides of the system (17) with the same vector $sq_k^{(n)}$ gives the following cubic equation for the scalar $\mu < 0$:

$$a + b\mu + \frac{1}{2}c\mu^2 + \frac{1}{6}d\mu^3 = 0, \tag{18}$$

where $a = \big|(P_k g^{(k)})^T q_k^{(n)}\big| = \big|(g^{(k)})^T q_k^{(n)}\big|$, $b = \lambda_n^{(k)} - f^{(3)}(x^{(k)})\big[(H_{k\varepsilon}^+ P_k^\perp g^{(k)}), \big(q_k^{(n)}\big)^2\big]$, $c = sf^{(3)}(x^{(k)})\big[\big(q_k^{(n)}\big)^3\big]$, $d = (f^{(4)}(x^{(k)}) - 3\big(f^{(3)}(x^{(k)})\big)^T H_{k\varepsilon}^+ f^{(3)}(x^{(k)})\big)\big[(q_k^{(n)})^4\big]$. The coefficient d in (18) must be positive when the sufficient condition for a minimum of the 4th order (2.20) in [24] is satisfied.

Note that the value of the multilinear form $f^{(4)}(x^{(k)})\big[(q_k^{(n)})^4\big]$ is equal to the value of the fourth directional derivative of the function $f(x)$ at the point $x^{(k)}$ in the direction of the vector $q_k^{(n)}$, that is, $f^{(4)}(x^{(k)})\big[\big(q_k^{(n)}\big)^4\big] = \dfrac{\partial^4\varphi(0)}{\partial\alpha^4}$, where $\varphi(\alpha) = f\big(x^{(k)} + \alpha q_k^{(n)}\big)$, $\alpha \in \mathbb{R}^1$. Similarly, the value of the multilinear form $f^{(3)}(x^{(k)})\big[\big(q_k^{(n)}\big)^3\big]$ is equal to the value of the third directional derivative of the function $f(x)$ at the point $x^{(k)}$ in the direction of $q_k^{(n)}$, that is, $f^{(3)}(x^{(k)})\big[\big(q_k^{(n)}\big)^3\big] = \dfrac{\partial^3\varphi(0)}{\partial\alpha^3}$.

Therefore, let us denote

$$y^{(k)} = f^{(3)}(x^{(k)})\big[\big(q_k^{(n)}\big)^2\big]. \tag{19}$$

Then

$$b = \lambda_n^{(k)} - (H_{k\varepsilon}^+ P_k^\perp g^{(k)})^T y^{(k)} = \lambda_n^{(k)} - (Q_{k1}\Lambda_{k1}^{-1} Q_{k1}^T g^{(k)})^T y^{(k)}, \quad c = s\frac{\partial^3 \varphi(0)}{\partial \alpha^3},$$

$$d = \left(\frac{\partial^4 \varphi(0)}{\partial \alpha^4} - 3(y^{(k)})^T H_{k\varepsilon}^+ y^{(k)}\right) = \left(\frac{\partial^4 \varphi(0)}{\partial \alpha^4} - 3(y^{(k)})^T Q_{k1}\Lambda_{k1}^{-1} Q_{k1}^T y^{(k)}\right).$$

The lemma is proved.

Lemma 2.2. *The vector* u_1, *as a component of the solution* $u = u_1 + u_2$ *of the system* (11), (12), *can be computed using the formula*

$$u_1 = -Q_{k1}\Lambda_{k1}^{-1} Q_{k1}^T \left(g^{(k)} + \frac{\mu^2}{2} y^{(k)}\right), \tag{20}$$

where $\mu < 0$ *is a solution of the cubic equation* (16), $y^{(k)}$ *is given by* (19).

Proof. By using (13) and (15), we get

$$u_1 = -H_{k\varepsilon}^+ P_k^\perp g^{(k)} - \frac{\mu^2}{2} H_{k\varepsilon}^+ f^{(3)}(x^{(k)})\left[\left(q_k^{(n)}\right)^2\right] = -H_{k\varepsilon}^+ P_k^\perp g^{(k)} - \frac{\mu^2}{2} H_{k\varepsilon}^+ y^{(k)}$$

$$= -H_{k\varepsilon}^+ \left(P_k^\perp g^{(k)} + \frac{\mu^2}{2} y^{(k)}\right) = -Q_{k1}\Lambda_{k1}^{-1} Q_{k1}^T \left(g^{(k)} + \frac{\mu^2}{2} y^{(k)}\right).$$

So, equality (20) holds.

Formulas (15), (16), and (20) contain the derivatives $\dfrac{\partial^4 \varphi(0)}{\partial \alpha^4}$, $\dfrac{\partial^3 \varphi(0)}{\partial \alpha^3}$, and also $f^{(3)}(x^{(k)})\left[\left(q_k^{(n)}\right)^2\right]$. In practical implementation of the fourth-order method, these derivatives can be calculated numerically, for example, by the formulas

$$\frac{\partial^4 \varphi(0)}{\partial \alpha^4} \approx \frac{\varphi_0(2h) - 4\varphi_0(h) + 6\varphi_0(0) - 4\varphi_0(-h) + \varphi_0(-2h)}{h^4},$$

$$\frac{\partial^3 \varphi(0)}{\partial \alpha^3} \approx \frac{\varphi_0(2h) - 2\varphi_0(h) + 2\varphi_0(-h) - \varphi_0(-2h)}{2h^3},$$

where $h > 0$ is some small number called step size. Recall that the vector $y^{(k)} = f^{(3)}(x^{(k)})\left[(q_k^{(n)})^2\right]$ is, in fact, the second directional derivative of the vector function $f^{(1)}(x)$ at the point $x^{(k)}$ in the direction of $q_k^{(n)}$. So, $f^{(3)}(x^{(k)})\left[\left(q_k^{(n)}\right)^2\right] = \dfrac{\partial^2 \theta(0)}{\partial \alpha^2}$, where $\theta(\alpha) = f^{(1)}(x^{(k)} + \alpha q_k^{(n)})$. Then the elements $y_i^{(k)}$, $i = 1, \ldots, n$, of the vector $y^{(k)}$ can be calculated numerically, for example, by one of two formulas $y_i^{(k)} \approx \dfrac{\theta_i(h) - 2\theta_i(0) + \theta_i(-h)}{h^2}$, $y^{(k)} \approx \dfrac{f^{(1)}(x^{(k)} + hq_k^{(n)}) - 2f^{(1)}(x^{(k)}) + f^{(1)}(x^{(k)} - hq_k^{(n)})}{h^2}$.

2.3. Algorithms for determining Step Sizes. First, consider the second-order combined method described by (2), (8), and (9) for solving the degenerate problem (1). We propose an algorithm to determine the step sizes $\alpha_{k1} > 0$ and $\alpha_{k2} > 0$ involved in (2) for the combined second-order method. This algorithm is based on the idea similar to the Hooke – Jeeves method for solving unconstrained optimization problems.

Initially, the step size $\hat{\alpha}_{k1} > 0$ is defined as a minimum point of the function of one variable $\varphi_1(\alpha) = f(x^{(k)} + \alpha u_1^{(k)})$. Secondly, the step size $\hat{\alpha}_{k2} > 0$ is defined as a minimum point of the function of one variable $\varphi_2(\alpha) = f(x^{(k)} + \hat{\alpha}_{k1} u_1^{(k)} + \alpha u_2^{(k)})$. Afterwards, the step size $\alpha_k > 0$ is

Algorithm 2. Fourth-order combined method

Input: initial point x^0; regularization parameter $\varepsilon > 0$; algorithm one-dimensional minimization OneDimMin (which is given in Subsection 2.3); algorithm of spectral decomposition of a matrix eigen (which is a standard function in R); maximum number of iterations K; accuracy by the gradient $tool\_grad > 0$; accuracy by the argument $\mathrm{tool\_arg} > 0$; accuracy by the function $\mathrm{tool\_fun} > 0$.

Initialization: $k = 0$; $f_0 = f(x^{(0)})$; $g^{(0)} = f^{(1)}(x^{(0)})$; $H_0 = f^{(2)}(x^{(0)})$;
while $\|g^{(k)}\| > \mathrm{tool\_grad}$:
    $(\Lambda_k, Q_k) = \mathrm{eigen}\,(H_k,\ \mathrm{symmetric} = \mathrm{TRUE})$, $\Lambda_k = \mathrm{diag}\,(\lambda_1^{(k)}, \ldots, \lambda_n^{(k)})$, $\lambda_i^{(k)} \geq \lambda_{i+1}^{(k)}$;
initialization: $r_k = 0$;
for $i = 0, \ldots, n$:
    if $|\lambda_i^{(k)}| > \varepsilon$ **then** $r_k = r_k + 1$;
    if $\lambda_i^{(k)} < -\varepsilon$ **then** $\lambda_i^{(k)} = -\lambda_i^{(k)}$;
if $r_k = n$ **then**
    $u^{(k)} = -Q_k \Lambda_k^{-1} Q_k^T g^{(k)}$;
else
    $Q_k = \begin{bmatrix} Q_{k1} & Q_{k2} \end{bmatrix}$, $\Lambda_k = \begin{bmatrix} \Lambda_{k1} & 0 \\ 0 & \Lambda_{k2} \end{bmatrix}$
    calculate arrays $f^{(3)}(x^{(k)})$, $f^{(4)}(x^{(k)})$;
    $u_2^{(k)}$ is determined as a solution of system (14);
    $u_1^{(k)}$ is calculated by (13);
    $u^{(k)} = u_1^{(k)} + u_2^{(k)}$;
a_k is determined as a minimum point of the function of one variable $\varphi_3(\alpha) = f(x^{(k)} + \alpha u^{(k)})$ with the initial approximation $\alpha_0 = 1$
$[\alpha_k, x^{(k+1)}, f_{k+1}] = \mathrm{OneDimMin}(x^{(k)}, f_k, u^{(k)}, 1)$;
if $\|x^{(k+1)} - x^{(k)}\|/(1 + \|x^{(k+1)}\|) \leq \mathrm{tool\_arg}$ **then break** with code=1;
if $|f_{k+1} - f_k|/(1 + |f_{k+1}|) \leq \mathrm{tool\_fun}$ **then break** with code $= 2$;
$x^{(k)} = x^{(k+1)}$: $f_k = f_{k+1}$; $k = k + 1$; $g^{(k)} = f^{(1)}(x^{(k)})$;
if $\|g^{(k)}\| \leq \mathrm{tool\_grad}$ **then break** with code=0;
if $k > K$ **then break** with code $= 3$;
$H_k = f^{(2)}(x^{(k)})$;

Output: last $x^{(k)}, f_k, g^{(k)}, k$.

determined as a minimum point of the function of one variable $\varphi_3(\alpha) = f(x^{(k)} + \alpha u^{(k)})$, where $u^{(k)} = \hat{\alpha}_{k1} u_1^{(k)} + \hat{\alpha}_{k2} u_2^{(k)}$.

These one-dimensional minimization problems for the functions $\varphi_i(\alpha)$, $i = 1, 2, 3$, may be solved numerically, according to the following algorithm. First, suppose some initial value $\alpha = \alpha_0$ is chosen. If $\varphi_i(\alpha) < \varphi_i(0)$, then α doubles until the function $\varphi_i(\alpha)$ decreases. If $\varphi_i(\alpha) > \varphi_i(0)$, then α is halved until the condition $\varphi_i(\alpha) \leq \varphi_i(0)$ is satisfied. Then the approximate minimum point of the function $\varphi_i(\alpha)(i = 1, 2)$ is calculated as the minimum point of the quadratic function constructed from the last three best points (provided that the values at these points are different), and the function $\varphi_3(\alpha)$ is calculated as the local minimum point of the cubic function constructed from the four last best points.

Algorithm 3. One-dimensional minimization

Input: $x^{(k)}$ is a current point; f_k is the value of $f(x)$ at the point $x^{(k)}$; $u^{(k)}$ is a search direction; initial value of the step size $\rho > 0$.

function definition $\varphi(\alpha) = f(x^{(k)} + \alpha u^{(k)})$;
initialization: $\alpha_0 = 0; \varphi_0 = f_k$;
$\alpha_1 = \rho; \varphi_1 = \varphi(\alpha_1)$;
if $\varphi_1 < \varphi_0$ **then**
    $\alpha_2 = 2\alpha_1; \varphi_2 = \varphi(\alpha_2)$;
    while $\varphi_1 < \varphi_0$:
        $\alpha_0 = \alpha_1; \varphi_0 = \varphi_1$;
        $\alpha_1 = \alpha_2; \varphi_1 = \varphi_2$;
        $\alpha_2 = 2\alpha_1; \varphi_2 = \varphi(\alpha_2)$;
else:
    while $\varphi_1 > \varphi_0$:
        $\alpha_2 = \alpha_1; \varphi_2 = \varphi_1$;
        $\alpha_1 = \alpha_1/2; \varphi_1 = \varphi(\alpha_1)$;
a_3 is determined as a minimum point of the quadratic function that fits to the following three points $(a_0, \varphi_0), (a_1, \varphi_1), (a_2, \varphi_2)$;
$\varphi_3 = \varphi(\alpha_3)$;
if $\varphi_3 < \varphi_1$ **then**
    $\alpha_1 = \alpha_3; \varphi_1 = \varphi_3$;

Output: a_1, $x^{(k+1)} = x^{(k)} + \alpha_1 u^{(k)}$, $f_{k+1} = \varphi_1$.

To find the approximate minimum of the function $\varphi_1(\alpha)$, the initial value $\alpha_0 = 1$ is taken. To minimize $\varphi_2(\alpha)$, it is possible to take the initial value $\alpha_0 = \max(1, \alpha_{(k-1),2})$, where $\alpha_{(k-1),2}$ is the value of the step size α_{k2}, obtained at the previous iteration.

Such an algorithm for choosing sizes of the steps $\alpha_{k1} > 0$ and $\alpha_{k2} > 0$ can be interpreted as follows. When the minimum point degenerates, in its neighborhood the objective function becomes "ravine", that is, across the "ravine" it is steep, and along the "ravine" it is sloping. Therefore, along the "ravine" (that is, in the subspace $\operatorname{Ker}(H_{k\varepsilon})$), it is necessary to take the largest possible step, while across the "ravine" (that is, in the orthogonal complement of $\operatorname{Ker}(H_{k\varepsilon})$), the step is most likely close to unity.

Now consider the fourth-order combined method defined by (2), (13), and (14) for solving the degenerate problem (1). In this method, the step sizes α_{k1} and α_{k2}, involved in (2), can be taken equal to each other. So, the iterative formula (2) becomes

$$x^{(k+1)} = x^{(k)} + \alpha_k(u_1^{(k)} + u_2^{(k)}), \quad k = 0, 1, 2, \ldots. \tag{21}$$

Then the step size $\alpha_k > 0$ is found as an approximate minimum point of the function of one variable $\varphi_3(\alpha) = f\big(x^{(k)} + \alpha(u_1^{(k)} + u_2^{(k)})\big)$ according to the algorithm described above for finding the minimum point of the function $\varphi_1(\alpha)$. The initial value for α is 1.

2.4. Analysis of the convergence rate of methods. To analyze the convergence rate of the methods, it is convenient to use the generalized necessary and sufficient conditions for a minimum of a degenerate unconstrained optimization problem proposed in [24]. The following two theorems were proved in [24]. We recall that $(f^{(2)}(x^*))^+$ is a pseudoinverse matrix [5] of $f^{(2)}(x^*)$, P is an

orthogonal projector onto the subspace R_1; $P^{\perp}$ is an orthogonal projector onto the subspace R_2, then $P = I - (f^{(2)}(x^*))^+ f^{(2)}(x^*)$, $P^{\perp} = (f^{(2)}(x^*))^+ f^{(2)}(x^*)$.

Theorem 2.1 (generalized necessary conditions for a minimum)**.** *Let $f(x)$ have a minimum at $x^* \in \mathbb{R}^n$. Suppose that $f(x)$ is p times ($p \geq 4, p$ is even) continuously differentiable in some neighborhood $V(x^*)$ of the point x^* and, for all $u \in \mathbb{R}^n$, the following condition holds*:

$$f^{(2l)}(x^*)[(Pu)^{2l}] = 0, \quad \textit{where} \quad l = 1, \ldots, \frac{p}{2} - 1. \tag{22}$$

Then, for all $u \in \mathbb{R}^n$, the following is true:

$$f^{(1)}(x^*) = 0, \quad f^{(2)}(x^*)[u^2] \geq 0, \tag{23}$$

$$f^{(2)}(x^*)[(P^{\perp}u)^2] \geq m_2 \|P^{\perp}u\|^2, \tag{24}$$

$$f^{(2l+1)}(x^*)[(Pu)^{2l+1}] = 0, \quad \textit{where} = 1, \ldots, \frac{p}{2} - 1, \quad f^{(p)}(x^*)[(Pu)^p] \geq 0, \tag{25}$$

$$f^{(l+1)}(x^*)\big[(P^{\perp}u), (Pu)^l\big] = 0, \quad \textit{where} \quad l = 1, \ldots, \frac{p}{2} - 1, \tag{26}$$

$$\left(f^{(p)}(x^*) - \frac{p!}{2\Big(\Big(\frac{p}{2}\Big)!\Big)^2}(f^{(\frac{p}{2}+1)}(x^*))^T (f^{(2)}(x^*))^+ (f^{(\frac{p}{2}+1)}(x^*))\right)[(Pu)^p] \geq 0,$$

where $m_2 > 0$.

Theorem 2.2 (generalized sufficient conditions for a minimum)**.** *Suppose that $f(x)$ is p times ($p \geq 4, p$ is even) continuously differentiable in some neighborhood $V(x^*)$ of x^*, at which the conditions* (22)–(26) *are satisfied. In addition, assume that for all $u \in \mathbb{R}^n$, the following estimate holds*:

$$\left(f^{(p)}(x^*) - \frac{p!}{2\Big(\Big(\frac{p}{2}\Big)!\Big)^2}(f^{(\frac{p}{2}+1)}(x^*))^T (f^{(2)}(x^*))^+ (f^{(\frac{p}{2}+1)}(x^*))\right)[(Pu)^p] \geq m_p \|Pu\|^p, \tag{27}$$

where $m_p > 0$. Then x^* is a point where $f(x)$ has a strict local minimum. Moreover, for all x in some sufficiently small neighborhood of x^*, the following inequality is fulfilled:

$$f(x) - f(x^*) \geq m_0(\|P^{\perp}v\|^2 + \|Pv\|^p), \tag{28}$$

where $v = x - x^*, m_0 > 0$.

The following lemma is also needed for the further analysis.

Lemma 2.3. *Suppose that the function $f(x)$ satisfies the conditions of Theorem* 2.2 *except the condition* (27)*. Assume that for all $u \in \mathbb{R}^n$, the following condition is fulfilled*:

$$\left(f^{(p)}(x^*) - \frac{(p/2+1)^2(p-1)!}{4\Big(\Big(\frac{p}{2}\Big)!\Big)^2}(f^{(\frac{p}{2}+1)}(x^*))^T (f^{(2)}(x^*))^+ (f^{(\frac{p}{2}+1)}(x^*))\right)[(Pu)^p] \geq \bar{m}_p \|Pu\|^p, \tag{29}$$

where $\bar{m}_p > 0$. *Then the condition* (27) *is true, and for all* x *from some sufficiently small neighborhood* $V(x^*)$, *in addition to the estimate* (28), *the following estimate holds*:

$$\big(f^{(1)}(x)\big)^T(x-x^*) \geq m_1(\|P^{\perp}v\|^2 + \|Pv\|^p), \tag{30}$$

where $v = x - x^*, m_1 > 0$.

Proof. First, let us show that the inequality (27) follows from the inequality (29). Consider the multilinear form $(f^{(\frac{p}{2}+1)}(x^*))^T(f^{(2)}(x^*))^+(f^{(\frac{p}{2}+1)}(x^*))[(Pu)^p]$, which appears in the conditions (27) and (29), and prove that its value is nonnegative for all $u \in \mathbb{R}^n$.

Let $z = f^{(\frac{p}{2}+1)}(x^*)[(Pu)^{p/2}]$. Note that $z \in \mathbb{R}^n$; therefore,

$$(f^{(\frac{p}{2}+1)}(x^*))^T(f^{(2)}(x^*))^+(f^{(\frac{p}{2}+1)}(x^*))[(Pu)^p] = (f^{(2)}(x^*))^+[z^2] \geq \frac{1}{\lambda_{\max}}\|P^{\perp}z\|^2 \geq 0,$$

where $\lambda_{\max}$ is the largest eigenvalue of the matrix $f^{(2)}(x^*)$.

The coefficient $\dfrac{(p/2+1)^2(p-1)!}{4\Big(\Big(\dfrac{p}{2}\Big)!\Big)^2}$ in the inequality (29) is greater than the coefficient $\dfrac{p!}{2\Big(\Big(\dfrac{p}{2}\Big)!\Big)^2}$ for all $p \geq 4$. Thus, if the inequality (29) is satisfied, then (27) is also true.

Since the function $f(x)$ is p times continuously differentiable on the neighborhood $V(x^*)$ of the point x^*, using the Taylor series expansion for all $x \in V(x^*)$, it follows that

$$f^{(1)}(x) = f^{(1)}(x^*) + \sum_{l=2}^{p}\frac{1}{(l-1)!}f^{(l)}(x^*)[(v)^{l-1}] + O(\|v\|^p),$$

where $v = x - x^*$. Then, for the sufficiently small neighborhood $V(x^*)$, taking into account the conditions (22) – (26) it is deduced that

$$\begin{aligned}
&\big(f^{(1)}(x)\big)^T(x-x^*)\\
&\quad= f^{(2)}(x^*)[(P^{\perp}v)^2] + \sum_{l=3}^{p}\frac{1}{(l-1)!}\sum_{i=0}^{l}C_l^i f^{(l)}(x^*)\big[(P^{\perp}v)^{l-i},(Pv)^l\big] + O(\|v\|^{p+1})\\
&\quad= f^{(2)}(x^*)[(P^{\perp}v)^2] + \frac{p/2+1}{\big(\frac{p}{2}\big)!}f^{(\frac{p}{2}+1)}(x^*)\big[(P^{\perp}v),(Pv)^{p/2}\big] + \frac{1}{(p-1)!}f^{(p)}(x^*)[(Pv)^p]\\
&\qquad+ O\big(\|P^{\perp}v\|^3\big) + O\big(\|P^{\perp}v\|\|Pv\|^{\frac{p}{2}+1}\big) + O\big(\|P^{\perp}v\|^2\|Pv\|^{\frac{p}{2}}\big) + O\big(\|v\|^{p+1}\big).
\end{aligned}$$

Thus,

$$\begin{aligned}
\big(f^{(1)}(x)\big)^T(x-x^*) &= f^{(2)}(x^*)\left[\left(P^{\perp}v + \frac{p/2+1}{2\big(\frac{p}{2}\big)!}(f^{(2)}(x^*))^+(f^{(\frac{p}{2}+1)}(x^*))[(Pv)^{\frac{p}{2}}]\right)^2\right]\\
&+ \frac{1}{(p-1)!}\left(f^{(p)}(x^*) - \frac{(p/2+1)^2(p-1)!}{4\Big(\Big(\frac{p}{2}\Big)!\Big)^2}(f^{(\frac{p}{2}+1)}(x^*))^T(f^{(2)}(x^*))^+(f^{(\frac{p}{2}+1)}(x^*))\right)[(Pv)^p]
\end{aligned}$$

$$+ O\big(\|P^{\perp}v\|^3\big) + O\big(\|P^{\perp}v\|\|Pv\|^{\frac{p}{2}+1}\big) + O\big(\|P^{\perp}v\|^2\|Pv\|^{\frac{p}{2}}\big) + O\big(\|v\|^{p+1}\big).$$

From (24) and (29) it is straightforward that

$$\begin{aligned}
&\big(f^{(1)}(x)\big)^T(x-x^*)\\
&\quad\geq m_2\left\|P^{\perp}v + \frac{p/2+1}{2\Big(\frac{p}{2}\Big)!}(f^{(2)}(x^*))^{+}(f^{(\frac{p}{2}+1)}(x^*))[(Pv)^{\frac{p}{2}}]\right\|^2 + \frac{\bar{m}_{(p)}}{(p-1)!}\|Pu\|^p\\
&\qquad - N_1\|P^{\perp}v\|^3 - N_2\|P^{\perp}v\|\|Pv\|^{\frac{p}{2}+1} - N_3\|P^{\perp}v\|^2\|Pv\|^{\frac{p}{2}} - N_4\|v\|^{p+1}, \qquad (31)
\end{aligned}$$

where $N1, N2, N3$, and $N4$ are some positive constants.

Let us consider $x \in V(x^*)$ such that $\|P^{\perp}v\| \geq \dfrac{p/2+1}{2\Big(\frac{p}{2}\Big)!}\|(f^{(2)}(x^*))^{+}\|\left\|f^{(\frac{p}{2}+1)}(x^*)\right\|\|Pv\|^{\frac{p}{2}}$, then

$$\begin{aligned}
&\left\|P^{\perp}v + \frac{\frac{p}{2}+1}{2\Big(\frac{p}{2}\Big)!}(f^{(2)}(x^*))^{+}(f^{(\frac{p}{2}+1)}(x^*))\big[(Pv)^{\frac{p}{2}}\big]\right\|\\
&\qquad\geq \|P^{\perp}v\| - \left\|\frac{\frac{p}{2}+1}{2\Big(\frac{p}{2}\Big)!}(f^{(2)}(x^*))^{+}(f^{(\frac{p}{2}+1)}(x^*))\big[(Pv)^{\frac{p}{2}}\big]\right\| \geq \frac{1}{2}\|P^{\perp}v\|,
\end{aligned}$$

Therefore, for a sufficiently small subset of $V(x^*)$, the following estimation holds:

$$\begin{aligned}
\big(f^{(1)}(x)\big)^T(x-x^*) &\geq \frac{1}{2}\left(\frac{m_2}{4}\|P^{\perp}v\|^2 + \frac{\bar{m}_p}{(p-1)!}\|Pv\|^p\right)\\
&\geq \min\left(\frac{m_2}{8}, \frac{\bar{m}_p}{2(p-1)!}\right)(\|P^{\perp}v\|^2 + \|Pv\|^p). \qquad (32)
\end{aligned}$$

Now choose $x \in V(x^*)$ such that $\|P^{\perp}v\| < \dfrac{\frac{p}{2}+1}{2\Big(\frac{p}{2}\Big)!}\|(f^{(2)}(x^*))^{+}\|\left\|f^{(\frac{p}{2}+1)}(x^*)\right\|\|[(Pv)]\|^{\frac{p}{2}}$ then $\|Pv\|^{\frac{p}{2}} > \dfrac{2\Big(\frac{p}{2}\Big)!}{\frac{p}{2}+1}\|(f^{(2)}(x^*))^{+}\|^{-1}\left\|f^{(\frac{p}{2}+1)}(x^*)\right\|^{-1}\|P^{\perp}v\|$. Moreover, from (24) it is clear that $\|(f^{(2)}(x^*))^{+}\| \leq \dfrac{1}{m_2}$, which gives $\|Pv\|^{\frac{p}{2}} > \dfrac{2\Big(\frac{p}{2}\Big)!}{\frac{p}{2}+1}\|(f^{(2)}(x^*))^{+}\|^{-1}\left\|f^{(\frac{p}{2}+1)}(x^*)\right\|^{-1}\|P^{\perp}v\| \geq \dfrac{2\Big(\frac{p}{2}\Big)!}{\frac{p}{2}+1}m_2\left\|f^{(\frac{p}{2}+1)}(x^*)\right\|^{-1}\|P^{\perp}v\|$. Then, from (31), for a sufficiently small subset of $V(x^*)$, the following is obtained:

$$\big(f^{(1)}(x)\big)^T(x-x^*) \geq \frac{1}{2}\frac{\bar{m}_p}{(p-1)!}\|Pv\|^p$$

$$\geq \frac{1}{4}\frac{\bar{m}_p}{(p-1)!}\|Pv\|^p + \frac{1}{4}\frac{\bar{m}_p}{(p-1)!}\left(\frac{2\left(\frac{p}{2}\right)!}{\frac{p}{2}+1}m_2\right)^2 \left\|f^{(\frac{p}{2}+1)}(x^*)\right\|^{-2}\|P^{\perp}v\|^2$$

$$\geq \min\left(\frac{\bar{m}_p}{4(p-1)!}, \frac{1}{4}\frac{\bar{m}_p}{(p-1)!}\left(\frac{2\left(\frac{p}{2}\right)!}{\frac{p}{2}+1}m_2\right)^2 \left\|f^{(\frac{p}{2}+1)}(x^*)\right\|^{-2}\right)(\|P^{\perp}v\|^2 + \|Pv\|^p). \tag{33}$$

Note that this implies the condition $\left\|f^{(\frac{p}{2}+1)}(x^*)\right\| > 0$. In the case of $\left\|f^{(\frac{p}{2}+1)}(x^*)\right\| = 0$, that is, $f^{(\frac{p}{2}+1)}(x^*) = 0$, from (31), for a sufficiently small subset of $V(x^*)$, the following is obtained:

$$\left(f^{(1)}(x)\right)^T(x-x^*) \geq \frac{1}{2}\left(m_2\|P^{\perp}v\|^2 + \frac{\bar{m}_p}{(p-1)!}\|Pv\|^p\right)$$

$$\geq \min\left(\frac{m_2}{2}, \frac{\bar{m}_p}{2(p-1)!}\right)(\|P^{\perp}v\|^2 + \|Pv\|^p). \tag{34}$$

Thus, according to (32)–(34), for all x in a sufficiently small deleted neighborhood of x^*, there is a positive constant m_1 such that the inequality (30) is satisfied.

***Corollary* 2.1.** *Suppose that the function $f(x)$ is p times ($p \geq 4, p$ is even) continuously differentiable in the neighborhood $V(x^*)$ of the point x^*, at which conditions* (22)–(26) *are satisfied. Let $f^{(2)}(x^*)$ be equal to the zero matrix, and for all $u \in \mathbb{R}^n$, the following holds*:

$$f^{(p)}(x^*)(Pu)^p \geq m_p\|Pu\|^p, \tag{35}$$

where $m_p > 0$. Then x^* is a point of strict local minimum of $f(x)$ and, for all x in a sufficiently small neighborhood $V(x^*)$, the following inequality is fulfilled:

$$f(x) - f(x)^* \geq m_0\|Pv\|^p,$$

$$\left(f^{(1)}(x)\right)^T(x-x^*) \geq m_1\|Pv\|^p, \tag{36}$$

where $v = x - x^*, m_0 > 0, m_1 > 0$.

The proof follows from Theorem 2.2 *and Lemma* 2.3, *since in this case the projector $P^{\perp} = 0$, and the projector $P = I, (f^{(2)}(x^*))^+ = 0$.*

Now let us analyze the convergence rate of the fourth-order combined method given by (21), (13) and (14) wherein it is considered $a_k = 1$. For ease of analysis, let us consider the following approach.

The combined fourth-order method described by (21), (13) and (14) is a combination of Newton's method and the method that requires the fourth-order derivatives. At each kth iteration, the entire space $\mathbb{R}^n$ is represented as the direct sum of two subspaces $\mathrm{Ker}\,(H_{k\varepsilon})$ and the orthogonal complement of it. The fourth-order method is applied to $\mathrm{Ker}\,(H_{k\varepsilon})$ and Newton's method is applied to the orthogonal complement of it. The convergence rate of the combined method is determined by the performance of the slowest method of the original two methods. It is well known that Newton's method in the case of nondegenerate matrix $f^{(2)}(x^*)$ has a quadratic convergence rate. Moreover, the sufficient condition for a minimum is in the form of inequality (24) (where $P^{\perp} = I$).

It is useful to discuss the fourth-order method given by (21), (13) and (14) in the case of the matrix $f^{(2)}(x^*)$ that is equal to the zero matrix. This simplifies significantly the study of the convergence rate. In this case, the matrix $H_{k\varepsilon}$ is equal to the zero matrix near the minimum point x^*, that is, the projector $P_k^{\perp} = 0$, the projector $P_k = I$, and the vector $u_1 = 0$. Therefore, (21), (13) and (14) are simplified.

Approximate the function $f(x)$, for every x in a neighborhood of the point $x^{(k)}$, by the following function

$$f_{k4}(x) = f_k(u_2) = f(x^{(k)}) + (P_k g^{(k)}, u_2) + \frac{1}{2}E_k\big[(u_2)^2\big] + \frac{1}{6}f^{(3)}(x^{(k)})\big[(u_2)^3\big] + \frac{1}{24}f^{(4)}(x^{(k)})[(u_2)^4], \tag{37}$$

which is obtained using the fourth-order Taylor expansion, where $x - x^{(k)} = u_2$.

The vector $u_2^{(k)}$ is defined as the minimum point of the function $f_{k4}(u_2)$. So, it satisfies the following system of equations

$$\frac{\partial f_{k4}(u_2)}{\partial u_2} = P_k g^{(k)} + E_{k\varepsilon}u_2 + \frac{1}{2}P_k f^{(3)}(x^{(k)})\big[(u_2)^2\big] + \frac{1}{6}P_k f^{(4)}(x^{(k)})\big[(u_2)^3\big] = 0. \tag{38}$$

However, the function $f_{k4}(x)$ given by (37) is a fourth-order function and its Hessian also must be equal to zero at the minimum point. Then the necessary condition for its minimum point is that the third derivative is equal to zero. Thus, near the minimum point, the vector $u_2^{(k)}$ can be determined from the system

$$\frac{\partial^3 f_{k4}(u_2)}{\partial u_2^3} = P_k f^{(3)}(x^{(k)})\big[(u_2)^2\big] + P_k f^{(4)}(x^{(k)})\big[(u_2)^3\big] = 0. \tag{39}$$

Theorem 2.3. *Suppose that $f(x)$ has a local minimum at the point $x^* \in \mathbb{R}^n$. Let $f(x)$ be* 4 *times continuously differentiable in the neighborhood $V(x^*)$ of x^*. Let the matrix $f^{(2)}(x^*)$ be equal to the zero matrix and, for all $u \in \mathbb{R}^n$, condition* (35) *is satisfied for $p = 4$. Then the fourth-order method given by* (21) *and* (38) *converges in a neighborhood of x^* and the rate of convergence is* 4/3.

Proof. Recall that the vector $u_2^{(k)}$ is found as the minimum point of the function $f_{k4}(x)$ given by (37) and, taking $a_k = 1$ in (21), it is clear that $x^{(k+1)} - x^{(k)} = u_2^{(k)}$ and $f_{k4}^{(1)}(x^{(k+1)}) = 0$. Therefore,

$$\big(f^{(1)}(x^{(k+1)})\big)^T(x^{(k+1)} - x^*) = \big(f^{(1)}(x^{(k+1)}) - f_{k4}^{(1)}(x^{(k+1)})\big)^T v^{(k+1)},$$

where $v^{(k+1)} = x^{(k+1)} - x^*$. Thus,

$$\big(f^{(1)}(x^{(k+1)})\big)^T(x^{(k+1)} - x^*) \le \Big\|f^{(1)}(x^{(k+1)}) - f_{k4}^{(1)}(x^{(k+1)})\Big\|\big\|v^{(k+1)}\big\|.$$

However by using (38), we get

$$f^{(1)}(x^{(k+1)}) - f_{k4}^{(1)}(x^{(k+1)}) = O\big(\big\|u_2^{(k)}\big\|^4\big).$$

Then there is a constant $M > 0$ such that

$$\Big\|f^{(1)}(x^{(k+1)}) - f_{k4}^{(1)}(x^{(k+1)})\Big\| \le M\Big\|u_2^{(k)}\Big\|^4,$$

and, therefore,

$$\big(f^{(1)}(x^{(k+1)})\big)^T(x^{(k+1)}-x^*)\le M\Big\|u_2^{(k)}\Big\|^4\|v^{(k+1)}\|.$$

Using the inequality (36) yields

$$m_1\|Pv^{(k+1)}\|^4\le\big(f^{(1)}(x^{(k+1)})\big)^T(x^{(k+1)}-x^*)\le M\Big\|u_2^{(k)}\Big\|^4\|v^{(k+1)}\|$$

or

$$\frac{\|Pv^{(k+1)}\|^4}{\|v^{(k+1)}\|}=\big\|v^{(k+1)}\big\|^3\le\frac{M}{m_1}\Big\|u_2^{(k)}\Big\|^4. \tag{40}$$

The vector $u_2^{(k)}$ is determined from the system of equations (39) multiplying both sides of this system by the vector u_2. This results in the following equation for u_2:

$$f^{(4)}(x^{(k)})[(u_2)^4]=-f^{(3)}(x^{(k)})\big[(u_2)^3\big].$$

Then, from condition (35) with $p=4$ and the continuity of the 4th derivative of the function $f(x)$ on a sufficiently small neighborhood of the minimum point, it follows that

$$\frac{m_4}{2}\|u_2\|^4=\frac{m_4}{2}\|Pu_2\|^4\le f^{(4)}(x^{(k)})[(u_2)^4]\le|f^{(3)}(x^{(k)})\big[(u_2)^3\big]|\le\big\|f^{(3)}(x^{(k)})\big\|\|u_2\|^3.$$

From which the following estimate is derived

$$\big\|u_2^{(k)}\big\|\le 2/m_4\big\|f^{(3)}(x^{(k)})\big\|. \tag{41}$$

Finally, expanding in the Taylor series, and using the condition (25) with $p=4$ (which is a necessary condition for a minimum [24], from the continuity of the 4th derivative of $f(x)$ in a neighborhood of the minimum point it is deduced that there exists a constant $M_4>0$ such that

$$\big\|f^{(3)}(x^{(k)})\big\|=\big\|f^{(3)}(x^*)+f^{(4)}(x^*)v^{(k)}+O\big(\big\|v^{(k)}\big\|^2\big)\big\|\le M_4\big\|v^{(k)}\big\|. \tag{42}$$

Then (40) – (42) give

$$\|v^{(k+1)}\|^3\le\frac{M}{m_1}\Big\|u_2^{(k)}\Big\|^4\le\frac{M}{m_1}\big(1/m_4\big\|f^{(3)}(x^{(k)})\big\|\big)^4\le\frac{16M(M_4)^4}{m_1(m_4)^4}\big\|v^{(k)}\big\|^4,$$

from which follows

$$\|v^{(k+1)}\|\le\left(\frac{16M(M_4)^4}{m_1(m_4)^4}\right)^{1/3}\big\|v^{(k)}\big\|^{4/3}.$$

The theorem is proved.

Now it is possible to analyze the convergence rate of the combined second-order methods given by (2), (8) and (9). For ease of analysis, consider the approach used above.

The combined second-order method described by (2), (8) and (9) is a combination of the Newton's method and the gradient method. At each kth iteration, the entire space $\mathbb{R}^n$ is represented as the direct sum of two subspaces $\mathrm{Ker}\,(H_{k\varepsilon})$ and the orthogonal complement of it. The gradient method is

applied to the subspace $\mathrm{Ker}\,(H_{k\varepsilon})$, and Newton's method is applied to the orthogonal complement of it. The convergence rate of the combined method is determined by the convergence rate of the slowest of the original two methods. Now let us analyze the method defined by (2), (8) and (9) in the case of the matrix $f^{(2)}(x^*)$ that is equal to the zero matrix. In this case, the method becomes the gradient method, wherein it is considered $\alpha_{k2} = \rho > 0$ (the gradient method with a fixed step). Therefore, the method given by (2), (8) and (9) takes the form

$$x^{(k+1)} = x^{(k)} + \rho u_2^{(k)}, \quad k = 0, 1, 2, \ldots, \tag{43}$$

where $u_2^{(k)} = -\dfrac{1}{\varepsilon} g^{(k)}$, $x^{(k+1)} - x^{(k)} = \rho u_2^{(k)}$.

Theorem 2.4. *Suppose that function $f(x)$ has a minimum at the point $x^* \in \mathbb{R}^n$. Let $f(x)$ be* 4 *times continuously differentiable in the neighborhood $V(x^*)$ of x^*, the matrix $f^{(2)}(x^*)$ is equal to the zero matrix, and, for all $u \in \mathbb{R}^n$, the condition* (35) *is satisfied. Then the gradient method with a fixed step given by* (43) *converges in a neighborhood of x^* and the rate of convergence is sublinear* [2].

Proof. Formula (43) gives $v^{(k+1)} = v^{(k)} - \dfrac{\rho}{\varepsilon} g^{(k)}$, where $v^{(k+1)} = x^{(k+1)} - x^*$. Then

$$\|v^{(k+1)}\|^2 = \|v^{(k)} - \frac{\rho}{\varepsilon} g^{(k)}\|^2 = \|v^{(k)}\|^2 - 2\frac{\rho}{\varepsilon}(g^{(k)})^T v^{(k)} + \left(\frac{\rho}{\varepsilon}\right)^2 \|g^{(k)}\|^2. \tag{44}$$

Using the inequality (36) with $p = 4$ yields

$$(g^{(k)})^T v^{(k)} \geq m_1 \|v^{(k)}\|^4. \tag{45}$$

By the Taylor expansion, using the continuity of the 4th derivative of $f(x)$ in a neighborhood of the minimum point, it is derived that there exists a constant $M_4 > 0$ such that

$$\|f^{(1)}(x^{(k)})\| = \left\| f^{(1)}(x^*) + \quad f^{(2)}(x^*)v^{(k)} + \frac{1}{2} f^{(3)}(x^*)(v^{(k)})^2 + \frac{1}{6} f^{(4)}(x^*)(v^{(k)})^3 + O\big(\big\|v^{(k)}\big\|^4\big) \right\|$$

$$\leq M_4 \big\|v^{(k)}\big\|^3. \tag{46}$$

Therefore, from (44) – (46) it is straightforward that

$$\|v^{(k+1)}\|^2 \leq \|v^{(k)}\|^2 - 2\frac{\rho}{\varepsilon} m_1 \|v^{(k)}\|^4 + \left(\frac{\rho}{\varepsilon}\right)^2 (M_4)^2 \|v^{(k)}\|^6$$

$$\leq \left(1 - 2\frac{\rho}{\varepsilon} m_1 \|v^{(k)}\|^2 + \left(\frac{\rho}{\varepsilon}\right)^2 (M_4)^2 \|v^{(k)}\|^4\right) \|v^{(k)}\|^2.$$

Then, for a sufficiently small neighborhood of the minimum point, the following estimate is obtained:

$$\|v^{(k+1)}\|^2 \leq \left(1 - \frac{\rho}{\varepsilon} m_1 \|v^{(k)}\|^2\right) \|v^{(k)}\|^2$$

or

$$\|v^{(k+1)}\| \leq \left(1 - \frac{\rho}{\varepsilon} m_1 \|v^{(k)}\|^2\right)^{\frac{1}{2}} \|v^{(k)}\|.$$

Thus, the convergence rate of the method defined by (43) is sublinear

3. A Quasi-Newton version of the methods. In practice, calculation of the Hessian matrix at each iteration can be overly time consuming. Consequently, it seems logic al to consider the quasi-Newton versions of both the combined second-order method given by (2), (8), (9) and the combined fourth-order method given by (21), (13), (14). To this end, the matrix H_k, used in (3), is updated according to the Broyden – Fletcher – Goldfarb – Shanno formula (BFGS) [2]. Numerical experiments show that as $x^{(k)}$ approaches a degenerate minimum point, which, in the case of these combined methods, means that the parameter value $r_k < n$, it is preferable to periodically recalculate the matrix H_k as $H_k = f^{(2)}(x^*)$. Otherwise, the number of iterations essentially increases. Apparently, this happens because the representation of the entire space as the direct sum of two subspaces is inaccurate due to the large error in computing of the Hessian matrix approximation.

4. Adaptive combined quasi-Newton-type method (ACQNM). Let us consider the adaptive combined method (ACQNM), which makes use of the above methods and is a quasi-Newton method. When the rank of the Hessian degeneracy is equal to one, ACQNM coincides with the fourth-order method defined by (21), (13) and (14). If the rank of the Hessian degeneracy is more than one, then ACQNM coincides with the second-order method defined by (2), (8) and (9). In ACQNM, the matrix H_k used in (3), is recalculated according to the BFGS formula [3] at each iteration. The step sizes in the formula (2) are determined by the algorithms described above.

The efficiency of ACQNM is supported by the numerical experiments that were carried out on generally accepted test functions for unconstrained optimization problems [1]. ACQNM was tested using R, Scilab, and Python. The results of the numerical experiments are presented below.

Algorithm 4. Adaptive combined quasi-Newton-type method (ACQNM)

Input: Initial point $x^{(0)}$; regularization parameter $\varepsilon > 0$; algorithm one-dimensional minimization OneDimMin (which is given in Subsection 2.3); algorithm of spectral decomposition of a matrix eigen (which is a standard function in R); step of numerical differentiation h; maximum number of iterations K; accuracy by the gradient $\mathrm{tool\_grad} > 0$; accuracy by the argument $\mathrm{tool\_arg} > 0$ accuracy by the function $\mathrm{tool\_fun} > 0$.

initialization: $k = 0$; $f_0 = f(x^{(0)})$; $g^{(0)} = f^{(1)}(x^{(0)})$; $H_0 = I$;
while $\|g^{(k)}\| > \mathrm{tool\_grad}$:
    $(\Lambda_k, Q_k) =$ eigen (H_k, symmetric = TRUE), $\Lambda_k = \mathrm{diag}\,(\lambda_1^{(k)}, \ldots, \lambda_n^{(k)}), \lambda_i^{(k)} \geq \lambda_{i+1}^{(k)}$;
initialization: $r_k = 0$;
for $i = 0, \ldots, n$:
    if $|\lambda_i^{(k)}| > \varepsilon$ **then** $r_k = r_k + 1$;
    if $\lambda_i^{(k)} < -\varepsilon$ **then** $\lambda_i^{(k)} = -\lambda_i^{(k)}$;
if $r_k = n$ **then**
    $u^{(k)} = -Q_k \Lambda_k^{-1} Q_k^T g^{(k)}, \;\; u_2^{(k)} = 0.$
    a_k is determined as a minimum point of the function of one variable $\varphi_1(\alpha) = f(x^{(k)} + \alpha u^{(k)})$ with the initial approximation $\alpha_0 = 1$
    $[a_k, x^{(k+1)}, f_{k+1}] = \mathrm{OneDimMin}(x^{(k)}, f_k, u^{(k)}, 1)$;
else
    $Q_k = \begin{bmatrix} Q_{k1} & Q_{k2} \end{bmatrix}, \Lambda_k = \begin{bmatrix} \Lambda_{k1} & 0 \\ 0 & \Lambda_{k2} \end{bmatrix}$
    if $r_k < n - 1$ **then**
        $u_1^{(k)} = -Q_{k1} \Lambda_{k1}^{-1} Q_{k1}^T g^{(k)}$;

$u_2^{(k)} = -\varepsilon^{-1} Q_{k2} Q_{k2}^T g^{(k)}$.

a_{k1} is determined as a minimum point of the function of one variable $\varphi_1(\alpha) = f(x^{(k)} + \alpha u_1^{(k)})$ with the initial approximation $\alpha_0 = 1$

$[\alpha_{k1}, \hat{x}^{(k+1)}, \hat{f}_{k+1}] = \text{OneDimMin}(x^{(k)}, f_k, u_1^{(k)}, 1)$, where $\hat{x}^{(k+1)} = x^{(k)} + \alpha_{k1} u_1^{(k)}$, $\hat{f}_{k+1} = f(\hat{x}^{(k+1)})$;

a_{k2} is determined as a minimum point of the function of one variable $\varphi_2(\alpha) = f(x^{(k)} + \alpha_{k1} u_1^{(k)} + \alpha u_2^{(k)})$ with the initial approximation $\alpha_0 = \max(1, a_{(k-1),2})$;

$[\alpha_{k2}, x^{(k+1)}, f_{k+1}] = \text{OneDimMin}(\hat{x}^{(k+1)}, \hat{f}_{k+1}, u_2^{(k)}, \max(1, a_{(k-1),2}))$;

else

calculate $y^{(k)} \approx \dfrac{f^{(1)}(x^{(k)} + hq_k^{(n)}) - 2f^{(1)}(x^{(k)}) + f^{(1)}(x^{(k)} - hq_k^{(n)})}{h^2}$;

μ is determined as a solution of the equation (16);

$u_2^{(k)} = \mu s q_k^{(n)}$, where $s = \text{sign}\,((g^{(k)})^T q_k^{(n)})$;

$u_1^{(k)} = -Q_{k1} \Lambda_{k1}^{-1} Q_{k1}^T (g^{(k)} + \frac{\mu^2}{2} y^{(k)})$;

$u^{(k)} = u_1^{(k)} + u_2^{(k)}$;

a_k is determined as a minimum point of the function of one variable $\varphi_3(\alpha) = f(x^{(k)} + \alpha u^{(k)})$ with the initial approximation $\alpha_0 = 1$;

$[\alpha_k, x^{(k+1)}, f_{k+1}] = \text{OneDimMin}(x^{(k)}, f_k, u^{(k)}, 1)$;

if $\|x^{(k+1)} - x^{(k)}\|/(1 + \|x^{(k+1)}\|) \le \text{tool\_arg}$ **then break** with code=1;

if $|f_{k+1} - f_k|/(1 + |f_{k+1}|) \le \text{tool\_fun}$ **then break** with code $= 2$;

$x^{(k)} = x^{(k+1)}$: $f_k = f_{k+1}$; $k = k + 1$; $g^{(k)} = f^{(1)}(x^{(k)})$;

if $\|g^{(k)}\| \le \text{tool\_grad}$ **then break** with code=0;

if $k > K$ **then break** with code $= 3$;

H_k is updated according to the BFGS formula;

Output: last $x^{(k)}, f_k, g^{(k)}, k$.

5. Results of the numerical experiments. ACQNM for solving the degenerate problem (1) was implemented in R. To guarantee the stability of the method, the negative diagonal elements λ_i^k of the matrix Λ_k given by (3) were replaced by $-\lambda_i^k$. The matrix H_k, used in (3), is updated according to the BFGS formula with the initial approximation $H_0 = I$. The regularization parameter of the numerical method ε was taken equal to $10^{-7} \times \max_{i=1,\ldots,n}(|\lambda_i^k|)$. Note that the matrix H_k is thought to be close to degenerate if its condition number $\text{cond}(H_k)$ is greater than 10^7. The derivatives $g^k = f'(x^k)$ were calculated numerically by the symmetric formula

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x|x_i + h_i) - f(x|x_i - h_i)}{2h_i} \tag{47}$$

with a step of numerical differentiation $h_i = h_0 \max(1, |x_i|)$, where $h_0 = 10^{-6}$.

ACQNM was tested on the following test functions for unconstrained optimization problems [1].

1. **Extended Rosenbrock function**: $f(x) = \sum_{i=1}^{n/2} \left[100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2\right]$, where the initial estimate is $x^0 = (-1.2, 1, \ldots, -1.2, 1)^T$, the minimum point is $x^* = (1, 1, \ldots, 1)^T$, the value of the objective function at the minimum point is $f(x^*) = 0$, $\text{rank}\,(f''(x^*) = n$, and $\text{cond}\,(f''(x^*)) = 2508.01$.

2. **Extended White & Holst function**: $f(x) = \sum_{i=1}^{n/2}\Big[100\big(x_{2i} - x_{2i-1}^3\big)^2 + (1 - x_{2i-1})^2\Big]$, the initial estimate is $x^0 = (-1.2, 1, \ldots, -1.2, 1)^T$, the minimum point is $x^* = (1, 1, \ldots, 1)^T$, the value of the objective function at the minimum point is $f(x^*) = 0$), $\operatorname{rank}(f''(x^*)) = n$, $\operatorname{cond}(f''(x^*)) = 10018.01$.

3. **Extended Wood function**: $f(x) = \sum_{i=1}^{n/4}[100(x_{4i-3}^2 - x_{4i-2})^2 + (x_{4i-3} - 1)^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1 - x_{4i-1})^2 + 10.1((x_{4i-2} - 1)^2 + (x_{4i} - 1)^2) + 19.8(x_{4i-2} - 1)(x_{4i} - 1)]$, where the initial estimate is $x^0 = (-3, -1, -3, -1, \ldots, -3, -1, -3, -1)^T$, the minimum point is $x^* = (1, 1, 1, 1, \ldots, 1, 1, 1, 1)^T$, the value of the objective function at the minimum point is $f(x^*) = 0$, $\operatorname{rank}(f''(x^*)) = n$, $\operatorname{cond}(f''(x^*)) = 1397.957$.

4. **Extended Powell function**: $f(x) = \sum_{i=1}^{n/4}[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$, where the initial estimate is $x^0 = (3, -1, 0, 1, \ldots, 3, -1, 0, 1)^T$, the minimum point is $x^* = (0, 0, 0, \ldots, 0, 0, 0)^T$, the value of the objective function at the minimum point is $f(x^*) = 0$, $\operatorname{rank}(f''(x^*)) = \dfrac{n}{2}$.

5. **Extended Freudenstein & Roth function**: $f(x) = \sum_{i=1}^{n/2}(-13 + x_{2i-1} + ((5 - x_{2i})x_{2i} - 2)x_{2i})^2 + (-29 + x_{2i-1} + ((x_{2i} + 1)x_{2i} - 14)x_{2i})^2$, the initial estimate of the minimum is $x^0 = (0.5, -2, \ldots, 0.5, -2)^T$, the minimum point of the objective function is $x^* = (11.4127790, -0.8968053, \ldots, 11.4127790, -0.8968053)^T$, the value of the objective function at the minimum point is $f(x^*) = \begin{cases} 97.96851 & \text{if } n = 4, \\ 293.9055 & \text{if } n = 12, \end{cases}$ $\operatorname{rank}(f''(x^*)) = n$, $\operatorname{cond}(f''(x^*)) = 1102.78$.

6. **Extended Tridiagonal 1 function**: $f(x) = \sum_{i=1}^{n/2}[(x_{2i-1} + x_{2i} - 3)^2 + (x_{2i-1} - x_{2i} + 1)^4]$, the initial estimate is $x^0 = (2, 2, \ldots, 2)^T$, the minimum point is $x^* = (1, 2, \ldots, 1, 2)^T$, the value of the objective function at the minimum point is $f(x^*) = 0$, $\operatorname{rank}(f''(x^*)) = \dfrac{n}{2}$.

7. **FLETCHCR function (CUTE)**: $f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i + 1 - x_i^2)^2$, the initial estimate is $x^0 = (0, 0, \ldots, 0)$, and the minimum point is not strict (therefore, it is not unique). The value of the objective function at the minimum point is $f(x^*) = 0$, $\operatorname{rank} f''(x^*) = n - 1$.

8. **My function1**: $f(x) = 1000(x_1 - 1000)^2 + 0.001x_2^4 + \sum_{i=3}^{n}(x_i - i)^2$, the initial estimate is $x^0 = (100, \ldots, 100)$, the minimum point is $x^* = (1000, 0, 3, 4, \ldots, n)$, the value of the objective function at the minimum point is $f(x^*) = 0$, $\operatorname{rank}(f''(x^*)) = n - 1$.

9. **My function2**: $f(x) = x_1^2 + x_1x_2^2 + x_2^4 + \sum_{i=3}^{n} x_i^2$, the initial estimate is $x^0 = (10, 14, 10, \ldots, 10)$, the minimum point is $x^* = (0, 0, \ldots, 0, 0)$, the value of the objective function at the minimum point is $f(x^*) = 0$, $\operatorname{rank}(f''(x^*)) = n - 1$.

10. **Mean-square approximation by polynomials**: $f(x) = \sum_{j=1}^{101}\Big[\sum_{i=1}^{n} x_i \cdot 0.01(j-1)^{i-1} - \sum_{i=1}^{n} x_i^* \cdot 0.01(j-1)^{i-1}\Big]^2$, where $n = 5$. The initial estimate is $x^0 = (2, 2, 2, 2, 2)$, the minimum point is $x^* = (1, 1, 1, 1, 1)$, and the value of the objective function at the minimum point is $f(x^*) = 0$, $\operatorname{rank} f''(x^*) = n - 1$.

Note that the functions 4, 6 – 10 are characterized by the Hessian matrices $f''(x^*)$ that are poorly conditioned. The functions 7 – 10 have the Hessian matrices with a degeneracy rank that is equal to 1. The numerical experiments for the functions 1 – 9 were carried out for $n = 4$ and $n = 100$.

It is interesting to compare ACQNM with the following standard computational tools: the **optim** function (method L-BFGS-B) of the mathematical package in R 4.3.2, the **minimize** (method BFGS) function of the mathematical package in Python (scipy.optimize version: 1.11.3), and the **optim** function (method 'Quasi-Newton with BFGS' of the numerical computational package Scilab 6.1.1. The calculations were performed with maximum precision. The results of the numerical experiments are presented in Tables 1 – 4, where the following variables are defined:

Funct indicates which function is under test;

n is a value of the parameter n;

n-r is the rank of Hessian degeneracy at the minimum point;

Dx $= \|\widetilde{x} - x^*\|$ is the Euclidean norm of the difference $\widetilde{x} - x^*$, where $\widetilde{x}$ is the approximation of the solution obtained by the optimization procedure;

Df $= |f(\tilde{x}) - f(x^*)|$;

Nitr is the number of iterations performed;

Nf is the number of calculations of the objective function performed;

Ngr is the number of calculations of the gradient of the objective function performed;

NormGr $= \|f'(\tilde{x})\|$;

code, message, or exitflag are exit codes returned by the optimization procedure;

For **ACQNM**, the parameter Nf does not account for the number of the evaluations of the objective function needed for numerical computation of the objective function gradient by the formula (47).

For **ACQNM**, the exit code takes the following values:

0 – required accuracy by the gradient is reached (10^{-20} was given);

1 – required accuracy by the argument is reached (10^{-10} was given);

2 – required accuracy by the function is reached (10^{-25} was given).

For the **optim** function (package R), the exit code message takes the following values:

1 – ERROR: ABNORMAL_TERMINATION_IN_LNSRCH;

2 – CONVERGENCE: REL_REDUCTION_OF_F $\leq$ FACTR*EPSMCH;

3 – NEW_X.

For the **minimize** function (package Python), the exit code takes the following values:

1 – Desired error not necessarily achieved due to precision loss;

2 – Optimization terminated successfully.

For the **optim** function (package Scilab), the exit code `err` takes the following values:

1 – Norm of projected gradient is lower than ...;

4 – Optim stops: maximum number of calls to f is reached;

5 – Optim stops: maximum number of iterations is reached;

9 – End of optimization, successful completion.

Tables 1 and 3 show the calculation results for the **optim** function of the mathematical package in R and the **minimize** function of the mathematical package in Python for $n = 4$ and $n = 100$. Tables 2 and 4 show the calculation results for the **optim** function (Quasi-Newton with BFGS' method) of the mathematical package Scilab and the **ACQNM** function for $n = 4$ and $n = 100$.

As Tables 1 – 4 show, the **optim (L-BFGS-B)** function in **R** and the **minimize (BFGS)** function of the **Python** scipy.optimize package both perform worse than the **optim** function of the Scilab math package in terms of accuracy and number of calculations of the objective function. At the same time, the **optim** procedure of the **Scilab** math package shows worse results than **ACQNM** in terms of accuracy and number of calculations of the objective function.

Table 1. Calculation results for the **optim** function in R and the **minimize** function in Python for $n = 4$

Funct			R optim (L-BFGS-B)							Python minimize (BFGS)						
	n	n-r	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
1	4	0	6.3e-04	8.0e-08		91	91	5.6e-04	1	1.4e-05	4.0e-11	58	542	106	4.1e-07	1
2	4	0	3.8e-03	1.5e-06		109	109	7.9e-04	1	4.0e-05	1.6e-10	79	697	137	2.8e-07	1
3	4	0	8.0e-06	4.4e-10		154	154	3.9e-04	2	4.3e-07	5.7e-13	89	652	128	3.2e-10	1
4	4	2	5.1e-05	6.7e-18		135	135	1.2e-10	1	1.2e-03	5.2e-11	48	457	89	8.9e-07	1
5	4	0	1.8e-05	1.7e-10		91	91	2.6e-04	2	3.9e-07	2.8e-14	21	190	38	9.7e-06	1
6	4	2	1.0e-05	3.9e-19		51	51	1.7e-09	2	4.8e-07	1.1e-16	54	670	132	1.2e-15	1
7	4	1		2.0e-10		51	51	5.2e-04	1		3.9e-13	12	292	56	1.8e-05	1
8	4	1	4.6e-05	1.2e-20		84	84	3.4e-09	2	1.2e-06	5.5e-14	69	390	78	6.8e-216	2
9	4	1	2.1e-05	1.5E-19		130	130	2.2e-12	2	7.0e-05	1.8e-16	50	442	86	1.2e-11	1
10	5	1	2.2e-12	2.0e-25		88	88	2.9e-12	2	3.0e-07	9.1e-15	13	2270	43	2.4e-11	1

Table 2. Calculation results for the **optim** function of the mathematical package Scilab and the **ACQNM function** for $n = 4$

Funct			Scilab optim (quasi-Newton with BFGS)							ACQNM						
	n	n-r	Dx	Df	Nitr	Nf	Ngr	NormGr	Err	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
1	4	0	2.3e-08	1.1e-16	50	89		4.0e-14	9	2.1e-13	9.9e-27	22	115	23	1.1e-12	1
2	4	0	1.0e-07	2.1e-15	49	172		4.8e-14	9	6.0e-13	6.4e-26	33	157	34	1.1e-11	1
3	4	0	7.0e-10	1.6e-18	81	180		3.9e-14	9	5.3e-15	4.1e-29	51	212	52	2.4e-13	1
4	4	2	1.2e-10	3.2e-40	93	175		2.1e-29	1	1.2e-06	2.7e-24	39	262	48	5.0e-16	1
5	4	0	4.2e-10	7.1e-14	27	158		3.1e-10	9	3.2e-08	5.7e-14	9	52	10	4.7e-07	2
6	4	2	4.0e-07	3.1e-26	62	377		3.0e-16	9	2.8e-06	1.2e-22	19	84	26	4.7e-12	1
7	4	1		7.1e-20	25	206		3.6e-09	9		5.5e-30	10	51	11	1.5e-13	1
8	4	1	9.1e-08	4.3e-32	90	179		4.4e-16	9	9.1e-07	1.4e-26	21	125	48	2.4e-13	2
9	4	1	1.1e-21	1.3e-84	213	229		3.2e-43	1	2.1e-07	7.7e-27	37	160	58	1.6e-13	2
10	5	1	1.5e-13	2.5e-28	14	42		9.0e-15	9	7.3e-11	2.5e-24	7	55	8	1.7e-11	1

Table 3. Calculation results for the **optim** function in R and the **minimize** function in Python for $n = 100$

Funct			R optim (L-BFGS-B)							Python minimize (BFGS)						
	n	n-r	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
1	100	0	0.002	8.4e-07		114	114	0.003	1	7.4e-05	1.1e-09	473	56066	555	1.1e-05	1
2	100	0	0.02	3.6e-05		122	122	0.01	1	2.1e-04	4.8e-09	712	87882	870	9.3e-05	1
3	100	0	7.2e-04	2.1e-07		257	257	0.002	1	2.6e-06	1.5e-11	789	89294	884	2.2e-06	1
4	100	50	3.5e-04	5.4e-16		705	705	1.3e-08	2	2.1e-02	38e-08	335	40816	404	1.5e-05	1
5	100	0	9.2e-05	4.3e-09		43	43	1.2e-03	2	7.3e-03	2.2e-09	181	27068	268	2.3e-04	1
6	100	50	2.2e-10	8.4e-29		47	47	2.7e-14	2	5.3e-05	2.8e-15	82	12634	125	2.9e-13	1
7	100	1		4.3e-08		1400	1400	2.8e-03	1		89.09	419	48592	481	1.4e-05	1
8	100	1	2.7e-09	7.3e-27		101	101	1.7e-13	2	1.7e-04	6.1e-14	53	12029	119	2.6e-10	1
9	100	1	3.3e-10	9.2e-27		66	66	1.9e-13	2	7.0e-05	5.5e-15	52	8395	83	2.3e-13	1

Tables 2 and 4 show the results of the calculations for the **optim** function of the Scilab mathematical package and **ACQNM** for $n = 4$ and $n = 100$. These results confirmed that **ACQNM** allows one to obtain more accurate solution to the problem. In this case, **ACQNM** uses approximately 1.5 – 2 times fewer calculations of the objective function for $n = 4$ (Table 2) as well as for $n = 100$ (Table 4). But the main point is that for problems 7 – 9 (for $n = 4$ and $n = 100$) as well as problem 10 (for $n = 5$), which have one-dimensional kernel of the Hessian matrix, **ACQNM** performs significantly fewer iterations, confirming its faster convergence rate.

Table 4. Calculation results for the optim function of the mathematical package Scilab and the ACQNM function for $n = 100$

Funct			Scilab optim (quasi-Newton with BFGS)							ACQNM						
	n	**n-r**	**Dx**	**Df**	**Nitr**	**Nf**	**Ngr**	**NormGr**	**Code**	**Dx**	**Df**	**Nitr**	**Nf**	**Ngr**	**NormGr**	**Code**
1	100	0	1.0e-07	2.7e-15	231	834		3.8e-12	9	4.1e-09	7.2e-18	72	492	73	8.8e-08	1
2	100	0	7.0e-07	5.4e-14	85	368		9.4e-13	9	1.2e-08	1.5e-17	228	992	229	2.9e-08	1
3	100	0	3.5e-09	4.0e-17	93	177		1.6e-12	9	1.1e-08	4.5e-17	454	1630	455	1.2e-08	1
4	100	50	3.4e-07	1.3e-23	501	679		2.2e-17	5	5.1e-05	7.0e-19	255	2153	300	3.1e-13	1
5	100	0	1.5e-08	5.7e-13	28	200		1.0e-07	9	2.4e-06	1.1e-11	16	106	17	1.2e-04	1
6	100	50	1.9e-06	1.2e-24	52	210		2.5e-13	9	8.5e-06	4.4e-22	29	114	46	7.4e-12	1
7	100	1		101.3	499	938		2.0e-07	9		50.8	199	675	200	6.4e-06	2
8	100	1	3.9e-07	1.5e-26	159	420		2.5e-13	9	3.3e-06	2.7e-25	30	148	67	7.9e-13	1
9	100	1	1.5e-21	6.9e-84	257	302		3.6e-42	1	2.2e-07	2.2e-27	43	174	70	4.3e-14	2

6. Conclusion. The second-order and the fourth-order numerical methods for solving degenerate unconstrained optimization problems are considered in the paper using a new approach. Namely, the entire space is represented as a sum of two orthogonal subspaces. This representation is based on the spectral decomposition of the Hessian matrix. This approach is particularly interesting because it allows using different methods on different subspaces. Moreover, any method can be applied to the kernel of the Hessian matrix.

The convergence rate of each of these methods is analyzed using generalized necessary and sufficient conditions for a minimum presented in [24]. However, for the sake of simplicity, the approach that gives an estimation for the convergence rate in the worst possible case is used, and thus, in practice, the convergence may be faster.

The proposed new adaptive combined quasi-Newton-type method (ACQNM), in fact, combines three methods and makes use of each method if the situation requires it. This method, in many cases, makes solving practical degenerate optimization problems more effective.

In view of the fact that the calculation of the spectral decomposition (3) of the matrix H_k at each iteration of ACQNM can be very costly, one may use the decomposition described in [26]. The decomposition from [26] of the symmetric matrices is less numerically stable then the decomposition (3). Note that the decomposition from [26] may be used at the iterations of the method while the matrix H_k is not close to a singular matrix.

Conflict of interest. The authors declare that they have no potential conflict of interest in relation to the study in this paper.

Funding. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Author contributions. All authors have contributed equally to the work.

References

1. N. Andrei, *A collection of 75 unconstrained optimization test functions*, Research Institute for Informatics, Technical Report, **6**, 1 – 9 (2018).
2. N. Andrei, *Modern numerical nonlinear optimization*, Springer, Cham (2022).
3. K. N. Belash, A. A. Tret'yakov, *Methods for solving degenerate problems*, Comput. Math. and Math. Phys., **28**, № 4, 90 – 94 (1988).
4. K. Ghazali, J. Sulaiman, Y. Dasril, D. Gabda, *Newton-SOR Iteration for Solving Large-Scale Unconstrained Optimization Problems with an Arrowhead Hessian Matrices*, J. Phys.: Conf. Ser., **1358**, № 1, 1 – 10 (2019).

5. G. Wang, Y. Wei, S. Qiao, *Generalized inverses: theory and computations*, Springer Nature, Singapore (2018).
6. I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, MIT Press (2016).
7. B. A. Hassan, M. A. Al Kahya, *A new class of quasi-Newton updating formulas for unconstrained optimization*, J. Interdiscip. Math., **24**, № 8, 2355 – 2366 (2021).
8. X. Han, J. Zhang, J. Chen, *New Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization*, Bull. Iran. Math. Soc., **43**, № 6, 2067 – 2084 (2017).
9. J.-P. Penot, *Higher-order optimality conditions and higher-order tangents sets*, SIAM J. Optim., **27**, № 4, 2508 – 2527 (2017).
10. B. Jimenez, V. Novo, *Higher-order optimality conditions for strict local minima*, Ann. Oper. Res., **157**, 183 – 192 (2008).
11. L. Li, M. Qin, H. Wang, *Regularized Newton Method with Correction for Unconstrained Convex Optimization*, Open J. Optim., **68**, № 1, 44 – 52 (2016).
12. N. Andrei, *Diagonal approximation of the Hessian by finite differences for unconstrained optimization*, J. Optim. Theory and Appl., **185**, № 3, 859 – 879 (2020).
13. X. Li, B. Wang, W. Hu, *A modified nonmonotone BFGS algorithm for unconstrained optimization*, J. Inequal. and Appl., **183**, 1 – 18 (2017).
14. N. G. Maratos, M. A. Moraitis, *Some results on the Sign recurrent neural network for unconstrained minimization*, Neurocomputing, **287**, 1 – 25 (2018).
15. D. Mehta, T. Chen, T. Tang, J. D. Hauenstein, *The loss surface of deep linear networks viewed through the algebraic geometry lens*; arXiv preprint arXiv:1810.07716 (2018).
16. T. D. Niri, M. M. Hosseini, M. Heydari, *An efficient improvement of the Newton method for solving nonconvex optimization problems*, Comput. Methods Different. Equat., **7**, № 1, 69-85 (2019).
17. W. Quapp, *Searching minima of an N-dimensional surface: a robust valley following method*, Comput. and Math. Appl., **41**, 407 – 414 (2001).
18. G. Ma, H. Lin, W. Jin, D. Han, *Two modified conjugate gradient methods for unconstrained optimization with applications in image restoration problems*, J. Appl. Math. and Comput., **68**, № 6, 4733 – 4758 (2022).
19. A. R. Sankar, V. N. Balasubramanian, *Are saddles good enough for deep learning?*; arXiv preprint arXiv:1706.02052 (2017).
20. C. Shen, X. Chen, Y. Liang, *A regularized Newton method for degenerate unconstrained optimization problems*, Optim. Lett., **6**, 1913 – 1933 (2012).
21. J. Lu, *Matrix decomposition and applications*, Amazon Digital Services LLC (2022).
22. E. Szczepanik, A. Prusinska, A. Tret'yakov, *The p-factor method for nonlinear optimization*, Schedae Inform., **21**, 141 – 157 (2012).
23. D. di Serafino, G. Toraldo, M. Viola, *Using gradient directions to get global convergence of Newton-type methods*, Appl. Math. and Comput., **409**, Article № 125612 (2021).
24. V. M. Zadachyn, *Higher-order optimality conditions for degenerate unconstrained optimization problems*, J. Optim., Different. Equat. and Appl., **30**, № 1, 88 – 97(2022); DOI: 10.15421/142204.
25. V. M. Zadachyn, *Modified Newton and quasi-Newtonian-type methods with pseudo-inversions for solving degenerate problems*, Ph. D. Thesis, Lomonosov Moscow State University, Moscow, CA (1988) (in Russian); https://search.rsl.ru/ru/record/01000049990.
26. V. I. Meleshko, V. M. Zadachin, *Factorizations and pseudo-inversions of singular perturbed matrices with nonfixed signs*, Izv. Vyss. Uchebn. Zaved. Mat., **11**, 42 – 50 (1987).
27. E. G. Birgin, J. M. Martinez, *The use of quadratic regularization with a cubic descent condition for unconstrained optimization*, SIAM J. Optim., **27**, № 2, 1049 – 1074 (2017).
28. E. G. Birgin, J. M. Martinez, *Newton-like method with mixed factorizations and cubic regularization for unconstrained minimization*, Comput. Optim. and Appl., **73**, 707 – 753 (2019).
29. S. Javed, A. Khan, *Efficient regularized Newton-type algorithm for solving convex optimization problem*, J. Appl. Math. and Comput., **68**, № 4, 2343 – 2363 (2022).
30. H. Zhang, Q. Ni, *A new regularized quasi-Newton method for unconstrained optimization*, Optim. Lett., **12**, № 7, 1639 – 1658 (2018).

Received 29.11.22