

Article

Properties and Application of Incomplete Orthogonalization in the Directions of Gradient Difference in Optimization Methods

Vladimir Krutikov ^{1,2} , Elena Tovbis ¹ , Svetlana Gutova ², Ivan Rozhnov ^{1,3}  and Lev Kazakovtsev ^{1,3,*} 

¹ Institute of Informatics and Telecommunications, Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarskii Rabochii Prospekt, Krasnoyarsk 660037, Russia; krutikovvn@rambler.ru (V.K.); sibstu2006@rambler.ru (E.T.)

² Department of Applied Mathematics, Kemerovo State University, 6 Krasnaya Street, Kemerovo 650043, Russia; gutova_ok@mail.ru

³ Laboratory “Hybrid Methods of Modeling and Optimization in Complex Systems”, Siberian Federal University, 79 Svobodny Prospekt, Krasnoyarsk 660041, Russia

* Correspondence: levk@bk.ru

Abstract

This paper considers the problem of unconstrained minimization of smooth functions. Despite the high efficiency of quasi-Newton methods such as BFGS, their performance degrades in ill-conditioned problems with unstable or rapidly varying Hessians—for example, in functions with curved ravine structures. This necessitates alternative approaches that rely not on second-derivative approximations but on the topological properties of level surfaces. As a new methodological framework, we propose using a procedure of incomplete orthogonalization in the directions of gradient differences, implemented through the iterative least-squares method (ILSM). Two new methods are constructed based on this approach: a gradient method with the ILSM metric (HY_g) and a modification of the Hestenes–Stiefel conjugate gradient method with the same metric (HY_XS). Both methods are shown to have linear convergence on strongly convex functions and finite convergence on quadratic functions. A numerical experiment was conducted on a set of test functions. The results show that the proposed methods significantly outperform BFGS (2 times for HY_g and 3.5 times for HY_XS in terms of iterations number) when solving ill-posed problems with varying Hessians or complex level topologies, while providing comparable or better performance even in high-dimensional problems. This confirms the potential of using topology-based metrics alongside classical quasi-Newton strategies.

Keywords: minimization method; relaxation subgradient method; iterative least squares; convergence rate

MSC: 90C30



Academic Editor: Shin-ya Matsushita

Received: 10 November 2025

Revised: 8 December 2025

Accepted: 16 December 2025

Published: 18 December 2025

Citation: Krutikov, V.; Tovbis, E.; Gutova, S.; Rozhnov, I.; Kazakovtsev, L. Properties and Application of Incomplete Orthogonalization in the Directions of Gradient Difference in Optimization Methods. *Mathematics* **2025**, *13*, 4036. <https://doi.org/10.3390/math13244036>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

We consider the problem of minimizing a differentiable function $f(x)$ on a finite-dimensional Euclidean space. This problem can be stated as

$$f(x) \rightarrow \min, x \in R^n.$$

Well-known methods [1,2] were developed to solve an unconstrained minimization problem, including the gradient method, which is based on the idea of function local linear approximation. Conjugate gradient methods (CGM) generate search directions that are

consistent with the geometry of the minimized function. In practice, the CGM shows faster convergence rates than gradient descent algorithms, so CGM is widely used in machine learning.

Quasi-Newton methods (QNM) are based on the idea of using a matrix of second derivatives reconstructed from the gradients of a function. The commonly used matrix update formula for quasi-Newton methods is BFGS [3–6]. Two quasi-Newton algorithms for solving unconstrained optimization problems based on two modified secant relations to achieve reliable approximations of the Hessian matrices of the objective function were presented in [7].

Ultra-high dimensions and strong nonlinearity can lead to extremely complex optimization landscapes, for which gradient-based optimization solvers will perform poorly or fail easily [8]. Without denying the merits of the gradient descent method, it must be said that it turns out to be very slow when moving along a ravine, and as the number of variables of the objective function increases, such behavior of the method becomes typical. Performance of QNM degrades in ill-conditioned problems with unstable or rapidly varying Hessians—for example, in functions with curved ravine structures.

Relaxation subgradient methods (RSM) have been widely used in optimization practice for many years and have found practical application in such areas as signal and image processing [9,10], classification [11], network design [12], maintenance routing [13], dynamic process modeling [14], and many others.

In our earlier studies, it turned out that the problem of finding the direction of descent in the RSM can be reduced to the problem of solving a system of inequalities on subgradient sets and mathematically formulated as a solution to the problem of minimizing some quality functional. In this case, the properties of the learning algorithm determined the convergence rate of the minimization method.

In relaxation processes of ε -subgradient type, successive approximations are constructed as follows:

$$x_{k+1} = x_k - \gamma_k s_k, \quad \gamma_k = \arg \min_{\gamma} f(x_k - \gamma s_k), \quad k = 0, 1, 2, \dots \quad (1)$$

Here, k is the iteration number, γ_k is the stepsize, and descent direction s_k is selected from the set $S(\partial_{\varepsilon} f(x_i))$ [1,15–19], $\partial_{\varepsilon} f(x_i)$ is ε -subgradient set at a point x_k , and $S(G) = \{s \in R^n \mid \min_{g \in G} (s, g) > 0\}$, $G \subset R^n$ is a set of feasible directions, g are gradients.

Denote by $\partial f(x) \equiv \partial f_{\varepsilon=0}(x)$ a subgradient set at a point x . If set $S(G)$ is not empty, then any vector $s \in S(G)$ is a solution to the set of inequalities:

$$(s, g) > 0, \quad \forall g \in G, \quad (2)$$

in other words, it sets the normal of the dividing plane of the origin and the set G . Here, (s, g) is a dot product of vectors. One of the solutions to (2) is a vector of minimal length from G denoted as $\eta(G)$.

The subgradient method was first proposed by Shor [15,20]. A number of effective approaches arose as a result of the development of the first subgradient methods with a space dilation [21,22]. The first relaxation subgradient methods were suggested in [23–25].

In Ref. [26], a method to solve the convex problems of nondifferentiable optimization relying on the basic philosophy of the conjugate gradient method and coinciding with it in the case of quadratic functions was presented. Authors in [27] propose a family of adaptive subgradient methods that dynamically incorporate knowledge of the geometry of the data observed in earlier iterations to perform more informative gradient-based learning. Spectral-step subgradient method for solving nonsmooth unconstrained optimization problems was demonstrated in [28]. In Ref. [29], a subgradient method based on step size

adjusting is developed to solve the system of nonsmooth equations. Subgradient projection methods are often applied to large-scale problems with decomposition techniques. Adaptive projection subgradient methods was proposed in [30,31]. Authors in [32] consider a projected-subgradient type method, derived from using a general distance-like function instead of the usual Euclidean squared distance.

As in QNM, RSM uses ratios to calculate the required characteristics. These relations represent equalities, solving which it is possible to determine the desired values. Due to the multiplicity of these equalities, it is necessary to resort to their solution by gradient methods as information becomes available. Thus, the use of the iterative least squares method (ILSM) is possible in this case.

In this paper, we consider modifications of subgradient methods applied to solving smooth minimization problems. In this case, the system of inequalities (2) will be formed by a set of gradients of some neighborhood of the current minimum. The descent direction s satisfying (2) will allow the method to go beyond this neighborhood as a result of the minimization step along this direction.

The main contributions of this paper are as follows:

(1) A methodological approach was proposed based on a procedure of incomplete orthogonalization in the directions of gradient differences, implemented through the iterative least-squares method. This approach uses the structural characteristics of the level surfaces instead of second-derivative approximations.

(2) Two methods were constructed based on this approach: a gradient method with the ILSM metric and a modification of the Hestenes–Stiefel conjugate gradient method with the same metric. Both methods were implemented and numerically studied comparing with the quasi-Newton BFGS method on various types of smooth functions. The test results indicate the efficiency of the proposed methods, especially when solving poorly conditioned problems with complex curved ravines and unstable characteristics of the second derivatives.

(3) The noted linear transformation of coordinates eliminates the linear background that worsens the convergence of the gradient method. In this work, we proved that algorithm HY_g, where the gradient method acquires accelerating abilities due to the use of the proposed metric transformation, has similar properties to Newton’s method, quasi-Newton methods, and subgradient methods with a change in the space metric. The qualitative nature of the convergence rate estimates for algorithm HY_g and Newton’s method coincide. Also, the equivalence of the conjugate gradient method with the new metric (HY_XS) to the conjugate gradient method on quadratic functions is proven.

The results obtained allow us to conclude that it is possible to use the studied methods along with quasi-Newton methods to solve smooth optimization problems with a high degree of conditionality.

The rest of the paper is organized as follows. Section 2 describes our variation in the gradient minimization method. In Section 3, we analyze the properties and convergence rate of the proposed algorithm. In Section 4, we study acceleration properties of the space dilation algorithm in the direction of the gradient difference. In Section 5, we propose the Hestenes–Stiefel method in a metric with incomplete orthogonalization in the direction of the gradient difference. In Section 6, we present the results of numerical experiments. Section 7 concludes the work.

2. Gradient Minimization Method with Incomplete Orthogonalization in the Direction of the Gradient Difference

For the gradients on the descent trajectory of (1), we use the notation $\nabla f(x_k) = g(x_k) = g_k$. The idea of creating an iterative method for solving the system of inequalities (2) is based on its transformation into a system of equalities [17–19].

To derive the formulas of the iterative process, we use an idealized model of the set G [17–19]. Let $G \subset R^n$ belongs to a certain hyperplane, and the vector $\eta(G)$ is also the vector of minimal length of this hyperplane. Then there is a solution to the system of equalities:

$$(s, g) = 1, \forall g \in G, \quad (3)$$

which simultaneously satisfies (2). Here, s is a descent direction, g is the gradient. The solution of system (3) can be obtained as the solution of the system of equalities:

$$(s, g_i) = 1, i = 0, 1, \dots, k. \quad (4)$$

One of the possible solutions to the system (4) can be found in the form of $s_{k+1} = \arg \min_s F_k(s)$, where $F_k(s)$ is the sum-of-squares function of the residuals:

$$F_k(s) = \sum_{i=0}^k w_i Q_i(s) + \frac{1}{2} \sum_{i=1}^n s_i^2, \dots Q_i(s) = \frac{1}{2} (1 - (s, g_i))^2.$$

Here, w_i are weighting factors. Such a solution, taking into account the regularizing component $\frac{1}{2} \sum_{i=1}^n s_i^2$, can be obtained by the iterative least squares method (ILSM) [33].

In Ref. [19], based on ILSM, an iterative process of finding the descent direction for the minimization method (1) using training information (4) was obtained using special weighting factors w_i in $F_k(s)$:

$$s_{k+1} = s_k + \frac{H_k g_k (1 - (s_k, g_k))}{(g_k H_k g_k)}, s_0 = 0. \quad (5)$$

$$H_{k+1} = H_k - (1 - \frac{1}{\alpha_k^2}) \frac{H_k g_k g_k^T H_k^T}{(g_k, H_k g_k)}, H_0 = I, \quad (6)$$

where $\alpha_k > 1$ is space dilation parameter, H_k is metric matrix. In (5), the direction adjustment is set so that the learning ratio $(s_{k+1}, g_k) = 1$ is fulfilled.

Using (5), (6), we derive the iterative process obtained earlier on the basis of heuristic considerations for solving problems of non-smooth optimization (r-algorithm [15]) in the form proposed in [16].

To obtain the formulas of the iterative process at iteration k , we transform the data (4) by subtracting adjacent equalities. We will obtain a new system:

$$(s, y_i) = 0, i = 0, 1, \dots, k-1, (s, g_k) = 1, y_i = g_i - g_{i+1}. \quad (7)$$

For a part of the data at $i = 0, 1, \dots, k-1$, we perform transformations (5), (6). Transformation (5) for $(s, y_i) = 0$ will take the following form:

$$s_{i+1} = s_i + \frac{H_i y_i (0 - (s_i, y_i))}{(y_i H_i y_i)}, s_0 = 0. \quad (8)$$

As a result of transformations (8), regardless of the values of the matrices H_i , by virtue of equality $s_0 = 0$ we obtain $s_i = 0, i = 0, 1, \dots, k-1$. Together with (8), we will carry out the transformation (6) sequentially:

$$H_{i+1} = H_i - (1 - \frac{1}{\alpha_k^2}) \frac{H_i y_i y_i^T H_i^T}{(y_i, H_i y_i)}, H_0 = I, i = 0, 1, \dots, k-1.$$

Transformation (6) for the last equality $(s, g_k) = 1$ from (7) can be omitted, because, as a result of transformation (5) for the last equality $(s, g_k) = 1$, a vector s_k collinear to the vector $H_k g_k$ will be obtained. Therefore, the iterative minimization process using transformations (5), (6) for data (7) at $k = 0, 1, 2, \dots$, will have the form (1), where the descent direction and the metric matrix are iterating using the following formulas:

$$s_k = H_k g_k, \quad (9)$$

$$H_{k+1} = H_k - (1 - \frac{1}{\alpha_k^2}) \frac{H_k y_k y_k^T H_k^T}{(y_k, H_k y_k)}, \quad H_0 = I, \quad y_k = g_k - g_{k+1}, \quad (10)$$

with fixed values of $\alpha_k^2 = \alpha^2 > 1$.

Thus, using the idea of obtaining the descent direction satisfying the system of inequalities (2), we derived the iterative process of the r-algorithm [15] in the form of [16], which was previously obtained based on heuristic considerations.

In Ref. [4], to speed up the process of finding the descent direction in method (5), (6), one of its special cases is presented. An algorithm for obtaining a direction s_k satisfying, in contrast to (5), simultaneously the last two learning relations $(s_{k+1}, g_k) = 1$ and $(s_{k+1}, g_{k-1}) = 1$ is proposed and theoretically justified. In this algorithm, the formulas for obtaining the direction of descent are as follows:

$$p_k = \begin{cases} g_k, & \text{if } (k \geq 1 \text{ and } (g_k, H_k g_{k-1}) > 0) \text{ or } k = 0, \\ g_k - \frac{g_{k-1}(g_k, H_k g_{k-1})}{g_{k-1}, H_k g_{k-1}}, & \text{otherwise.} \end{cases} \quad (11)$$

$$s_{k+1} = s_k + H_k p_k [1 - (s_k, g_k)] / (g_k, H_k p_k). \quad (12)$$

In this case, the metric matrix is adjusted according to formulas (10) with a varying space dilation parameter $\alpha_k^2 > 1$. Note that in order to ensure convergence when solving non-smooth problems, the choice of the descent direction is not always carried out according to Formulas (11) and (12).

Here and below, we will denote an algorithm by specifying the sequence of actions indicated by the corresponding equations. In Ref. [17], estimates of optimal parameter $\alpha_k^2 > 1$ values are obtained, and it is shown that the algorithm (1)–(11)–(12)–(10) in the quadratic case generates a sequence of conjugate descent vectors. In the case of smooth functions, it becomes possible to use in the algorithm (1)–(11)–(12)–(10) the conjugate gradient method instead of formulas for generating the descent direction (11), (12).

The transformation of the ILSM metric (10) in the minimization algorithm has the property of partially orthogonalizing the descent vectors to the directions of the adjacent gradients difference on the descent trajectory. In the case of minimizing the quadratic function, the use of transformations (10) to form the descent direction increases the degree of conjugacy.

In Ref. [18], estimates of the linear convergence rate for the algorithm (1)–(9)–(10) are obtained on strongly convex functions with a Lipschitz gradient. It is proven that on such functions, this method has accelerating properties qualitatively similar to those of Newton's method. The fact that the Newton method [18] and quasi-Newton methods have accelerating properties is explained by using either matrices of second derivatives or their approximations. Accelerating properties of the algorithm (1)–(9)–(10) due to the property of matrices (10) in the form of partial orthogonalization of the descent vectors to the directions of the adjacent gradients difference on the descent trajectory.

It was shown in [18] that the minimization algorithm (1) with the calculation of the descent direction based on (11), (12), and the transformation of matrices (10) is identical to the conjugate gradient method when minimizing quadratic functions. In this paper, the

Hestenes–Stiefel conjugate gradient method with metric matrices (10) was developed to solve problems of minimizing smooth functions. The finite convergence of such a method on quadratic functions is proved.

3. Properties and Convergence Rate of the Gradient Minimization Method with Incomplete Orthogonalization in the Direction of the Gradient Difference

Here, we analyze the properties of the algorithm (1)–(9)–(10). The algorithm consists of the following three steps:

1. Compute the next minimum point;

$$x_{k+1} = x_k - \gamma_k s_k, \quad \gamma_k = \arg \min_{\gamma} f(x_k - \gamma s_k), \quad k = 0, 1, 2, \dots$$

2. Compute descent direction $s_k = H_k g_k$;

3. Compute metric matrix update:

$$H_{k+1} = H_k - (1 - \frac{1}{\alpha_k^2}) \frac{H_k y_k y_k^T H_k^T}{(y_k, H_k y_k)}, \quad H_0 = I, \quad y_k = g_k - g_{k+1}.$$

Formulas for correction vector p_k , and metric matrix update H_{k+1} are as follows:

$$p_k = x_{k+1} = x_k - \gamma_k s_k, \quad s_k = H_k g_k, \quad \gamma_k = \arg \min_{\gamma} f(x_k - \gamma s_k), \quad k = 0, 1, 2, \dots, \quad (13)$$

$$H_{k+1} = H_k - (1 - \frac{1}{\alpha_k^2}) \frac{H_k y_k y_k^T H_k^T}{(y_k, H_k y_k)}, \quad \alpha_M \geq \alpha_k \geq \alpha_m > 1, \quad y_k = g_k - g_{k+1}, \quad (14)$$

where k is iteration number, x_k is current minimum point, s_k is descent direction, H_k is metric matrix at iteration k , g_k is gradient at iteration k , γ is stepsize, α_k is space dilation parameter, y_k is gradient difference, x_0 is initial point and H_0 is initial matrix.

As will be shown below, method (13)–(14) is invariant with respect to linear transformation of coordinates. By invariance, we mean the similarity of the process type in different systems. This means that when the iterations of the method are transferred to a new coordinate system, the iterative process is preserved. In this case, no conditions are imposed on the function, the only condition is its differentiability. For instance, in the case of a quadratic function, after transferring to a coordinate system where the Hessian has eigenvalues equal to one, the gradient differences $y_k = g_k - g_{k+1}$ will coincide with the method offsets $v_k = x_k - x_{k+1}$ during the iteration. Consequently, in the new coordinate system, the choice of the descent direction in (13) looks like the process of partial orthogonalization of the gradient to the previous descent directions using matrices (14). With complete orthogonalization, such a minimization process along the orthogonal system of directions for a quadratic function with unit Hessian will be finite [1].

The limiting variant of algorithm (13)–(14) for $\alpha_k^2 = \infty$ takes the form of the previously known conjugate gradient method [1]. In this method, the descent directions are orthogonalized to all previous differences in adjacent gradients. This property does not allow the method to optimize in a subspace defined by gradient differences, that is, it excludes it from the optimization process. Such a method will not be effective, especially for large dimensions. In algorithm (13)–(14), only a partial reduction in the components of the descent direction in the subspace of the preceding differences in adjacent gradients is performed. This does not block the possibility of minimization in the examined subspace. Transformation (14) suppresses to a greater extent the gradient components directed orthogonally to the ravine directions. This is one of the qualitative justifications explaining

the accelerating properties of the transformation (14), on the basis of which the r-algorithm was developed [15].

Cases of low efficiency of quasi-Newton methods are associated with functions in which the Hessian eigenvalues decrease as they approach the minimum. In this case, the inverse matrices of quasi-Newton methods increase in the examined subspace, which makes it difficult for the method to enter the unexplored subspace. In algorithms (13)–(14), by reducing the influence of the examined subspace, the unexplored subspace always has an advantage. For this reason, for some classes of functions, algorithm (13)–(14) may be more effective than quasi-Newton methods, which will be confirmed by numerical tests.

Transformation (14) in algorithm (13)–(14) can be understood as a scaling transformation for the gradient method. This transformation may be useful as a scaler in other methods. In this paper, we will consider the use of transformation (14) in the Hestenes–Stiefel conjugate gradient method.

We investigate the convergence rate of algorithm (13)–(14). Let us consider the conditions under which the convergence rate and accelerating properties of the r-algorithm are estimated.

Condition 1. We will assume that the function $f(x)$, $x \in R^n$ differentiable and strongly convex in R^n and is minimized, i.e., there exists $\rho > 0$ such that $\forall x, y \in R^n$ and the following inequality holds:

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) - \alpha(1 - \alpha)\rho \|x - y\|^2 / 2,$$

and its gradient $g(x) = \nabla f(x)$ satisfies the Lipschitz condition:

$$\|g(x) - g(y)\| \leq L\|x - y\| \quad \forall x, y \in R^n, \quad L > 0.$$

Here, L is Lipschitz parameter.

Denote by x^* the minimum point of function $f(x)$, $f^* = f(x^*)$, $f_k = f(x_k)$. The iteration of the gradient-matched method for exact one-dimensional descent has the form:

$$x_{k+1} = x_k - \beta_k s_k, \quad \beta_k = \underset{\beta \geq 0}{\operatorname{argmin}} f(x_k - \beta s_k), \quad (s_k, g_k) > 0, \quad k = 0, 1, 2, \dots, \quad (15)$$

where the initial point x_0 is set. Formula (15) does not specify the descent direction S , they only impose the condition $(s_k, g_k) > 0$ on it. Although the direction $s_k = H_k g_k$ from (13) also satisfies the condition from (1) due to the positive definiteness of the metric matrices.

The following theorem shows that changes in the gradient as a result of method (15) iterations with an exact one-dimensional descent lead to a decrease.

Theorem 1 [18]. Let the function satisfy Condition 1. Then, for the sequence f_k , $k = 0, 1, 2, \dots$ specified by (15), the following estimate takes place:

$$f_{k+1} - f^* \leq (f_0 - f^*) \exp \left[-\frac{\rho^2}{L^2} \sum_{i=0}^k \frac{\|y_i\|^2}{\|g_i\|^2} \right], \quad (16)$$

where $y_i = g_{i+1} - g_i$, f^* is optimal function value, f_0 is initial function value, L is Lipschitz parameter, ρ is defined in Condition 1.

Here we derive the recurrent formulas for converting the determinants and trace of the metric matrices under consideration. Denote $A_k = H_k^{-1}$. Denote by $Sp(A)$ a trace of matrix A and by $\det(A)$ a determinant of matrix A . For an arbitrary matrix $A > 0$, we denote by $A^{1/2}$ a symmetric matrix for which $A^{1/2} > 0$ and $A^{1/2} A^{1/2} = A$.

Lemma 1. Let $H_k > 0$ and matrix H_{k+1} is obtained as a result of the transformation (14):

$$H_{k+1} = H_k - (1 - \frac{1}{\alpha_k^2}) \frac{H_k y_k y_k^T H_k^T}{(y_k, H_k y_k)}, \quad \alpha_M \geq \alpha_k \geq \alpha_m > 1, \quad y_k = g_k - g_{k+1}.$$

Then, $H_{k+1} > 0$ and

$$A_{k+1} = A_k + (\alpha_k^2 - 1) \frac{y_k y_k^T}{(y_k, H_k y_k)}, \quad (17)$$

$$Sp A_{k+1} = Sp A_k + (\alpha_k^2 - 1) \frac{(y_k, y_k)}{(y_k, H_k y_k)}, \quad (18)$$

$$\det H_{k+1} = \det H_k / \alpha_k^2, \quad \det A_{k+1} = \alpha_k^2 \det A_k. \quad (19)$$

Proof of Lemma 1. We transform the right-hand side of expression (14) using the notation introduced earlier:

$$H_{k+1} = H_k^{1/2} \left[I - (1 - \frac{1}{\alpha_k^2}) \frac{H_k^{1/2} y_k y_k^T H_k^{1/2}}{(H_k^{1/2} y_k, H_k^{1/2} y_k)} \right] H_k^{1/2} = H_k^{1/2} \left[I - (1 - \frac{1}{\alpha_k^2}) \frac{v_k v_k^T}{(v_k, v_k)} \right] H_k^{1/2}, \quad v_k = H_k^{1/2} y_k \quad (20)$$

Inverting the matrix (20) element by element, we obtain (17). Formula (18) follows from (17). Calculating the determinant element by element for the matrices in (20), we obtain (19). $\square$

The following theorem substantiates the linear convergence rate of algorithm (13)–(14).

Theorem 2. Let the function $f(x)$ satisfy Condition 1. Then, for the sequence $\{f_k\}$, $k = 0, 1, 2, \dots$ given by the algorithm (13)–(14), with a limited initial matrix H_0 :

$$m_0 \leq (H_0 z, z) / (z, z) \leq M_0, \quad \forall z \in R^n, z \neq 0, \quad (21)$$

where M_0 and m_0 are maximal and minimal eigenvalues of the matrix H_0 , respectively, the following estimation takes place:

$$f_{k+1} - f^* \leq (f_0 - f^*) \exp \left\{ -\frac{\rho^2(k+1)}{L^2 n} \left[\frac{\ln(\alpha_m^2)}{(\alpha_M^2 - 1)} + \frac{n \ln(m_0/M_0)}{(k+1)(\alpha_M^2 - 1)} \right] \right\}. \quad (22)$$

Proof of Theorem 2. We transform (18):

$$Sp(A_{k+1}) = Sp(A_k) \left[1 + \frac{(\alpha_k^2 - 1)(y_k, y_k)}{Sp(A_k)(H_k y_k, y_k)} \right]. \quad (23)$$

Due to the exact one-dimensional descent in (13), the following condition is satisfied:

$$(s_k, g_{k+1}) = (H_k g_k, g_{k+1}) = 0.$$

The matrix H_k is positive definite, therefore, using the last equality, we obtain the following:

$$(H_k y_k, y_k) = (H_k g_k, g_k) + (H_k g_{k+1}, g_{k+1}) - 2(H_k g_k, g_{k+1}) \geq (H_k g_k, g_k).$$

Hence, taking into account the inequality

$$Sp(A_k) \geq M_k,$$

where M_k is the maximum eigenvalue of the matrix A_k , we find:

$$Sp(A_k)(H_k y_k, y_k) \geq Sp(A_k)(H_k g_k, g_k) \geq \frac{Sp(A_k)}{M_k}(g_k, g_k) \geq (g_k, g_k).$$

The last estimate allows us to reduce inequality (23) to the following form:

$$Sp(A_{k+1}) \leq Sp(A_k) \left[1 + (\alpha_k^2 - 1) \frac{\|y_k\|^2}{\|g(x_k)\|^2} \right]. \quad (24)$$

Based on the ratio between the arithmetic mean and geometric mean of the eigenvalues of the matrix $A > 0$, we have

$$Sp(A)/n \geq [\det(A)]^{1/n}.$$

Using the last inequality in (24), taking into account (19), we obtain:

$$\frac{Sp(A_0)}{n} \prod_{i=0}^k \left[1 + (\alpha_M^2 - 1) \frac{\|y_i\|^2}{\|g(x_i)\|^2} \right] \geq \frac{Sp(A_{k+1})}{n} \geq (\det(A_{k+1}))^{1/n} = [(\alpha_m^2)^{k+1} \det(A_0)]^{1/n}.$$

The latter ratio, based on inequality

$$1 + p \leq \exp(p),$$

we convert to the following form:

$$\frac{Sp(A_0)}{n} \exp \left[(\alpha_M^2 - 1) \sum_{i=0}^k \frac{\|y_i\|^2}{\|g(x_i)\|^2} \right] \geq (\alpha_m^2)^{(k+1)/n} (\det(A_0))^{1/n}. \quad (25)$$

By virtue of condition (21), inequalities hold:

$$Sp(A_0)/n \leq 1/m_0, (\det(A_0))^{1/n} \geq 1/M_0.$$

Taking these relations into account, we transform inequality (25):

$$\frac{1}{m_0} \exp \left[(\alpha_M^2 - 1) \sum_{i=0}^k \frac{\|y_i\|^2}{\|g(x_i)\|^2} \right] \geq \frac{\alpha_m^{2(k+1)/n}}{M_0}.$$

Taking the logarithm of the latter, we find:

$$\left[(\alpha_M^2 - 1) \sum_{i=0}^k \frac{\|y_i\|^2}{\|g(x_i)\|^2} \right] \geq \frac{(k+1) \ln(\alpha_m^2)}{n} + \ln\left(\frac{1}{M_0}\right) - \ln\left(\frac{1}{m_0}\right).$$

From this follows the inequality:

$$\sum_{i=0}^k \frac{\|y_i\|^2}{\|g(x_i)\|^2} \geq \frac{(k+1) \ln(\alpha_m^2)}{n(\alpha_M^2 - 1)} + \frac{\ln(m_0/M_0)}{(\alpha_M^2 - 1)},$$

which, together with estimate (16) of Theorem 1, proves (22). $\square$

The obtained convergence rate estimates do not explain the method's high convergence rate, for example, on quadratic functions. To justify the method's acceleration properties, we have to demonstrate its invariance under a linear coordinate transformation and then use estimate (22) in the coordinate system in which the ρ/L ratio is maximal. It is possible

to increase this ratio, for example, in the case of quadratic functions, where its maximum value is 1.

4. Acceleration Properties of the Space Dilation Algorithm in the Direction of the Gradients Difference

To estimate the convergence rate greater than linear in minimization methods, the minor variability of the Hessian in the extremum region is primarily used, which can be estimated based on the Hessian properties. We examine the acceleration properties of the algorithm without assumptions about the function's second derivative matrices.

According to estimate (22), the convergence rate of the r-algorithm is determined by the magnitude of the ρ/L ratio. In our study, similar to study [18], we will assume the existence of a linear transformation of coordinates that increases the magnitude of this ratio.

Let a function $f(x)$ satisfy Condition 1. We define a linear transformation of variables

$$\hat{x} = Px, \quad (26)$$

where $P \in R^n$ is a non-singular matrix. In the new coordinate system, the function being minimized will have the following form:

$$f_P(\hat{x}) = f(P^{-1}\hat{x}) = f(x). \quad (27)$$

The function (27) formed in this way also satisfies Condition 1 with parameter of strong convexity ρ and Lipschitz parameter L .

Denote $V \in R^{n \times n}$ a non-singular matrix such that for parameters ρ and L of functions $f_V(\hat{x})$ with

$$\hat{x} = Vx \quad (28)$$

and $f_P(\hat{x})$ for an arbitrary non-singular matrix $P \in R^{n \times n}$ in (26), the inequality holds

$$\frac{\rho_V}{L_V} \geq \frac{\rho_P}{L_P}. \quad (29)$$

As was established in [18], transformation (28) plays the role of a distinguished coordinate system, which is the best from the point of view of the convergence rate of gradient methods. For a non-degenerate quadratic function, we can consider a coordinate system where $\rho/L = 1$ and the maximum and minimum eigenvalues are equal. In the general case, we only assume the possibility of eliminating the differences in the elongation of the level surfaces.

Next, we will show that algorithm (13)–(14), applied to the function $f(x)$, and algorithm (13)–(14), applied to the function $f_P(\hat{x})$ defined in (22), under appropriate initial conditions, construct sequences of minimization points related by the transformation (26). This will allow us to use the estimates of the convergence rate of algorithm (13)–(14) in the preferred coordinate system.

Theorem 3. Let the initial conditions of the algorithm (13)–(14), applied to minimize the functions $f(x)$ and $f_P(\hat{x})$, defined in (27), be related by the equalities:

$$\hat{x}_0 = Px_0, \quad \hat{H}_0 = PH_0P^T. \quad (30)$$

Then the characteristics of these processes are related by the following relations:

$$f_P(\hat{x}_k) = f(x_k), \quad \hat{x}_k = Px_k, \quad \nabla f_P(\hat{x}_k) = P^{-T} \nabla f(x_k), \quad \hat{H}_k = PH_kP^T, \quad (k = 0, 1, 2, \dots). \quad (31)$$

Proof of Theorem 3. For the gradients of functions $f_P(\hat{x})$ and $f(x)$ the following relation holds:

$$\nabla f_P(\hat{x}) = P^{-T} \nabla f(x).$$

From this and (30) follows (31) for $k = 0$. Let us assume that equalities (31) are satisfied $\forall k = 0, 1, \dots, i$. Let us show their feasibility for $k = i + 1$. From (13) for $k = i$ after left-hand multiplying by P , taking into account the proved equalities (31), we obtain:

$$Px_{i+1} = Px_i - \gamma_i PH_i P^T P^{-T} \nabla f(x_i) = \bar{x}_i - \gamma_i \hat{H}_i \nabla f_P(\hat{x}_i). \quad (32)$$

Hence, according to the definition of the function f_P , at the stage of one-dimensional minimization the equality $\gamma_i = \bar{\gamma}_i$ is satisfied. Therefore, the right-hand side of (32) is the implementation of the step in the new coordinate system. Consequently:

$$\hat{x}_i = Px_i, \nabla f_P(\hat{x}_i) = P^{-T} \nabla f(x_i), \hat{y}_i = \nabla f_P(\hat{x}_{i+1}) - \nabla f_P(\hat{x}_i) = P^{-T} y_i. \quad (33)$$

Multiplying (14) with the current indices on the left by P , and on the right by P^T , taking into account (33) and the equality:

$$(y_i, P^{-1} PH_i P^T P^{-T} y_i) = (\hat{H}_i \hat{y}_i, \hat{y}_i),$$

we get

$$PH_{i+1} P^T = PH_i P^T - \left(1 - \frac{1}{\alpha_i^2}\right) \frac{PH_i P^T P^{-T} y_i y_i^T P^{-1} PH_i P^T}{(y_i, P^{-1} PH_i P^T P^{-T} y_i)} = \hat{H}_i - \left(1 - \frac{1}{\alpha_i^2}\right) \frac{\hat{H}_i \hat{y}_i \hat{y}_i^T \hat{H}_i^T}{(\hat{H}_i \hat{y}_i, \hat{y}_i)},$$

where the right side is the implementation of (14) in the new coordinate system.

Finally, we get $PH_{i+1} P^T = \hat{H}_{i+1}$. Therefore, equalities (31) will also be valid for $k = i + 1$. Continuing the process of induction, we obtain a proof of the theorem. $\square$

In the following theorem, we use algorithms (13)–(14) in a distinguished coordinate system (28) with property (29).

Theorem 4. Let the function $f(x)$ satisfy Condition 1. Then, for the sequence $\{f_k\}$, $k = 0, 1, 2, \dots$ given by the algorithm (13)–(14), with a limited initial matrix H_0 , according to (21) the following estimation takes place:

$$f_{k+1} - f^* \leq (f_0 - f^*) \exp \left\{ -\frac{\rho_V^2(k+1)}{L_V^2 n} \left[\frac{\ln(\alpha_m^2)}{(\alpha_M^2 - 1)} + \frac{n \ln(\hat{m}_0 / \hat{M}_0)}{(k+1)(\alpha_M^2 - 1)} \right] \right\}. \quad (34)$$

where $\hat{m}_0, \hat{M}_0$ are the minimum and maximum eigenvalues of the matrix $\hat{H}_0 = V H_0 V^T$, respectively, in the selected coordinate system (28) having the property (29).

Proof of Theorem 4. According to the results of Theorem 3, we can choose an arbitrary coordinate system to estimate the convergence rate of the minimization process in the algorithm. Therefore, using estimate (22) in a coordinate system with matrix $P = V$, we obtain estimates (34). $\square$

The first term in square brackets (34) characterizes the constant in the method's convergence rate estimate, and the second term represents the cost of adjusting the metric matrix.

Let us consider the acceleration effect of algorithm (13)–(14) compared to well-known methods: steepest descent and Newton’s method. For these methods, the convergence rate estimate is as follows:

$$f_{k+1} - f^* \leq q^{k+1}(f_0 - f^*). \quad (35)$$

Under Condition 1 imposed on the function, for the steepest descent, where $s_k = g_k$, the decay rate is as follows [1]:

$$q = 1 - \frac{\rho}{L}. \quad (36)$$

For Newton’s method $s_k = [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$ and decay rate is [18]:

$$q = 1 - \frac{\rho^2}{L^2}. \quad (37)$$

The convergence rate for Newton’s method, due to its invariance with respect to the linear transformation of coordinates, has the form [18]:

$$q = 1 - \rho_V^2 / L_V^2. \quad (38)$$

Given that

$$l_V / L_V \gg l / L, \quad (39)$$

Taking into account (36)–(38), the convergence rate of Newton’s method will be significantly higher than that of the steepest descent.

Estimate (34) for the algorithm (13)–(14) is equivalent to the estimate for Newton’s method (37) in terms of the influence of constants on the convergence rate. Under condition (39), the convergence rate estimate for the algorithm (13)–(14) is preferable to that for the steepest descent method, which is confirmed further by a computational experiment.

Thus, algorithm (13)–(14) on strongly convex functions, without assuming the existence of second derivatives, exhibits acceleration properties compared to the steepest descent method.

5. Hestenes–Stiefel Method in a Metric with Incomplete Orthogonalization in the Direction of Gradient Difference

Transformation (14) in the algorithm (13)–(14) can act as a scaling transformation for the steepest descent. This transformation may also be useful in other methods.

In Ref. [17], it is shown that the sequence of approximations of the minimum generated by the algorithm (1)–(11)–(12), with the transformation of matrices (10), or equivalently with (14), coincides with the sequence generated by the conjugate gradient method. Transformations (11), (12) are based on the simultaneous solution of the system of equalities $(s_{k+1}, g_k) = 1$ and $(s_{k+1}, g_{k-1}) = 1$. The equality $(s_{k+1}, g_k) - (s_{k+1}, g_{k-1}) = (s_{k+1}, g_k - g_{k-1}) = 0$ holds for their difference. In the Hestenes–Stiefel method [30], the new direction is also chosen based on the equality $(s_{k+1}, g_k - g_{k-1}) = 0$. In the quadratic case, the new direction in the Hestenes–Stiefel method will be conjugate to the current descent direction. In the case of smooth functions, we can replace the transformations (11), (12) for obtaining the descent direction with the transformations in the Hestenes–Stiefel method.

The Hestenes–Stiefel conjugate gradient method [34] has the form:

$$x_{k+1} = x_k - \gamma_k s_k, \quad \gamma_k = \underset{\gamma}{\operatorname{argmin}} f(x_k - \gamma s_k), \quad k = 0, 1, 2, \dots, \quad (40)$$

$$s_0 = g_0, s_{k+1} = g_{k+1} - \frac{(g_{k+1}, y_k)}{(s_k, y_k)} s_k, y_k = g_k - g_{k+1}. \quad (41)$$

From (41), it follows that the new descent direction of the current gradient difference is orthogonal $(s_{k+1}, y_k) = 0$. For a quadratic function, this implies conjugacy of adjacent descent vectors, regardless of the accuracy of the one-dimensional search.

In the Hestenes–Stiefel method with a space metric transformation (14), we also use the property $(s_{k+1}, y_k) = 0$ to obtain the descent vector transformation formulas:

$$x_{k+1} = x_k - \gamma_k s_k, \gamma_k = \underset{\gamma}{\operatorname{argmin}} f(x_k - \gamma s_k), k = 0, 1, 2, \dots, \quad (42)$$

$$s_0 = H_0 g_0, s_{k+1} = H_k g_{k+1} - \frac{(H_k g_{k+1}, y_k)}{(s_k, y_k)} s_k, y_k = g_{k+1} - g_k, \quad (43)$$

$$H_0 = I, H_{k+1} = H_k - (1 - \frac{1}{\alpha_k^2}) \frac{H_k y_k y_k^T H_k^T}{(y_k, H_k y_k)}, \alpha_M \geq \alpha_k \geq \alpha_m > 1. \quad (44)$$

Algorithms (42)–(44) has the properties of the conjugate gradient method. To justify this property, we need the following theorem.

Theorem 5. Let the matrices $H_i, i = 1, 2, \dots, k$, be obtained as a result of transformations (44), and let the vectors $y_i, i = 0, 1, \dots, k-1$, used in (44), be orthogonal to the vector g_{k+1} . Then, $H_k g_{k+1} = g_{k+1}$.

Proof of Theorem 5. We will carry out the proof by induction. Let $H_i g_{k+1} = g_{k+1}$ for $i < k-1$. Then, taking into account the orthogonality of the vectors $y_i, i = 0, 1, \dots, k-1$, to the vector g_{k+1} , we obtain

$$(H_{i+1}, g_{k+1}) = \left(H_i - (1 - \frac{1}{\alpha_k^2}) \frac{H_i y_i y_i^T H_i^T}{(y_i, H_i y_i)}, g_{k+1} \right) = g_{k+1} - (1 - \frac{1}{\alpha_k^2}) \frac{(y_i, H_i g_{k+1})}{(y_i, H_i y_i)} H_i y_i = g_{k+1} - (1 - \frac{1}{\alpha_k^2}) \frac{(y_i, g_{k+1})}{(y_i, H_i y_i)} H_i y_i = g_{k+1}.$$

Continuing the process of induction, we obtain the statement of the theorem. $\square$

Regarding the convergence of algorithms (42)–(44) on quadratic functions, the following theorem holds.

Theorem 6. When minimizing quadratic functions, the sequences of points $x_k, k = 0, 1, 2, \dots$, of algorithms (40)–(41), (42)–(44) with the same initial points and initial matrix $H_0 = I$ coincide.

Proof of Theorem 6. We will prove the theorem by induction. Let the points x_k of algorithms (40)–(41), (42)–(44) coincide for $i < k$. In conjugate gradient methods, when minimizing a quadratic function using exact one-dimensional descent, the gradient vectors are mutually orthogonal. Therefore, the current gradient g_i at a new point is the same for both algorithms and orthogonal to all previous gradients. The vector s_i is obtained by formula (43) using the matrix $H_{i-1} g_i$. The matrix H_{i-1} is obtained using vectors $y_j, j = 0, 1, \dots, i-2$, in which the gradient vectors $g_j, j = 0, 1, \dots, i-1$, orthogonal to the vector g_i , are used. From here, according to Theorem 5, we obtain $H_{i-1} g_i = g_i$. Using this equality in (43), we obtain the coincidence of the descent direction with the Hestenes–Stiefel method

$$s_i = H_{i-1} g_i - \frac{(H_{i-1} g_i, y_{i-1})}{(s_{i-1}, y_{i-1})} s_{i-1} = g_i - \frac{(g_i, y_{i-1})}{(s_{i-1}, y_{i-1})} s_{i-1}.$$

Consequently, the new iteration of each process will be performed by minimization along the same directions and from the same point. Continuing the induction process further, we obtain a proof of the theorem. $\square$

The level surface topology of the function being minimized may be similar in many ways to the topology of the quadratic function in a neighborhood significantly larger than the neighborhood in which the function and its quadratic representation are similar. In such a case, it can be assumed that algorithms (13)–(14) and (42)–(44), which do not use approximations of the Hessians, will be more effective than quasi-Newton methods.

6. Numerical Experiment

To identify the potential of transformation (14) for solving problems of minimizing smooth functions and to test the effectiveness of the presented algorithms (13)–(14) and (42)–(44), a numerical study was conducted. The methods are compared with the quasi-Newton BFGS method.

All methods used in the study utilized the one-dimensional search described in [35], where the gradient and function value are used as information for organizing the method. This is particularly advantageous in conditions where the costs of computing the gradient and function are comparable.

In the following tables we will denote:

- (1) BFGS—quasi-Newtonian BFGS method;
- (2) HY_g—algorithm (13)–(14);
- (3) HY_XS—algorithm (42)–(44).

When solving an ill-conditioned minimization problem with high accuracy, the quasi-Newton method BFGS is typically used, or, if possible, Newton's method. For this reason, we chose the BFGS method as the benchmark for comparison with the methods under study.

The set of test functions includes a quadratic function. Since we know the function's condition number, we can estimate its complexity for minimization. Also, based on the results of minimizing the quadratic function, we obtain information about the method's behavior in a certain neighborhood of the current minimization point of the real function, where its quadratic representation is valid.

The tests include functions with both linear and curvilinear ravines. With curvilinear ravines, the Hessian eigenvectors change as we move toward the minimum, which leads to obsolescence of the metric matrices and a decrease in the convergence rate. Further successful progress requires reconfiguring the method's parameters. On such functions we will observe the behavior of the method with the variability of the quadratic representation of the function as we move towards the minimum.

We also consider a function whose level surface topology matches that of the quadratic function. In this case, it is of interest to compare the algorithms (13)–(14) and (42)–(44) with quasi-Newton methods, which rely less on the properties of the Hessian matrices.

The final test problem reflects the variability of scaling across variables as we move towards the extremum. In this case, there is an active change in the level surface topology due to changes in the scales along the coordinate axes.

In all methods, the function and gradient were calculated simultaneously. Tables show the number of iterations and the total number of function and gradient calculations for the selected methods. The problem dimension in each experiment varies from 100 to 1000. The stopping criterion was $f(x^k) - f^* \leq \varepsilon$.

6.1. Quadratic Function

We use the following quadratic function:

$$f_Q(x, [amax]) = \frac{1}{2} \sum_{i=1}^n a_i x_i^2, a_i = amax^{\frac{i-1}{n-1}}. \quad (45)$$

The limits of eigenvalues a_i of this function are $\lambda_{\min} = 1$ and $\lambda_{\max} = a_{\max}$. Starting point is $x_0 = (100, 100, \dots, 100)$. The stopping criterion was $f(x^k) - f^* \leq 10^{-10}$.

Tables 1 and 2 show the results of function minimization for different degrees of conditionality $a_{\max} = 10^4$ and $a_{\max} = 10^8$. Best results are given in bold.

Table 1. Results of function $f_Q(x, [a_{\max} = 10^4])$ minimization.

n	BFGS		HY_g		HY_XS	
	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations
100	107	324	197	463	166	408
200	176	464	304	666	203	472
300	234	575	404	865	254	573
400	284	672	494	1045	292	650
500	328	760	577	1211	327	720
600	367	837	656	1370	360	787
700	403	909	728	1515	392	851
800	437	978	795	1649	424	915
900	466	1036	856	1771	454	975
1000	495	1093	912	1884	482	1032

Table 2. Results of function $f_Q(x, [a_{\max} = 10^8])$ minimization.

n	BFGS		HY_g		HY_XS	
	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations
100	136	443	474	1111	476	1155
200	263	785	725	1668	651	1559
300	377	1086	967	2219	796	1838
400	481	1342	1217	2745	934	2118
500	576	1547	1455	3303	1080	2429
600	675	1783	1695	3863	1253	2818
700	769	2018	1932	4409	1414	3169
800	853	2198	2164	4942	1551	3484
900	944	2428	2389	5432	1686	3778
1000	1026	2608	2620	5996	1813	4067

For function with a low degree of conditionality the BFGS quasi-Newton method does not have enough number of iterations to create a suitable metric matrix. Perhaps for this reason, the HY_XS algorithm is equivalent in performance to the BFGS. On this test, HY_XS method outperforms the HY_g method.

In Table 2, the problem conditionality is higher, and the required number of iterations to solve the problem is commensurate with the problem dimension. BFGS manages to construct the metric matrix and, as a result, achieves a high convergence rate in the final iterations and outperforms other algorithms. Here, HY_XS outperforms HY_g.

This is an example of an ill-conditioned problem with a fixed Hessian, where the quasi-Newton method (BFGS) is significantly more efficient than other methods. Based on the results of this example, we can conclude that if the matrix of second derivatives in a real-world problem is ill-conditioned and does not change significantly over a sufficiently wide range, then the quasi-Newton method will outperform the HY_g and HY_XS algorithms.

6.2. Function with Multidimensional Ellipsoidal Ravine

The following function has a multidimensional ellipsoidal ravine. Minimization occurs when moving along a curvilinear ravine to the minimum point.

$$f_E(x, [amax, bmax]) = (1 - x_1)^2 + a \max \left(1 - \sum_{i=1}^n x_i^2 / b_i \right)^2, b_i = bmax^{\frac{i-1}{n-1}} \quad (46)$$

The stopping criterion was $f(x^k) - f^* \leq 10^{-4}$.

Tables 3 and 4 demonstrate results of function f_E minimization. In Table 3 starting point is $x_0^1 = (-1, 0.1, \dots, 0.1)$, in Table 4 starting point is $x_0^2 = (-1, 2, 3, \dots, n)$. Best results are given in bold.

Table 3. Results of function $f_E(x, [a_{max} = 10^2, b_{max} = 10^3])$ minimization from starting point x_0^1 .

<i>n</i>	BFGS		HY_g		HY_XS	
	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations
100	58	196	354	1008	46	125
200	114	372	682	1837	58	150
300	135	463	949	2495	37	118
400	154	485	1052	2691	37	95
500	154	496	1162	2991	49	132
600	185	599	1017	2524	39	118
700	185	573	917	2200	39	104
800	207	675	1168	2858	42	117
900	224	739	490	1171	40	110
1000	236	750	1042	2457	36	104

Table 4. Results of function $f_E(x, [a_{max} = 10^2, b_{max} = 10^3])$ minimization from starting point x_0^2 .

<i>n</i>	BFGS		HY_g		HY_XS	
	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations
100	937	2465	242	689	74	288
200	1434	3729	364	1008	139	507
300	1729	4526	603	1575	146	513
400	1929	5042	661	1661	258	857
500	2127	5590	819	2019	180	602
600	2267	5945	840	2034	204	679
700	2356	6169	1017	2454	261	852
800	2428	6381	1160	2769	271	924
900	2532	6676	1355	3249	207	697
1000	2631	6932	1540	3697	248	817

For this starting point, the BFGS and HY_XS methods reach a minimum almost immediately, where the conditioning level increases as the minimum is approached. Therefore, the BFGS and HY_XS algorithms are more efficient than the HY_g method.

At the minimum point, the function is degenerate. Therefore, as the degree of conditioning increases while the minimum is approached, the HY_g method, which does not use conjugate directions, turns out to be the worst performer.

For this starting point, the methods initially enter a ravine far from the minimum point and move along the bottom of the curvilinear ravine. The elongation of the isosurfaces

and the directions of elongation in the curvilinear ravine change, preventing the BFGS method from effectively and quickly adjusting the metric matrix. Here, the HY_XS and HY_g algorithms have an advantage. Moreover, the conjugacy of the descent directions in the HY_XS algorithm allows for obtaining a solution much faster in the neighborhood of a degenerate minimum point.

6.3. Function with Multidimensional Ellipsoidal Ravine and Non-Degenerate Minimum Point

The next function also has a multi-dimensional ellipsoidal ravine.

$$f_{EX}(x, [amax, bmax]) = (1 - x_1)^2 + a \max \left(1 - \sum_{i=1}^n \frac{x_i^2}{b_i} \right)^2 + \frac{1}{2} \sum_{i=1}^n \frac{x_i^2}{b_i}, \quad b_i = b \max^{\frac{i-1}{n-1}}. \quad (47)$$

The starting point is $x_0 = (-1, 2, 3, \dots, n)$. Stopping criterion is $f(x^k) - f^* \leq 10^{-10}$. The function has an additional quadratic term, so the function ceases to be degenerate at the minimum point. Due to this, gradient methods are able to find the minimum of function f_{EX} with higher accuracy than for function f_E . Table 5 shows the results of function f_{EX} minimizing. Best results are given in bold.

Table 5. Results of function $f_{EX}(x, [a_{max} = 10^2, b_{max} = 10^3])$ minimization.

<i>n</i>	BFGS		HY_g		HY_XS	
	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations
100	862	2198	281	817	295	1005
200	1354	3436	433	1097	349	1180
300	1704	4356	617	1438	490	1257
400	1913	4909	816	1848	623	1561
500	2167	5581	879	1946	454	1339
600	2315	5967	992	2173	649	1572
700	2471	6397	1154	2511	563	1540
800	2576	6683	1333	2888	679	1724
900	2683	6969	1441	3105	802	1934
1000	2765	7184	1601	3441	804	1951

Despite the fact that the minimum point is non-degenerate, BFGS performs worse due to its slow progress along the ravine bottom. The HY_g algorithm spends significant time reaching the minimum in the neighborhood of the minimum. In the HY_XS algorithm, the minimization stage in the minimum region is completed more quickly by using conjugate directions. Here, HY_XS, which combines the ability to more efficiently progress along the bottom of a curvilinear ravine and the conjugacy of descent vectors, proves more effective than other algorithms.

6.4. Non-Quadratic Function

The following function has topological similarity of level surfaces to the quadratic function.

$$f_{Q2}(x, [amax]) = \left(\sum_{i=1}^n a_i x_i^2 \right)^2, \quad a_i = a \max^{\frac{i-1}{n-1}}. \quad (48)$$

Starting point is $x_0 = (1, 1, \dots, 1)$. Stopping criterion is $f(x^k) - f^* \leq 10^{-10}$. This is equivalent to reducing the quadratic term $\sum_{i=1}^n a_i x_i^2$ to a value smaller than 10^{-5} . Table 6 demonstrates the results of function f_{Q2} minimization. Best results are given in bold.

Table 6. Results of function $f_Q^2(x, [a_{max} = 10^4])$ minimization.

<i>n</i>	<i>BFGS</i>		<i>HY_g</i>		<i>HY_XS</i>	
	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations
100	1028	2639	141	381	171	544
200	1848	4787	224	565	201	599
300	2511	6553	297	701	234	646
400	3056	7982	359	814	337	696
500	3527	9232	420	933	346	714
600	3973	10,429	474	1040	355	732
700	4335	11,378	520	1120	365	752
800	4644	12,204	567	1209	378	778
900	4951	13,034	611	1286	390	802
1000	5204	13,697	653	1354	402	826

Here, a low convergence rate of the BFGS method is observed. As the method approaches a minimum, the Hessian elements tend to zero. Consequently, the elements of the inverse matrix increase. In Ref. [17], it was noted that as the matrix of second derivatives in the surveyed space grows, the approximated matrices in quasi-Newton methods also grow. This complicates the exit to the unexplored subspace, which slows the convergence rate. In the HY_g and HY_XS algorithms, on the contrary, the possibility of entering the unexplored subspace always increases, which explains the advantage of these algorithms for this function. The results for the HY_XS algorithm confirm the advantages of using the metric in the conjugate gradient method.

6.5. Function with Scaling by Variables

This function used additional variables c_i to change the scales a_i for each variable.

$$f_{abc}(x, [amax, bmax]) = \frac{1}{2} \sum_{i=1}^n a_i c_i x_i^2, \quad a_i = a \max^{i-1}, \quad c_i = \frac{b \max}{b_i} \left(\frac{x_i^2}{1 + x_i^2} \right) + b_i \left(1 - \frac{x_i^2}{1 + x_i^2} \right), \quad b_i = b \max^{i-1}$$

This function near the extremum will have the form

$$f_{abc}(x, [amax, b \max]) \approx \frac{1}{2} \sum_{i=1}^n a_i b_i x_i^2.$$

Far from the extremum, we obtain a function in which the coefficients are used in reverse order:

$$f_{abc}(x, [amax, b \max]) \approx \frac{1}{2} \sum_{i=1}^n a_i \frac{b \max}{b_i} x_i^2.$$

Changes in the scales of the coefficients according to the presented limiting variants of the function can be represented by the following transition:

$$a_i \frac{b \max}{b_i} \rightarrow a_i b_i.$$

It follows that the first coefficient decreases by a factor of $b \max$, and the last coefficient increases by a factor of $b \max$. The starting point is $x_0 = (100, 100, \dots, 100)$. The stopping criterion is $f(x^k) - f^* \leq 10^{-10}$. Table 7 demonstrates the results of function f_{abc} minimization. Best results are given in bold.

Table 7. Results of function $f_{abc}(x, [a_{max} = 10^4, b_{max} = 10^3])$ minimization.

<i>n</i>	<i>BFGS</i>		<i>HY_g</i>		<i>HY_XS</i>	
	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations
100	1819	4735	502	1407	587	1635
200	2992	7450	716	2033	763	2016
300	4004	10,030	922	2430	956	2498
400	4483	11,193	1154	2987	1124	2856
500	4961	12,371	1431	3651	1262	3205
600	5263	13,130	1397	3862	1422	3590
700	5577	13,949	1554	4241	1526	3842
800	5850	14,737	1858	4760	1677	4189
900	6026	15,107	1917	4949	1802	4489
1000	6154	15,439	2124	5376	1914	4783

The BFGS method performs the worst on this function. This is due to the high variability of the Hessian. Here, the space scanning strategy with excludes previously explored subspaces, which is used in the HY_g and HY_XS algorithms, proves suitable. The use of vector conjugacy in the HY_XS algorithm has little effect on the algorithm's performance compared to the HY_g method.

It should be noted that the metrics in QNM and in RSM have different meanings. There are obstacles to good convergence in QNM: the first is a rapid change in the directions of the Hessian eigenvectors, which is observed in tests, Tables 2 and 3; the second is a rapid change in the scales of variables (Tables 6 and 7). Another case of poor convergence is the tendency of the second derivatives to zero as they approach the minimum (Table 5 illustrates this case).

The following Table 8 summarizes the results of minimization for all functions with $n = 1000$.

Table 8. Results of function minimization, $n = 1000$.

<i>Function</i>	<i>BFGS</i>		<i>HY_g</i>		<i>HY_XS</i>	
	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations	Number of Iterations	Number of Function and Gradient Calculations
$f_Q, a_{max} = 10^4$	495	1093	912	1884	482	1032
$f_Q, a_{max} = 10^8$	1026	2608	2620	5996	1813	4067
f_E, x_0^1	236	750	1042	2457	36	104
f_E, x_0^2	2631	6932	1540	3697	248	817
f_{EX}	2765	7184	1601	3441	804	1951
f_{Q^2}	5204	13,697	653	1354	402	826
f_{abc}	6154	15,439	2124	5376	1914	4783

Let us draw the main conclusions based on the results of Table 8.

1. The BFGS method significantly outperforms the HY_g and HY_XS algorithms in ill-posed problems, where the Hessian does not undergo significant changes depending on the position in the minimization space. This thesis is confirmed by the data for the function f_Q with $a_{max} = 10^8$. For such a function, BFGS manages to construct a suitable coordinate transformation, after which the algorithm's convergence rate significantly increases. In such situations, the HY_XS method has advantages over the HY_g algorithm due to the conjugate gradient method.

2. Function f_E has an ellipsoidal ravine and a degenerate minimum point. From the first starting point, with a steeper ravine than from the second starting point, the main obstacle is the degeneracy of the minimum. Algorithms BFGS and HY_XS, which use the properties of a quadratic function, complete the minimization stage in the extremum region significantly faster than HY_g. As the degeneracy of the ravine increases, BFGS's progress along the ravine becomes slow due to the high variability of the Hessian. Algorithms HY_g and HY_XS are more successful here, and the properties of the conjugate gradient method of HY_XS lead to a significant acceleration of the convergence rate. A similar situation arises when minimizing the function f_{EX} .
3. The level surfaces of the function f_{Q^2} are typologically equivalent to the level surfaces of a quadratic function. In this situation, based on an analysis of the data in Table 7, we can conclude that the metric transformation in algorithms HY_g and HY_XS is more effective than the metric transformation of the quasi-Newton method.
4. It can be seen that the overheads of the HY_g method are reduced by approximately 2 times compared to the quasi-Newton method, and the overheads of the HY_XS method were 3.5 times lower on average than those of the quasi-Newton method. Overall, the computational experiment confirms the acceleration properties of the algorithms HY_g and HY_XS, obtained by using the metric transformation (14), and their applicability alongside quasi-Newton methods in solving complex, ill-conditioned minimization problems.

7. Conclusions

The conditionality of the minimization problem determines the spread of the isosurface elongations in different directions, which in turn determines the complexity of the problem's solution. In minimization practice, it is often possible to reduce the isosurface elongations through a linear coordinate transformation, thereby increasing the convergence rate of the gradient method used in the new coordinate system. For strongly convex functions with a Lipschitz gradient, such a coordinate transformation reduces the difference between the strong convexity and Lipschitz constants. If the function is twice differentiable, such a transformation reduces the spread of the Hessian's eigenvalues.

The noted linear transformation of coordinates eliminates the linear background that worsens the convergence of the gradient method. The property of eliminating such background is possessed by Newton's method, quasi-Newton methods, and subgradient methods with a change in the space metric. In this work, we proved that algorithm HY_g, where the gradient method acquires accelerating abilities due to the use of the metric transformation (14), has similar properties. The qualitative nature of the convergence rate estimates for algorithm HY_g and Newton's method coincide.

Taking into account the efficiency of transformation (14), we propose the conjugate gradient method HY_XS, also based on this transformation. The equivalence of this method to the conjugate gradient method on quadratic functions is proven. To evaluate the efficiency of the proposed method on test functions, we compare it with the quasi-Newton BFGS and the algorithm HY_g.

The test results demonstrate the efficiency of algorithm HY_XS. On almost all test problems, its convergence rate is significantly higher than that of the HY_g method. A comparison with the quasi-Newton method shows that method HY_XS is more efficient in the case of a high degree of variability of the Hessian matrices of the function being minimized.

A computational experiment shows that methods HY_g and HY_XS are effective in solving ill-conditioned problems with complex curvilinear ravines and unstable characteristics of the second derivatives. The main conclusion is that these methods, along with

quasi-Newton methods, are applicable to solving problems of minimizing smooth functions with a high degree of conditioning.

The present study was conducted on smooth functions. It is probably of interest to extend these metric transformations to increase the convergence rate of the method on non-smooth functions. It is clear, purely speculatively, that by converting the metric, the elongation of the level surfaces in different directions can be equalized in the non-smooth case. However, we retain a detailed study of this issue for future research.

Author Contributions: Conceptualization, V.K.; methodology, V.K., E.T. and S.G.; software, V.K.; validation, L.K., E.T. and S.G.; formal analysis, L.K. and S.G.; investigation, V.K.; resources, I.R. and L.K.; data curation, S.G.; writing—original draft preparation, V.K. and S.G.; writing—review and editing, E.T. and L.K.; visualization, V.K. and E.T.; supervision, V.K., I.R. and L.K.; project administration, L.K. and I.R.; funding acquisition, L.K. and I.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Higher Education of the Russian Federation, project no. FEFE-2023-0004.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Polyak, B.T. *Introduction to Optimization*; Optimization Software: New York, NY, USA, 1987.
2. Nocedal, J.; Wright, S. *Numerical Optimization*; Series in Operations Research and Financial Engineering; Springer: New York, NY, USA, 2006.
3. Broyden, C.G. The convergence of a class of double-rank minimization algorithms. *J. Inst. Math. Appl.* **1970**, *6*, 76–79. [\[CrossRef\]](#)
4. Fletcher, R. A new approach to variable metric algorithms. *Comput. J.* **1970**, *13*, 317–322. [\[CrossRef\]](#)
5. Goldfarb, D. A family of variable metric methods derived by variational means. *Math. Comput.* **1970**, *24*, 23–26. [\[CrossRef\]](#)
6. Shanno, D.F. Conditioning of quasi-Newton methods for function minimization. *Math. Comput.* **1970**, *24*, 647–656. [\[CrossRef\]](#)
7. Javad Ebadi, M.; Fahs, A.; Fahs, H.; Dehghani, R. Competitive secant (BFGS) methods based on modified secant relations for unconstrained optimization. *Optimization* **2023**, *72*, 1691–1706. [\[CrossRef\]](#)
8. Vaziri, A.; Fang, H. Optimal inferential control of convolutional neural networks. In Proceedings of the 2025 American Control Conference (ACC), Denver, CO, USA, 8–10 July 2025; pp. 2603–2610. [\[CrossRef\]](#)
9. Rehman, H.; Peng, Z.-Y.; Yao, J.-C. Approximate subgradient extragradient methods for solving variational inequality problems: Convergence analysis and applications in signal and image processing. *Commun. Nonlinear Sci. Numer. Simulat.* **2026**, *152*, 109211. [\[CrossRef\]](#)
10. Xie, Z.; Li, M. Accelerated subgradient extragradient methods with increasing self-adaptive step size for variational inequalities. *Commun. Nonlinear Sci. Numer. Simulat.* **2025**, *147*, 108794. [\[CrossRef\]](#)
11. Sunthrayuth, P.; Adamu, A.; Muangchoo, K.; Ekvittayaniphon, S. Strongly convergent two-step inertial subgradient extragradient methods for solving quasi-monotone variational inequalities with applications. *Commun. Nonlinear Sci. Numer. Simulat.* **2025**, *150*, 108959. [\[CrossRef\]](#)
12. Carrabs, F.; Gaudioso, M.; Miglionico, G. A two-point heuristic to calculate the stepsize in subgradient method with application to a network design problem. *EURO J. Comput. Optim.* **2024**, *12*, 100092. [\[CrossRef\]](#)
13. Bulbul, K.G.; Kasimbeyli, R. Augmented Lagrangian based hybrid subgradient method for solving aircraft maintenance routing problem. *Comput. Oper. Res.* **2021**, *132*, 105294. [\[CrossRef\]](#)
14. Zhang, Y.; Khan, K. Evaluating subgradients for convex relaxations of dynamic process models by adapting current tools. *Comput. Chem. Eng.* **2024**, *180*, 108462. [\[CrossRef\]](#)
15. Shor, N.Z. *Methods of Minimization of Non-Differentiable Functions and Their Applications*; Naukova Dumka: Kiev, Ukraine, 1979. (In Russian)
16. Skokov, V.A. Note on minimization methods employing space stretching. *Cybern. Syst. Anal.* **1974**, *10*, 689–692. [\[CrossRef\]](#)

17. Krutikov, V.; Gutova, S.; Tovbis, E.; Kazakovtsev, L.; Semenko, E. Relaxation Subgradient Algorithms with Machine Learning Procedures. *Mathematics* **2022**, *10*, 3959. [\[CrossRef\]](#)
18. Tovbis, E.; Krutikov, V.; Kazakovtsev, L. Newtonian Property of Subgradient Method with Optimization of Metric Matrix Parameter Correction. *Mathematics* **2024**, *12*, 1618. [\[CrossRef\]](#)
19. Krutikov, V.N.; Petrova, T. Relaxation method of minimization with space extension in the subgradient direction. *Ekonom. Mat. Met.* **2003**, *39*, 106–119.
20. Shor, N. *Minimization Methods for Nondifferentiable Functions*; Springer: Berlin/Heidelberg, Germany, 1985.
21. Nemirovskii, A.S.; Yudin, D.B. *Problem Complexity and Method Efficiency in Optimization*; Wiley: Chichester, UK, 1983.
22. Cao, H.; Song, Y.; Khan, K. Convergence of subtangent-based relaxations of nonlinear programs. *Processes* **2019**, *7*, 221. [\[CrossRef\]](#)
23. Wolfe, P. Note on a method of conjugate subgradients for minimizing nondifferentiable functions. *Math. Program.* **1974**, *7*, 380–383. [\[CrossRef\]](#)
24. Lemarechal, C. An extension of Davidon methods to non-differentiable problems. *Math. Program. Study* **1975**, *3*, 95–109.
25. Demyanov, V.F. Nonsmooth optimization. In *Nonlinear Optimization; Lecture Notes in Mathematics*; Springer: Berlin/Heidelberg, Germany, 1981; Volume 1989, pp. 55–163.
26. Nurminskii, E.A.; Thien, D. Method of conjugate subgradients with constrained memory. *Autom. Remote Control* **2014**, *75*, 646–656. [\[CrossRef\]](#)
27. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
28. Loreto, M.; Xu, Y.; Kotval, D. A numerical study of applying spectral-step subgradient method for solving nonsmooth unconstrained optimization problems. *Comput. Oper. Res.* **2019**, *104*, 90–97. [\[CrossRef\]](#)
29. Long, Q.; Wu, C.; Wang, X. A system of nonsmooth equations solver based upon subgradient method. *Appl. Math. Comput.* **2015**, *251*, 284–299. [\[CrossRef\]](#)
30. Combettes, P.L. Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections. *IEEE Trans. Image Process.* **1997**, *6*, 493–506. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Yukawa, M.; Slavakis, K.; Yamada, I. Adaptive parallel quadraticmetric projection algorithms. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 1665–1680. [\[CrossRef\]](#)
32. Beck, A.; Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.* **2003**, *31*, 167–175. [\[CrossRef\]](#)
33. Zhukovskii, E.L.; Liptser, R.S. On a recurrence method for computing normal solutions of linear algebraic equations. *USSR Comp. Math. Math. Phys.* **1972**, *12*, 1–18. (In Russian) [\[CrossRef\]](#)
34. Hestenes, M.R.; Stiefel, E. Method of Conjugate Gradients for Solving Linear Systems. *J. Res. Natl. Bur. Stand.* **1952**, *49*, 409–436. [\[CrossRef\]](#)
35. Bunday, B.D. *Basic Optimization Methods*; Edward Arnold Limited: London, UK, 1984.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.