CYBERNETICS and COMPUTER TECHNOLOGIES

UDC 519.853.6 : 519.613.2 DOI:10.34229/2707-451X.24.3.2

V.M. ZADACHYN

A COMBINED QUASI-NEWTON-TYPE METHOD USING 4TH-ORDER DIRECTIONAL DERIVATIVES FOR SOLVING DEGENERATE PROBLEMS OF UNCONSTRAINED OPTIMIZATION

A combined quasi-Newton-type method is presented for solving degenerate unconstrained optimization problems, based on an orthogonal decomposition of the Hessian approximation matrix and division of the entire space into two orthogonal subspaces. On one subspace (the kernel of the Hessian approximation matrix), a method is applied where derivatives in the direction of the 4th order are computed, while on the orthogonal complement to it, a quasi-Newtonian method is applied.

Keywords: *unconstrained optimization, quasi-Newton methods, degenerate minimum point, spectral matrix decomposition, Machine Learning.*

Introduction. Unconstrained optimization methods are of great importance in machine learning [1–5]. Currently, gradient descent methods (such as ADAM [6]) are primarily used in training neural networks because they are less computationally intensive and require less memory. But gradient descent methods have a significant drawback, which is a very low convergence rate [7, 8], especially for poorly conditioned problems. Quasi-Newton optimization methods have much higher convergence rates, but they require more memory and computational resources, although there are memory-limited variants of quasi-Newton methods (L-BFGS) [9] for moderately large dimensions. In the author's opinion, as computational hardware becomes more powerful in the future, quasi-Newton optimization methods will still be applied, both in machine learning in general and for training neural networks as well.

In solving practical problems in machine learning, such as tuning nonlinear regression models, the extremum point of the selected optimality criterion is often found to be degenerate, which significantly complicates its search. Therefore, degenerate problems are the most challenging in optimization. There is an even larger class of optimization problems in which the objective function is poorly conditioned in a neighborhood of the minimum point. Although these problems are not formally degenerate, existing numerical methods also have a low convergence rate in this case.

Known numerical methods for solving the general unconstrained optimization problem, up to the second order inclusive, have very low convergence rate when solving degenerate problems [7, 8]. This is because to significantly increase the convergence rate in this case, it is necessary to use derivatives of higher order than the second [10], but this makes numerical methods very computationally expensive.

Despite the development of fairly efficient Newton and quasi-Newton methods for unconstrained optimization [5, 7], interest in them has not waned [11–23]. In the case of solving degenerate unconstrained optimization problems,

an approach related to regularization of numerical methods is applied [11–18]. The essence of regularization of a Newton-type or quasi-Newton-type numerical method is to make the Hessian matrix, or its approximation, positive definite. This allows the method to work in a degenerate case but does not provide the opportunity to solve the problem with high precision.

This paper aims to develop an efficient quasi-Newton method for solving degenerate unconstrained optimization problems, the idea of which (unlike regularization) is to divide the entire space into the sum of two orthogonal subspaces. This idea was introduced in [23]. The division of the space at each iteration of the method is based on the spectral decomposition of the matrix approximating the Hessian of the objective function by the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula [3]. On each of the subspaces, the objective function exhibits distinct behavior, and therefore, an appropriate minimization method is applied to it. Specifically, a combination of the quasi-Newton method and a method with computing derivatives of the fourth order in the direction is used.

1. Combined quasi-Newton method

A degenerate unconstrained optimization problem is considered:

$$min\ f(x),\ x\epsilon R^n, \tag{1}$$

where $f(x)$ is a function that is four times differentiable, for which there exists $x^* \in R^n$ is a local minimum point of the function $f(x)$, and the Hessian matrix $f^{(2)}(x^*)$ is degenerate.

The quasi-Newton combined method for solving a problem (1) is proposed, which constructs an iterative sequence of approximations to the minimum point according to the formula:

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

where $x^{(0)}$ is the initial approximation of the minimum point, $u_1^{(k)},\ u_2^{(k)}$ are orthogonal vectors, $\alpha_{k1} > 0$, $\alpha_{k2} > 0$ are step multipliers along the respective directions $u_1^{(k)},\ u_2^{(k)}$. The vectors $u_1^{(k)},\ u_2^{(k)}$ are determined as follows.

At each k^{th} iteration of the method, the matrix H_k, which is an approximation of the Hessian matrix $f^{(2)}\left(x^{(k)}\right)$, is computed using the BFGS formula [3]:

$$H_{k+1} = H_k + \frac{y_k y_k{}^T}{y_k{}^T s_k} - \frac{H_k s_k (H_k s_k)^T}{s_k{}^T H_k s_k}, H_0 = I, \tag{3}$$

where $s_k = x^{(k)} - x^{(k-1)}$, $y_k = g^{(k)} - g^{(k-1)}$, the vector $g^{(k)} = f^{(1)}\left(x^{(k)}\right)$ and I is an identity matrix.

Since the matrix H_k, is symmetric, according to its spectral decomposition, it can be represented as

$$H_k = Q_k \Lambda_k Q_k^T, \tag{4}$$

where Q_k is an orthogonal matrix, $\Lambda_k = diag(\lambda_i^{(k)})$, and $\lambda_i^{(k)}(\ i = 1, ..., n)$ are the eigenvalues of the matrix H_k, ordered in descending order by absolute value.

Let's represent the diagonal matrix Λ_k in a block form as follows

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

where $\Lambda_{k1} = diag(\lambda_i^{(k)})$, $\left|\lambda_i^{(k)}\right| / \left|\lambda_1^{(k)}\right| > \varepsilon_k\ \ (i = 1, ..., r_k), r_k \le n, \varepsilon_k > 0$ is a parameter of the numerical method at the k^{th} iteration, $\Lambda_{k2} = diag(\lambda_i^{(k)})$, $\left|\lambda_i^{(k)}\right| / \left|\lambda_1^{(k)}\right| \le \varepsilon_k,\ (i = (r_k + 1), ..., n)$. Then we can also express the matrix Q_k in a block form as follows $Q_k = [\ Q_{k1}\quad Q_{k2}]$, where Q_{k1} is a block of size $n \times r_k$, and Q_{k2} is a block of size $n \times (n - r_k)$.

Now, the matrix H_k , in connection with (4), can be represented as follows:

$$H_k = [\, Q_{k1} \quad Q_{k2}] \begin{bmatrix} \Lambda_{k1} & 0 \\ 0 & \Lambda_{k2} \end{bmatrix} \begin{bmatrix} Q_{k1}^T \\ Q_{k2}^T \end{bmatrix} = Q_{k1}\Lambda_{k1}Q_{k1}^T + Q_{k2}\Lambda_{k2}Q_{k2}^T = H_{k\varepsilon} + E_{k\varepsilon}, \tag{5}$$

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

Next, we construct orthogonal projectors: $P_k = I - Q_{k1}Q_{k1}^T$ and $P_k^{\perp} = I - P_k = Q_{k1}Q_{k1}^T$ onto the subspace $Ker(H_{k\varepsilon}) = \{x \in R^n \mid H_{k\varepsilon}\, x = 0\}$ and its orthogonal complement, respectively. Note that the orthogonality of the matrix Q_k implies that $P_k = Q_{k2}Q_{k2}^T$. Now, different optimization methods can be applied on the subspaces $R_{k2} = Ker(H_{k\varepsilon})$ and $R_{k1} = R^n \ominus R_{k2}$ (its orthogonal complement), depending on the behavior of the function $f(x)$ on them. In this case, the parameter $\varepsilon_k > 0$ serves as a criterion for dividing the space into the orthogonal sum of two subspaces at each iteration. The algorithm for selecting this parameter will be described below.

Now, let's define the orthogonal vectors $u_1^{(k)}$ and $u_2^{(k)}$ in the method (2).

First, consider the projection of the function $f(x)$ onto the subspace R_{k1} in a neighborhood of the point $x^{(k)}$, that is, the function $f_{k1}(u_1) = f(x^{(k)} + u_1)$, where the vector $u_1 \epsilon\, R_{k1}$. We approximate the function $f_{k1}(u_1)$ by a Taylor series expansion up to the second order in a neighborhood of the point 0:

$$f_{k1}(u_1) = f(x^{(k)} + u_1) \approx f(x^{(k)}) + (P_k^{\perp} g^{(k)}, u_1) + \frac{1}{2} f^{(2)}(x^{(k)})[(u_1)^2].$$

Since H_k is an approximation of $f^{(2)}(x^{(k)})$, $H_k = H_{k\varepsilon} + E_{k\varepsilon}$ from (5), $E_{k\varepsilon}u_1 = 0$, then

$$f_{k1}(u_1) \approx f(x^{(k)}) + (P_k^{\perp} g^{(k)}, u_1) + \frac{1}{2} H_{k\varepsilon}[(u_1)^2].$$

Then the vector $u_1^{(k)}$ from (2) is determined as the point of the minimum of the approximation of the function $f_{k1}(u_1)$, i.e., from the equation:

$$\frac{\partial f_{k1}(u_1)}{\partial u_1} \approx P_k^{\perp} g^{(k)} + H_{k\varepsilon}u_1 = 0,$$

from which we obtain:

$$u_1^{(k)} = -H_{k\varepsilon}^{+} P_k^{\perp} g^{(k)} = -Q_{k1}\Lambda_{k1}^{-1} Q_{k1}^T P_k^{\perp} g^{(k)}, \tag{6}$$

where $H_{k\varepsilon}^{+}$ is a pseudoinverse matrix of the matrix $H_{k\varepsilon}$. This means we apply the quasi-Newton algorithm over the orthogonal complement of the subspace $Ker(H_{k\varepsilon})$.

Note that applying a similar approach to determine the vector $u_2^{(k)}$ in (2) on the subspace R_{k2} would require calculating the full arrays of $f^{(3)}(x^{(k)})$ and $f^{(4)}(x^{(k)})$, which would be very labor-intensive. Therefore, we will use a slightly different approach.

Let $u = x - x^{(k)}$ and make the change of variables $u' = Q_k^T u$. Then

$$u' = Q_k^T u = \begin{bmatrix} Q_{k1}^T \\ Q_{k2}^T \end{bmatrix} u = \begin{pmatrix} Q_{k1}^T u \\ Q_{k2}^T u \end{pmatrix} = \begin{pmatrix} u_1' \\ u_2' \end{pmatrix},$$

where $u_1' = Q_{k1}^T u$ is a vector of dimension r_k, and $u_2' = Q_{k2}^T u$ is a vector of dimension $n - r_k$.

Note that in the new coordinate system, the center coincides with the point $x^{(k)}$, the columns of the matrix Q_k^T are the basis vectors, and the elements of the vector $u' = \begin{pmatrix} u_1' \\ u_2' \end{pmatrix}$ are the coordinates. In this new coordinate system, the entire space is already the product of two subspaces to which the vectors u_1' and u_2' belong.

Now, consider the projections of the function $f(x)$ in a neighborhood of the point $x^{(k)}$ along the orthogonal vectors $q_{k2}^{(i)}$ $(i = (r_k + 1), \dots, n)$, i.e., the functions $f_{k2i}(\alpha) = f\left(x^{(k)} + \alpha \cdot q_{k2}^{(i)}\right)$, where $\alpha \in R^1$ and $q_{k2}^{(i)}$ is the i^{th} column of the matrix Q_{k2}. Approximate the functions $f_{k2i}(\alpha)$ by a Taylor series up to the fourth order in a neighborhood of point 0:

$$f_{k2i}(\alpha) \approx f\left(x^{(k)} + \alpha \cdot q_{k2}^{(i)}\right) \approx f_{k2i}(0) + \alpha \cdot f_{k2i}^{(1)}(0) + \frac{1}{2}\alpha^2 f_{k2i}^{(2)}(0) + \\ + \frac{1}{6}\alpha^3 f_{k2i}^{(3)}(0) + \frac{1}{24}\alpha^4 f_{k2i}^{(4)}(0). \quad (7)$$

Now, let's define the elements ${u'_{2i}}^{(k)}$ of the vector ${u'_2}^{(k)}$ as the points of minimum for the respective approximations of the functions $f_{k2i}(\alpha)$.

Since for $i = (r_k + 1), \dots, n$

$$f_{k2i}^{(2)}(0) = f^{(2)}\left(x^{(k)}\right)\left[\left(q_{k2}^{(i)}\right)^2\right] \approx (H_{k\varepsilon} + E_{k\varepsilon})\left[\left(q_{k2}^{(i)}\right)^2\right] = E_{k\varepsilon}\left[\left(q_{k2}^{(i)}\right)^2\right] = \lambda_i^{(k)},$$

where $\left|\lambda_i^{(k)}\right| \le \varepsilon_k \left|\lambda_1^{(k)}\right|$, then for small ε_k it will be $f_{k2i}^{(2)}(0) \approx 0$. Therefore, according to [24], the necessary fourth-order condition for the minimum point of the function $f_{k2i}(\alpha)$ from (7) will be 3 equations:

$$f_{k2i}^{(1)}(\alpha) \approx f_{k2i}^{(1)}(0) + \alpha \cdot f_{k2i}^{(2)}(0) + \frac{1}{2}\alpha^2 f_{k2i}^{(3)}(0) + \frac{1}{6}\alpha^3 f_{k2i}^{(4)}(0) = 0, \quad (8)$$

$$f_{k2i}^{(2)}(\alpha) \approx f_{k2i}^{(2)}(0) + \alpha \cdot f_{k2i}^{(3)}(0) + \frac{1}{2}\alpha^2 f_{k2i}^{(4)}(0) = 0, \quad (9)$$

$$f_{k2i}^{(3)}(\alpha) \approx f_{k2i}^{(3)}(0) + \alpha \cdot f_{k2i}^{(4)}(0) = 0, \quad (10)$$

From (10) and (9), it follows that $f_{k2i}^{(2)}(0) + \alpha \cdot \frac{1}{2} f_{k2i}^{(3)}(0) = 0$, and then from (8), we obtain

$$f_{k2i}^{(1)}(0) + \frac{1}{6}\alpha^3 \cdot f_{k2i}^{(4)}(0) = 0,$$

and finally:

$${u'_{2i}}^{(k)} = (-f_{k2i}^{(1)}(0)/f_{k2i}^{(4)}(0))^{1/3},\ i = (r_k + 1), \dots, n. \quad (11)$$

Since $u'_2 = Q_{k2}^T u$, the vector $u_2^{(k)}$ from (2) is determined by the formula

$$u_2^{(k)} = Q_{k2} \cdot {u'_2}^{(k)}. \quad (12)$$

Derivatives up to the 4th order of the functions $f_{k2i}(\alpha)$ of the scalar variable α are the derivatives of the function $f(x)$ in the directions $q_{k2}^{(i)}$ $(i = (r_k + 1), \dots, n)$. These derivatives can be calculated using symmetric formulas:

$$f_{k2i}^{(1)}(0) \approx \frac{-f_{k2i}(2h) + 8f_{k2i}(h) - 8f_{k2i}(h) + f_{k2i}(-2h)}{12h}, \quad (13)$$

$$f_{k2i}^{(2)}(0) \approx \frac{-f_{k2i}(2h) + 16f_{k2i}(h) - 30f_{k2i}(0) + 16f_{k2i}(h) - f_{k2i}(-2h)}{12h^2}, \quad (14)$$

$$f_{k2i}^{(3)}(0) \approx \frac{f_{k2i}(2h) - 3f_{k2i}(h) + 3f_{k2i}(h) - f_{k2i}(-2h)}{h^3}, \quad (15)$$

$$f_{k2i}^{(4)}(0) \approx \frac{f_{k2i}(2h) - 4f_{k2i}(h) + 6f_{k2i}(0) - 4f_{k2i}(-h) + f_{k2i}(-2h)}{h^4}, \quad (16)$$

where $h > 0$ is some small number called step size. As can be seen from formulas (13) – (16), this requires only 4 additional values of the function $f(x)$. Thus, to determine the vector $u_2^{(k)}$, an additional $4 \times (n - r_k)$ computations of the objective function $f(x)$ values are required, which, for small $(n - r_k)$, is not particularly significant.

Also, note that when the sufficient condition for the 4th-order minimum formulated in [24] is met, the values of $\cdot f_{k2i}^{(4)}(0)$ in a neighborhood of the minimum point will be strictly positive, making formula (11) valid.

1.1. Calculation of Step Lengths

In method (2), (6), (11), (12), it is necessary to determine two step lengths: α_{k1} and α_{k2} along the directions $u_1^{(k)}$ and $u_2^{(k)}$ respectively, in different orthogonal subspaces. On the one hand, this requires additional computations, but on the other hand, it allows the method to progress in the subspace of the matrix kernel H_k independently of progress in the other subspace.

The results of numerical experiments have shown that the most effective algorithm for determining the step lengths α_{k1} and α_{k2} is an algorithm similar to the method of coordinate descent [5, 6]. That is, initially, the step length $\alpha_{k1} > 0$ is determined as the point of approximate minimum of the function $\psi_1(\alpha) = f(x^{(k)} + \alpha\, u_1^{(k)})$, and then the step length $\alpha_{k2} > 0$ is determined as the point of approximate minimum of the function $\psi_2(\alpha) = f(x^{(k)} + \alpha_{k1}\, u_1^{(k)} + \alpha\; u_2^{(k)})$.

1.2. Selection of the regularization parameter for the quasi-Newton combined method

As mentioned earlier, the regularization parameter $\varepsilon_k > 0$ of the quasi-Newton combined method (2), (6), (11), (12) serves as a criterion for dividing the space into the orthogonal sum of two subspaces at each k^{th} iteration. It is clear that its value strongly affects the course of the iteration process. In the numerical implementation of the method (2), (6), (11), (12), the following algorithm was applied to select the parameter ε_k

Before applying the method, a range of possible values for $\varepsilon_k > 0$ is specified, for example, $[\varepsilon_{min,}\, \varepsilon_{max}]$ (in numerical experiments, $\varepsilon_{min,} = 10^{-11}$ and $\varepsilon_{max} = 10^{-3}$ were taken). Initially, for $k > 0$, $\varepsilon_k = \varepsilon_{min}$ is chosen. Then, when the iteration process of the method (2), (6), (11), (12) slows down, i.e., $\| x^{(k+1)} - x^{(k)} \| / (1 + \| x^{(k+1)} \|) \le \delta_{arg}$, where $\delta_{arg} > 0$ is a given parameter (for example, $\delta_{arg} = 10^{-10}$), ε_k is increased so that the rank of the matrix H_{k+1} becomes smaller on the next iteration, i.e., $r_{k+1} < r_k$. The iteration process of the method (2), (6), (11), (12) is then continued. The iteration process of the method (2), (6), (11), (12) is completed when, on the k-th iteration, $\frac{|\lambda_{r_k}^{(k)}|}{|\lambda_1^{(k)}|} > \varepsilon_{max}$. From this algorithm, it is clear that if, at all iterations, the condition number of the matrix $H_{k\varepsilon}$ satisfies $cond(H_{k\varepsilon}) < 1/\varepsilon_{max}$, then the described method coincides with the ordinary quasi-Newton method.

1.3. Analysis of the Convergence Rate of the Method

As the results of the numerical experiments presented in Section 3 show, the method (2), (6), (11), (12) exhibits a faster convergence rate (requiring fewer iterations to achieve accuracy) when solving the degenerate problem (1) compared to the algorithms used for comparison. However, this article does not include theoretical research on the convergence rate of the method (2), (6), (11), (12) in solving the degenerate problem (1) due to limitations on the article's scope. It is worth emphasizing that sufficient conditions of higher order for a degenerate minimum point can be applied for this purpose, as presented in [24], along with an approach to analyzing the convergence rate of the method in the case of a degenerate problem (1), as utilized in [25].

2. Results of the Numerical Experiments

The quasi-Newton combined method (2), (6), (11), (12) for solving the degenerate problem (1) was implemented in the Python environment. To ensure the stability of the method against divergence, negative diagonal elements λ_i^k of the Λ_k matrix in (4) were replaced by $-\lambda_i^k$. The derivatives $g^{(k)} = f^{(1)}(x^{(k)})$ were numerically computed using the symmetric formula:

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x|x_i + h_i) - f(x|x_i - h_i)}{2h_i} \tag{17}$$

with a step size $h_i = h_0 \cdot \max(1, |x_i|)$, where $h_0 = 10^{-7}$.

The method (2), (6), (11), (12) was tested on test functions for unconstrained optimization problems from the collection presented in [26]. In total, this collection contains 75 functions, so only optimization problems with degenerate minimum points were selected for testing the method. The numbering of the test functions used corresponds to the numbering in the collection [26]:

4. ***Extended White & Holstl***: $f(x) = \sum_{i=1}^{n/2}\left[(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1})^2\right]$, the initial estimate is $x^{(0)} = (-1.2, 1, \ \ldots, -1.2, 1)^T$, the minimum point is $x^* = (1, 1, \ \ldots, 1, 1)^T$, the value of the objective function at the minimum point is $f(x^*) = 0, rank(f^{(2)}(x^*)) = n - 1$.

15. ***Extended Tridiagonal 1***: $f(x) = \sum_{i=1}^{n/2}(x_{2i-1} + x_{2i} - 3)^2 + (x_{2i-1} - x_{2i} + 1)^4$, 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 $f(x^*) = 0, rank(f^{(2)}(x^*)) = n - 1$.

21. ***Extended Powell***: $f(x) = \sum_{i=1}^{n/4}\left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + \right.$ $\left. +10(x_{4i-3} - x_{4i})^4\right]$, the initial estimate is $x^{(0)} = (3, -1, 0, 1, \ldots, 3, -1, 0, 1)^T$, the minimum point is $x^* = (0, 0, \ \ldots, 0, 0)^T$, the value of the objective function at the minimum point is $f(x^*) = 0, rank(f^{(2)}(x^*)) = n/2$.

38. ***Extended Hiebert***: $f(x) = \sum_{i=1}^{n/2}(x_{2i-1} - 10)^2 + (x_{2i-1}x_{2i} - 500)^2$, the initial estimate is $x^0 = (0, 0, \ldots, 0)^T$, the minimum point is $x^* = (10, 50, \ldots, 10, 50)^T$, the value of the objective function at the minimum point is $f(x^*) = 0, rank(f^{(2)}(x^*)) = n - 1$.

45. ***Almost Perturbed Quartic***: $f(x) = (x_1 + x_n)^2/100 + \sum_{i=1}^{n} i{x_i}^4$, the initial estimate is $x^{(0)} = (0.5, 0.5, \ldots, 0.5)^T$, the minimum point is $x^* = (0, 0, \ldots, 0^T$, the value of the objective function at the minimum point is $f(x^*) = 0, rank(f^{(2)}(x^*)) \ll n$.

58. ***GENROSNB (CUTE)***: $f(x) = (x_1 - 1)^2 + \sum_{i=1}^{n} 100(x_i - {x_{i-1}}^2)^2$, the initial estimate is $x^{(0)} = (-1.2, 1, \ldots, -1.2, 1)^T$, the minimum point is $x^* = (1, 1, \ldots, 1, 1)^T$, the value of the objective function at the minimum point is $f(x^*) = 0, rank(f^{(2)}(x^*)) = n - 1$.

Additionally, the following functions were taken for testing:

77. ***Mean-square approximation by polynomials***: $f(x) = \sum_{j=1}^{101}\left[\sum_{i=1}^{n} x_i 0.01(j-1)^{i-1} - \right.$ $\left. - \sum_{i=1}^{n} x_i^* 0.01(j-1)^{i-1}\right]^2$, the initial estimate is $x^{(0)}$ = (2, 2, …, 2), the minimum point is x^* = (1, 1, …, 1), the value of the objective function at the minimum point is $f(x^*) = 0, rank(f^{(2)}(x^*)) = n - 1$.

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

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

The numerical experiments were conducted for almost all functions for dimensions $n = 4$ and $n = 400$, except for function 77, for which the dimensions were $n = 5$ and $n = 400$. Dimensions 4 and 5 were chosen because they are classical dimensions for these functions. Dimension 400 was chosen to analyze the performance of optimization procedures on problems with higher dimensions, as the results typically vary significantly between simple and complex problems.

The comparison was conducted with quasi-Newton procedures of mathematical packages: R, Python, Scilab, MATLAB. The following abbreviations are used in Tables 1–9 to denote these procedures:

R is 'optim' procedure (method 'BFGS') from R Online Compiler;

Py is 'minimize' procedure (method 'BFGS') from scipy version 1.11.4 (Google Colab);

Sc is 'optim' procedure (method 'gn') from Scilab on cloud;

M is 'fminunc' procedure (method 'quasi-newton') from MATLAB Online;

CQNAM is procedure implementing method (2), (6), (11), (12) in Python.

The search for the minimum point with all procedures was conducted with the highest possible accuracy.

The results of the numerical experiments are presented in Tables 1–9, where:

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

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

Nitr is a number of iterations performed;

Nf is a number of function evaluations performed;

Ngr is a number of gradient evaluations performed;

NormGr is the Euclidean norm $\left\|f^{(1)}(\tilde{x})\right\|$;

code is the termination code returned by the respective optimization procedure.

In the CQNAM procedure, the termination code (code) takes the following values:

0 is the specified gradient accuracy is achieved (specified as 10^{-20});

1 is the specified argument accuracy is achieved (specified as 10^{-10});

4 is maximum number of iterations is reached (set to 3000).

In the 'optim' procedure (Scilab package), the termination code (err) takes the following values:

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

5 is "Optim stops: maximum number of iterations is reached" (set to 3000);

9 is "End of optimization, successful completion".

In the scipy.minimize procedure (Python), the termination code (message) takes the following values:

0 is "Optimization terminated successfully";

1 is "Desired error not necessarily achieved due to precision loss";

2 is "CG iterations didn't converge. The Hessian is not positive definite";

4 is "Maximum number of iterations has been exceeded" (set to 3000).

In the fminunc procedure (MATLAB package), the termination code (exitflag) takes the following values:

5 is "Predicted decrease in the objective function was less than the FunctionTolerance tolerance" (set to 10^{-40}).

The results of numerical experiments are presented in Tables 1–9. As seen from Tables 1–9, the worst performance on all examples is observed with the 'fminunc' procedure from the MATLAB package. Next is the 'optim' procedure from the R package. Better results are achieved with the 'scipy.minimize' procedure (Python) and the 'optim' procedure from the Scilab package, although the 'optim' procedure from the Scilab

package performed poorly on function 38. It is also noteworthy that the algorithm of the 'scipy.minimize' (Python) procedure requires more iterations to achieve the desired accuracy. Overall, the CQNAM procedure (implementing the method presented in this paper) gives a decent result on almost all examples, while applying significantly fewer iterations compared to other procedures. This is especially important for high dimensions, as one numerical gradient computation according to formula (17) is equivalent to 2*n* function evaluations. Although the CQNAM procedure uses a higher number of function evaluations on examples 21 and 45 compared to the 'optim' procedure from the Scilab package, it is still significantly fewer in the total number of function evaluations considering both function and gradient evaluations. It should also be noted that the high number of function evaluations by the CQNAM procedure on examples 21 and 45 is due to the application of formulas (15), (16), as well as the fact that in examples 21 and 45, the rank of the degenerate Hessian in a neighborhood of the minimum point is very high, namely, $rank(f^{(2)}(x^*)) = n/2$. Therefore, it can be concluded that the application of the method (2), (6), (11), (12) is most effective for problems with a low rank of degeneracy of the Hessian in a neighborhood of the minimum point. In the case of a large rank of Hessian degeneration in a neighborhood of the minimum point, it is better to apply the variant of the method described in [23].

TABLE 1. Calculation results for the function **4** for $n = 4$ and $n = 400$ ($rank(f^{(2)}(x^*)) = n - 1$)

	$n = 4$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	3.6e-03	1.3e-06		167	55	0.25		4.0e-02	1.6e-04		2383	810	2.5	
Py	4.0e-11	1.6e-22	74	105	94	1.3e-13	1	5.3e-10	2.8e-20	1733	2047	2047	3.6e-10	1
Sc	1.0e-07	1.9e-15	41	80		9.3e-08	8	1.5e-06	2.1e-13	389	432		9.2e-10	9
M	3.9e-05	1.5e-10	36	215		2.3e-06	5	3.6e-04	1.3e-08	45	22857		6.2e-04	5
CQNAM	2.4e-15	1.8e-30	44	275	45	6.8e-14	1	1.6e-13	3.3e-27	668	10571	669	1.6e-12	1

TABLE 2. Calculation results for the function **15** for $n = 4$ and $n = 400$ ($rank(f^{(2)}(x^*)) = n - 1$)

	$n = 4$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	2.6e-04	9.8e-15		83	66	6.1e-07		4.2e-09	1.8e-22		45	31	3.9e-11	
Py	4.5e-09	8.6e-34	56	73	73	1.0e-21	0	9.7e-05	2.6e-18	124	194	187	4.2e-12	1
Sc	3.0e-07	1.3e-26	52	278		1.5e-15	9	2.6e-06	2.0e-24	63	537		2.4e-12	1
M	1.0e-05	2.5e-16	25	170		1.4e-08	5	1.0e-04	2.5e-14	25	14837		1.4e-07	5
CQNAM	7.9e-08	7.8e-29	35	196	36	1.2e-15	1	3.1e-06	1.9e-24	39	300	40	1.4e-14	1

TABLE 3. Calculation results for the function **21** for $n = 4$ and $n = 400$ ($rank(f^{(2)}(x^*)) = n/2$)

	$n = 4$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	5.0e-04	2.7e-13		320	168	2.3e-06		4.4e-04	7.3e-16		1939	666	1.6e-06	
Py	6.2e-07	1.7e-25	97	115	103	2.0e-12	1	9.8e-04	3.7e-14	3000	3150	3150	6.5e-14	4
Sc	3.1e-09	9.7e-35	89	123		1.0e-22	0	1.5e-06	1.5e-25	960	13977		4.6e-18	1
M	1.0e-05	2.5e-16	25	170		1.4e-08	5	1.3e-02	1.1e-09	39	19248		2.5e-04	5
CQNAM	3.2e-08	9.4e-30	76	806	77	6.6e-16	1	7.0e-06	5.9e-23	754	18204	755	4.6e-16	1

TABLE 4. Calculation results for the function **38** for $n = 4$ and $n = 400$ $(rank(f^{(2)}(x^*)) = n - 1)$

	$n = 4$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	5.4e-09	1.3e-18		313	120	3.80e-08		1.8e-09	5.8e-19		5023	1617	6.9e-08	
Py	0.0	0.0	114	153	153	0.0	0	2.5e-09	2.5e-19	2815	3242	3230	2.2e-10	1
Sc	2.4e-13	2.2e-27	94	165		3.9e-12	9	55156.	100.1	3001	3551		2945.7	5
M	3.4e-07	2.7e-11	77	618		3.7e-06	5	2.8e-04	5.8e-09	81	47719		2.9e-04	5
CQNAM	7.3e-15	3.1e-30	18	171	19	5.6e-13	1	4.1e-10	6.7e-21	40	726	41	1.0e-10	1

TABLE 5. Calculation results for the function **45** for $n = 4$ and $n = 400$ $(rank(f^{(2)}(x^*)) \ll n)$

	$n = 4$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	2.6e-04	4.8e-15		62	48	3.1e-07		6.4e-04	1.1e-13		17158	3000	1.2e-06	
Py	1.9e-08	9.8e-31	142	218	206	2.9e-17	1	6.2e-04	7.3e-14	3000	3003	3003	5.8e-10	4
Sc	3.6e-13	1.5e-50	176	242		1.4e-30	1	7.0e-07	7.7e-26	3001	3098		5.7e-19	1
M	7.1e-04	1.5e-13	51	312		1.1e-30	5	6.4e-04	1.0e-13	285	115087		8.6e-10	5
CQNAM	2.2e-08	2.2e-31	58	526	59	4.0e-23	0	2.6e-06	1.9e-23	1200	6624	1201	4.3e-17	1

TABLE 6. Calculation results for the function **58** for $n = 4$ and $n = 400$ $(rank(f^{(2)}(x^*)) = n - 1)$

	$n = 4$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	1.2e-02	1.9e-06		384	106	6.8e-02		19.8	1.3e-04		1178	373	8.9e-02	
Py	1.2e-10	1.9e-22	66	95	84	2.4e-13	1	19.6	1.3e-08	3000	3943	3943	4.8e-05	4
Sc	2.0e-07	9.2e-16	66	133		1.3e-06	9	19.6	6.6e-09	3001	3760		1.1e-05	5
M	2.8e-04	9.7e-10	63	486		5.3e-05	5	19.7	5.3e-06	357	192881		3.3e-03	5
CQNAM	8.1e-15	7.8e-31	55	202	56	1.7e-15	1	19.6	4.9e-09	3000	16391	3001	6.6e-05	4

TABLE 7. Calculation results for the function **77** for $n = 5$ $(rank(f^{(2)}(x^*)) = n)$ and $n = 400$ $(rank(f^{(2)}(x^*)) = n - 1)$

	$n = 5$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	5.2e-07	1.0e-16		582	357	1.2e-08		3.5e-06	5.8e-16		65	49	1.9e-09	
Py	1.8e-13	4.1e-29	35	78	72	1.8e-14	1	1.8e-02	5.8e-14	143	195	183	6.9e-11	1
Sc	1.0e-12	3.8e-28	27	51		2.6e-14	9	9.0e-07	4.8e-18	55	155		1.2e-11	9
M	1.3e-06	9.5e-15	50	306		1.5e-13	5	3.7e-05	1.0e-13	91	36892		1.0e-08	5
CQNAM	1.2e-14	5.8e-30	12	81	13	1.9e-14	1	1.8e-06	1.2e-19	39	564	40	2.8e-11	1

TABLE 8. Calculation results for the function **78** for $n = 4$ and $n = 400$ ($rank(f^{(2)}(x^*)) = n - 1$)

	$n = 4$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	3.7e-03	2.0e-13		155	85	1.7e-09		1.8e-07	2.3e-25		74	56	9.7e-13	
Py	1.5e-08	5.2e-35	83	175	168	2.2e-16	2	2.7e-07	1.3e-29	81	163	158	3.3e-14	1
Sc	8.3e-14	4.8e-56	116	772		4.4e-16	9	2.6e-06	7.9e-25	64	102		1.7e-12	9
M	1.0e-03	5.5e-08	55	320		2.5e-07	5	4.4e-03	5.6e-08	54	28872		1.1e-07	5
CQNAM	4.6e-08	4.5e-33	27	187	28	2.1e-23	0	1.7e-11	9.4e-47	58	371	59	2.2e-14	1

TABLE 9. Calculation results for the function **79** for $n = 4$ and $n = 400$ ($rank(f^{(2)}(x^*)) = n - 1$)

	$n = 4$							$n = 400$						
	Dx	Df	Nitr	Nf	Ngr	NormGr	Code	Dx	Df	Nitr	Nf	Ngr	NormGr	Code
R	1.1e-05	1.3e-20		75	57	4.6e-09		4.6e-09	6.0e-22		59	43	4.9e-11	
Py	1.6e-12	2.2e-41	109	261	261	1.2e-21	0	7.7e-14	2.1e-41	112	129	129	9.3e-21	0
Sc	1.4e-21	3.3e-84	205	225		4.2e-43	1	1.2e-21	2.0e-84	274	3262		1.2e-42	1
M	7.1e-05	2.5e-17	67	365		3.0e-08	5	8.4e-05	4.7e-16	85	38496		3.3e-07	5
CQNAM	6.0e-09	1.0e-33	42	264	43	1.1e-17	1	1.5e-09	4.0e-36	54	319	55	1.5e-19	1

Conclusion

The combined quasi-Newton method presented offers a solution for degenerate unconstrained optimization problems. It is based on an orthogonal decomposition of the approximate Hessian matrix and dividing the entire space into two orthogonal subspaces. On one subspace (the kernel of the approximate Hessian matrix), a fourth-order method is applied, while on its orthogonal complement, a quasi-Newton method is utilized. Each of these subspaces employs a separate one-dimensional search to determine the step size in the corresponding direction.

The idea of splitting the entire space into the sum of two (or more) orthogonal subspaces when solving complex optimization problems is quite promising in terms of applying a combination of different numerical methods on separate subspaces.

The effectiveness of the presented combined method is confirmed by numerical experiments conducted on widely accepted test functions for unconstrained optimization problems. It is noteworthy that the proposed method allows obtaining quite accurate solutions to test tasks in the case of degenerate minima with significantly lower costs for computing the gradients of the objective function compared to optimization procedures in well-known mathematical packages.

Plans include conducting theoretical research on the convergence rate of the presented combined method in solving degenerate optimization problems (1).

There is hope that as computational technology becomes more efficient in the future, quasi-Newton methods will be applied in solving optimization problems in machine learning. They have a much faster convergence rate than gradient descent methods, especially in poorly conditioned and degenerate problems.

Acknowledgement

The author would like to thank the reviewer for his very helpful remarks and valuable comments.

References

1. Ring W. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*. 2012. **22 (**2). P. 596–627. https://doi.org/10.1137/11082885X
2. Maratos N.G. Some results on the Sign recurrent neural network for unconstrained minimization. *Neurocomputing*. 2017. **287.** P. 1–21. https://doi.org/10.1016/j.neucom.2017.09.036
3. Tirer T., Bruna J. Extended Unconstrained Features Model for Exploring Deep Neural Collapse. *Proceedings of the 39th International Conference on Machine Learning, in Proceedings of Machine Learning Research.* 2012. 162: 21478–21505. Available from https://proceedings.mlr.press/v162/tirer22a.html
4. Wang Z., Li P., Li X., Pham H. A Modified Three-Term Type CD Conjugate Gradient Algorithm for Unconstrained Optimization Problems. *Mathematical Problems in Engineering*. 2020. P. 1–14. *(Machine Learning and its Applications in Image Restoration)* https://doi.org/10.1155/2020/4381515
5. Berahas A.S., Jahani M., Richtárik P., Takáč M. Quasi-Newton methods for machine learning: forget the past, just sample. *Optimization Methods and Software*. 2022. **37** (5). P. 1668–1704. https://doi.org/10.1080/10556788.2021.1977806
6. Kingma D.P., Ba J. Adam: A Method for Stochastic Optimization. 2014. https://arxiv.org/abs/1412.6980
7. Avriel M. Nonlinear Programming: Analysis and Methods. *Dover Publishing.* 2003.
8. Nocedal J., Wright S.J. Numerical Optimization. *New York: Springer.* 2006.
9. Liu D.C., Nocedal J. On the Limited Memory Method for Large Scale Optimization. *Mathematical Programming.* 1989. **45** (3). P. 503–528. https://doi.org/10.1007/BF01589116
10. Birgin E.G., Gardenghi J.L., Martínez J.M., Santos S.A. On the use of third-order models with fourth-order regularization for unconstrained optimization. *Optimization Letters.* 2020. **14.** P. 815–838. https://doi.org/10.1007/s11590-019-01395-z
11. Li D.H., Fukushima M., Qi L., Yamashita N. Regularized newton methods for convex minimization problems with singular solutions. *Comput. Optim. Appl.* 2004. **28.** P. 131–147. https://doi.org/10.1023/B:COAP.0000026881.96694.32
12. Shen C., Chen X., Liang Y. A regularized Newton method for degenerate unconstrained optimization problems. *Optimization Letters*. 2012. **6.** P. 1913–1933. https://doi.org/10.1007/s11590-011-0386-z
13. Graspa T. N. A modified Newton direction for unconstrained optimization. *Optimization*. 2014. **63** (7). P. 983–1004. https://doi.org/10.1080/02331934.2012.696115
14. Taheri S., Mammadov M., Seifollahi S. Globally convergent algorithms for solving unconstrained optimization problems. *Optimization*. 2015. **64** (2). P. 249–263. https://doi.org/10.1080/02331934.2012.745529
15. Li X., Wang B., Hu W. A modified nonmonotone BFGS algorithm for unconstrained optimization. *Journal of Inequalities and Applications.* 2017. **183.** https://doi.org/10.1186/s13660-017-1453-5
16. Ghazali K., Sulaiman J., Dasril Y., Gabda D. Newton-SOR Iteration for Solving Large-Scale Unconstrained Optimization Problems with an Arrowhead Hessian Matrices. *Journal of Physics: Conference Series*. 2019. **1358 (**1). P. 1–10. https://doi.org/10.1088/1742-6596/1358/1/012054
17. Niri T.D., Hosseini M.M., Heydari M. An efficient improvement of the Newton method for solving nonconvex optimization problems. *Computational Methods for Differential Equations*. 2019. **7 (**1). P. 69–85.
18. Wang H., Qin M. A Regularized Newton Method with Correction for Unconstrained Nonconvex Optimization. *Journal of Mathematics Research*. 2015. **13 (**2). P. 7–17. https://doi.org/10.5539/jmr.v7n2p7
19. Li Q. A Modified Fletcher-Reeves-Type Method for Nonsmooth Convex Minimization. *Stat., Optim. Inf. Comput.* 2014. **2.** P. 200–210. https://doi.org/10.19139/soic.v2i3.64
20. Andrei N. A new accelerated diagonal quasi-Newton updating method with scaled forward finite differences directional derivative for unconstrained optimization. *Optimization.* 2021. **70** (2). P. 345–360. https://doi.org/10.1080/02331934.2020.1712391
21. Cartis C., Gould N.I.M., Toint Ph.L. A concise second-order complexity analysis for unconstrained optimization using high-order regularized models. *Optimization Methods and Software.* 2020. **35** (2). P. 243–256. https://doi.org/10.1080/10556788.2019.1678033
22. Yang Y. A robust BFGS algorithm for unconstrained nonlinear optimization problems. *Optimization*. 2024. **73** (3). P. 851–873. https://doi.org/10.1080/02331934.2022.2124869
23. Zadachyn V. M. Combined method for solving degenerate problems of unconditional optimization. *Information processing systems*. 2020. **1** (160). P. 52–58. (in Ukrainian) https://doi.org/10.30748/soi.2020.160.06
24. Zadachyn V.M. Higher-order Optimality Conditions for Degenerate Unconstrained Optimization Problems. *Journal of Optimization, Differential Equations and Their Application*. 2022. **30** (1). P. 88–97. https://doi.org/10.15421/142204

25. Zadachyn V.M. On the rate of convergence of the gradient method with a degenerate minimum. *USSR Cybernetics* (ISSN 0023-1274). 1989. **1.** P. 99–101. (in Russian)
26. Andrei N. Collection of 75 Unconstrained Optimization Test Functions. *Research Institute for Informatics, Technical Report.* 2018. **6.** P. 1–9.

Received 21.04.2024

Viktor Zadachyn,
PhD, Associate Professor, Department of Information Systems,
Simon Kuznets Kharkiv National University of Economics, Ukraine.
https://orcid.org/0000-0002-8107-4639
zadachinvm@gmail.com

UDC 519.853.6 : 519.613.2

Viktor Zadachyn

A Combined Quasi-Newton-Type Method Using 4th-Order Directional Derivatives for Solving Degenerate Problems of Unconstrained Optimization

Simon Kuznets Kharkiv National University of Economics, Ukraine
Correspondence: zadachinvm@gmail.com

Introduction. Methods of unconstrained optimization play a significant role in machine learning [1–6]. When solving practical problems in machine learning, such as tuning nonlinear regression models, the extremum point of the chosen optimality criterion is often degenerate, which significantly complicates its search. Therefore, degenerate problems are the most challenging in optimization. Known numerical methods for solving the general unconstrained optimization problem, up to the second order, have very low convergence rates when solving degenerate problems [7, 8]. This is explained by the fact that for a significant improvement in convergence rate in this case, it is necessary to use higher-order derivatives in the method than the second order [10].

The purpose of the paper is to develop an efficient quasi-Newton method for solving degenerate unconstrained optimization problems, the idea of which (unlike regularization) involves dividing the entire space into the sum of two orthogonal subspaces. This idea was introduced in [23]. The space division at each iteration of the method is based on the spectral decomposition of the matrix approximating the Hessian of the objective function using the BFGS formula [3]. Each subspace exhibits its own behavior of the objective function, and therefore, an appropriate minimization method is applied on it.

Results. A combined quasi-Newton method is presented for solving degenerate unconstrained optimization problems, based on orthogonal decomposition of the Hessian approximation matrix and division of the entire space into the sum of two orthogonal subspaces. On one subspace (the kernel of the Hessian approximation matrix), a method is applied where derivatives in the direction of the 4th order are computed, while on the orthogonal complement to it, a quasi-Newtonian method is applied. A separate one-dimensional search is applied on each of these subspaces to determine the step multiplier in the respective direction.

The effectiveness of the presented combined method is confirmed by numerical experiments conducted on widely accepted test functions for unconstrained optimization problems. The proposed method allows obtaining fairly accurate solutions to test tasks in case of degeneracy of the minimum point with significantly lower computational costs for gradient calculations compared to optimization procedures of well-known mathematical packages.

Conclusions. The idea of dividing the entire space into the sum of two (or possibly more) orthogonal subspaces when solving complex optimization problems is quite promising in terms of applying a combination of different numerical methods on separate subspaces.

In the future, it is planned to conduct theoretical research on the convergence rate of the presented combined method in solving degenerate unconstrained optimization problems.

Keywords: unconstrained optimization, quasi-Newton methods, degenerate minimum point, spectral matrix decomposition, Machine Learning.

УДК 519.853.6 : 519.613.2

В.М. Задачин

Комбінований метод квазі-ньютоновського типу з застосуванням похідних по напрямку 4-го порядку для розв'язання вироджених задач безумовної оптимізації

Харківський національний економічний університет імені С. Кузнеця, Україна
Листування: zadachinvm@gmail.com

Вступ. Методи безумовної оптимізації мають велике значення у машинному навчанні [1–6]. Під час розв'язання практичних задач у машинному навчанні, наприклад, при налаштуванні регресійних нелінійних моделей, точка екстремуму обраного критерію оптимальності нерідко виявляється виродженою, що значно ускладнює її пошук. Тому саме виродженні задачі є найбільш складними в оптимізації. Відомі чисельні методи розв'язання загальної задачі безумовної оптимізації, до другого порядку включно, мають дуже низьку швидкість збіжності в разі розв'язання вироджених задач [7, 8]. Це пояснюється тим, що для істотного підвищення швидкості збіжності в цьому випадку необхідно використання у методі похідних більш високого порядку, ніж другий [10].

Мета роботи. Розроблення ефективного методу квазі-ньютоновського типу для розв’язання вироджених задач безумовної оптимізації, ідея якого (на відміну від регуляризації) полягає у поділі всього простору на суму двох ортогональних підпросторів. Ця ідея була представлена в роботі [23]. Поділ простору на кожній ітерації методу засновано на спектральному розкладанні матриці, що наближує гессіан цільової функції по формулі BFGS [3]. На кожному підпросторі є своя поведінка цільової функції, і тому на ньому застосовується відповідний метод мінімізації.

Результати. Розроблено комбінований метод квазі-ньютоновського типу для розв'язання вироджених задач безумовної оптимізації, заснований на ортогональному розкладанні наближення матриці Гессе та поділі всього простору на суму двох ортогональних підпросторів. На одному підпросторі (ядрі наближення матриці Гессе) застосовується метод, в якому обчислюються похідні по напрямку 4-го порядку, а на ортогональному доповненні до нього – квазі-ньютоновський метод. На кожному з цих підпросторів застосовується окремий одномірний пошук для визначення крокового множника у відповідному напрямку.

Ефективність представленого комбінованого методу підтверджується чисельними експериментами, які були проведені на загальноприйнятих тестових функціях для задач безумовної оптимізації. Запропонований метод дозволяє отримати досить точні розв'язки тестових завдань у разі виродження точки мінімуму зі значно меншими затратами на обчислення градієнтів цільової функції, ніж це роблять процедури оптимізації відомих математичних пакетів.

Висновки. Ідея поділу всього простору на суму двох (а може і більше) ортогональних підпросторів під час розв’язання складних задач оптимізації є доволі перспективною з точки зору застосування комбінації різних чисельних методів на окремих підпросторах.

В подальшому планується провести теоретичне дослідження швидкості збіжності представленого комбінованого методу в разі розв'язання виродженої задачі безумовної оптимізації.

Ключові слова: безумовна оптимізація, квазі-ньютоновські методи, вироджена точка мінімуму, спектральне розкладання матриці, Machine Learning.