**Received:** May 21, 2022 ; **Accepted:** October 6, 2023 .
DOI. 10.30473/coam.2023.64138.1204
Winter-Spring (2024) Vol. 9, No. 1, (49-65)
**Research Article**

Open Access

**Control and Optimization in Applied Mathematics - COAM**

# Efficient Solution of Nonlinear Unconstrained Optimization Problems using Quasi-Newton Method: A Revised Approach

**Hajar Alimorad**

Department of Mathematics, Jahrom University, Jahrom, P.O. Box 74135-111, Iran.

✉ **Correspondence:**
Hajar Alimorad
**E-mail:**
h.alimorad@jahromu.ac.ir

**How to Cite**
Alimorad, H. (2024). "Efficient solution of nonlinear unconstrained optimization problems using quasi-Newton method: A revised approach", Control and Optimization in Applied Mathematics, 9(1): 49-65.

**Abstract.** While many real-world optimization problems typically involve multiple constraints, unconstrained problems hold practical and fundamental significance. They can arise directly in specific applications or as transformed versions of constrained optimization problems. Newton's method, a notable numerical technique within the category of line search algorithms, is widely used for function optimization. The search direction and step length play crucial roles in this algorithm. This paper introduces an algorithm aimed at enhancing the step length within the Broyden quasi-Newton process. Additionally, numerical examples are provided to compare the effectiveness of this new method with another approach.

**Keywords.** Optimization, Hessian matrix, Quasi-Newton method, Constrained and unconstrained problems.

**MSC.** 90C30; 90C53; 49M15.

## 1 Introduction

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable and smooth function. We consider the minimization problem,

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

and assume that the solution set of (1) is nonempty. To solve this problem, we first define some key terms. In this paper, we use the notation $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ to indicate the gradient of a differentiable function $f(x)$ and $H(x)$ to display the Hessian matrix of $f(x)$. In this regard, we use $g(x) = \nabla f(x)$ to simplify the expression of methods.

Newton's method can be used to determine the minimum value of a scalar objective function numerically. This solution is a kind of gradient method that solves the optimization of unconstraint functions. To do this, we first require to approximate the objective function $f(x)$ with the first three terms of its Taylor series around the point $x$:

$$f(\bar{x}) = f(x) + g(x)(\bar{x} - x) + \frac{1}{2}(\bar{x} - x)^t H(x)(\bar{x} - x).$$

If $\bar{x}$ is the minimizer of objective function $f(x)$, then $g(\bar{x}) = 0$. As a result, we have

$$g(\bar{x}) = g(x) + H(x)(\bar{x} - x) = 0, \tag{2}$$

therefore, $H(x)(\bar{x} - x) = -g(x)$ can be approximated as:

$$d = -H^{-1}g, \quad \bar{x} = x + d.$$

Accordingly, the condition for minimizing the solution can be stated as follows:

$$g(x^*) = 0, \quad d^t H(x^*)d > 0,$$

The Hessian matrix, which is the second sentence in the previous phrase, must be positive definite. Therefore, when applying Newton's method to minimize the function $f(x)$ in terms of optimization variables, the search direction is determined as $d = -H^{-1}g$. Subsequently, by finding the search direction $d$, the updated values of optimization variables in Newton's iterative method are obtained as $\bar{x} = x + d$.

When the Hessian matrix is positive definite, it can be demonstrated that $d = -H^{-1}g$ represents a decreasing direction. In many cases, obtaining the Hessian matrix can be a challenging task. In such situations, quasi-Newton methods can be employed to estimate the Hessian matrix through an iterative process.

Three examples of this can be summarized as follows [2]:

1. Broyden method:
   In this method, the Hessian matrix $H(x_j)$ is approximated by matrix $B_j$, where $B_j$ is the approximate Hessian matrix at the $j$th iteration of Newton's method. The option of $B_0 = H(x_0)$, $\triangle g = g_j - g_{j-1}$, $\triangle g = g_j - g_{j-1}$ and $\triangle x = x_j - x_{j-1}$ are considered in the approximation.

$$B_j = B_{j-1} + \frac{(\triangle g - B_{j-1}\triangle x)(\triangle x)^t}{(\triangle x)^t(\triangle x)}.$$

2. (Broyden, Fletcher, Goldfarb, and Shanno) BFGS method:
   The approximation of the Hessian matrix in step $j$ in this method is as follows:

$$B_j = B_{j-1} + \frac{(\triangle g)(\triangle g)^t}{(\triangle g)^t(\triangle x)} - \frac{B_{j-1}(\triangle x)(\triangle x)^t B_{j-1}}{(\triangle x)^t B_{j-1}(\triangle x)} .$$

3. (Symmetric Rank-one (SR1) method):
   The $SR1$ method approximates the Hessian matrix during the $j$-th iteration as described below:

$$B_j = B_{j-1} + \frac{(\triangle g - B_{j-1}\triangle x)(\triangle g - B_{j-1}\triangle x)^t}{(\triangle g - B_{j-1}\triangle x)^t(\triangle x)} .$$

Newton's method has undergone various improvements of unconstrained optimization to achieve convergence. For convex minimization problems with singular Hessian at solutions and objectives that are twice continuously differentiable, two regularized Newton methods were employed [6]. Additionally, local quadratic convergence was proven in [6] under a local error bound condition, without requiring isolated nonsingular solutions.

The regularized Newton's method has been applied to unconstrained convex optimization in [8]. Ueda and Yamashita introduced a regularized algorithm for non-convex minimization problems and established a global complexity bound while analyzing superlinear convergence [13]. In [15], a novel method with superlinear convergence for monotone equations was presented, utilizing a local error-bound assumption that is weaker than the standard non-singularity condition. For any convex function with a bounded optimal set, the regularized Newton's method (RNM) generates a sequence that converges to the optimal set from any starting point. In [10], a regularized Newton's method was proposed to solve unconstrained non-convex minimization problems without assuming the non-singularity of solutions. The authors demonstrated global and fast local convergence under appropriate conditions. Furthermore, a new modification of Newton's method incorporating a constant step length at each iteration was proposed in [3]. The convergence analysis for this iterative algorithm was established under suitable conditions.

In references [9] and [12], a Rational Approximation with Linear Numerator and Denominator (RALND) function, denoted as $r_i(x) : \mathbb{R}^n \to \mathbb{R}, i = 1, 2, \ldots, m$, is defined by

$$r_i(x) = d_0 + \frac{a_j^t(x - x_j)}{1 + b_j^t(x - x_j)}, \tag{3}$$

using undetermined coefficient vectors $a_j, b_j \in \mathbb{R}^n$. Based on work presented in [11], the following conditions are considered:

$$r_i(x_j) = g_i(x_j), \quad \nabla r_i(x_j) = \nabla g_i(x_j), \quad \nabla r_i(x_{j-1}) = \nabla g_i(x_{j-1}),$$

and

$$d_0 = g_i(x_j), \quad a_j = \nabla g_i(x_j),$$
$$b_j = \frac{1}{c_0}\left(\sqrt{\frac{c_0}{c_1}}(\nabla g_i(x_j))^t - (\nabla g_i(x_{j-1}))^t\right),$$

where

$$c_0 = (\nabla g_i(x_{j-1}))^t(x_{j-1} - x_j), \quad c_1 = (\nabla g_i(x_j))^t(x_{j-1} - x_j).$$

To solve the problem (1), we obtain the RALND function,

$$r_i(x) = g_i(x_j) + \frac{(\nabla g_i(x_j))^t (x - x_j)}{1 + \frac{1}{c_0}\left(\sqrt{\frac{c_0}{c_1}}(\nabla g_i(x_j))^t - (\nabla g_i(x_{j-1}))^t\right)(x - x_j)}, \tag{4}$$

where $x_j, x_{j-1} \in \mathbb{R}^n$ are the current and the previous points.

By utilizing Equation (4), we can express the RALND function with the same vector $b_j$ for all nonlinear functions $g_i(x)$ at point $x_j$, and approximate the nonlinear gradient equations $g(x) = 0$ as follows:

$$g(x_j + s) \approx R(x_j + s) = g(x_j) + \frac{H(x_j)s}{1 + b_j^t s} = 0, \tag{5}$$

where $s = x - x_j$ , $R(x) = (r_1(x), r_2(x), ..., r_n(x))^t$ and $H(x_j)$ is Hessian matrix of $f(x)$ in point $x_j$. By Considering $s = s_j = x_{j+1} - x_j$ and linearizing the approximate Equation (5):

$$\begin{aligned} &\frac{g(x_j)\left(1 + b_j^t s_j\right) + H(x_j)s_j}{1 + b_j^t s_j} = 0 \\ &\Rightarrow g(x_j) + g(x_j)b_j^t s_j + H(x_j)s_j = 0 \\ &\Rightarrow g(x_j)b_j^t s_j + H(x_j)s_j = -g(x_j). \end{aligned}$$

Therefore, we can obtain a new iterated formula

$$\left(H(x_j) + g(x_j)b_j^t\right) s_j = -g(x_j). \tag{6}$$

Assuming the matrix $H(x_j) + g(x_j)b_j^t$ is invertible, then

$$x_{j+1} = x_j - \left(H(x_j) + g(x_j)b_j^t\right)^{-1} g(x_j), \tag{7}$$

where $b_j \in \mathbb{R}^n$, when $b_j = 0$, the iterated formula (7) reduces to classical Newton's method.

**Remark 1.** In terms of computation, it is important to note that if in the relation (7),

$$H(x_j) + g(x_j)b_j^t,$$

is not invertible, instead of using $\left(H(x_j) + g(x_j)b_j^t\right)^{-1}$, we can solve the equation system (6).

In Theorem 2, we prove that for $\alpha > 0$, if $\|H^{-1}(x^*)\| < \alpha$, then $\left(H(x_j) + g(x_j)b_j^t\right)$ is invertible.

If this condition is not satisfied and $\left(H(x_j) + g(x_j)b_j^t\right)$ is not invertible, we prove that the system (6) has infinite solutions; indicating that the system does not have solutions.

**Theorem 1.** If the matrix $\left(H(x_j) + g(x_j)b_j^t\right)$ is not invertible, then the equation system (6) has infinite solutions.

*Proof.* Let $H(x_j) = [a_1 \dots a_n]$ and $g(x_j)b_j^t = [b_1 \dots b_n]$ be matrices, where $a_i$ and $b_j$ represent the columns of matrices $H(x_j)$ and $g(x_j)b_j^t$, respectively. For simplicity, let's denote $-g(x_j) = b$. It is evident that the vectors $b_1, b_2, \dots, b_n$ serve as coefficients for $b$. Our goal is to prove the following:

$$\text{rank}\left[H(x_j) + g(x_j)b_j^t \ \vdots \ b\right] = rank\left[H(x_j) + g(x_j)b_j^t\right],$$

the matrix $\left[H(x_j)+g(x_j)b_j^t \ \vdots \ b\right]$ is the augmented matrix of equations system (6). In this context, according to Theorem 1.2 in reference [7], the system has infinite number of solutions.

$$\text{rank}\left[H(x_j)+g(x_j)b_j^t\right]=\begin{cases}\text{rank}\left(H(x_j)\right), & (i)\\ \text{if vector } b \text{ is a linear dependent with respect to}\\ \text{matrix columns } H(x_j);\\ \text{rank}\left(H(x_j)\right)+1, & (ii)\\ \text{if vector } b \text{ is a linear independent with respect to}\\ \text{matrix columns } H(x_j).\end{cases}$$

In state $(i)$, since vector $b$ is linearly dependent on the matrix columns $H(x_j)$:

$$\text{rank}\left[H(x_j)+g(x_j)b_j^t\right]=\text{rank}\left[H(x_j)+g(x_j)b_j^t \ \vdots \ b\right].$$

In state $(ii)$, we consider

$$c_1(a_1+b_1)+c_2(a_2+b_2)+\cdots+c_n(a_n+b_n)+c_{n+1}b=0. \tag{8}$$

Since $b_i$s', $1\leq i\leq n$, coefficients of $b$, can be expressed as follows:

$$b_1=v_1b, \quad b_2=v_2b, \quad \ldots, \ b_n=v_nb,$$

we can then, rewrite Equation (8) in the following form:

$$c_1a_1+c_2a_2+\cdots+c_na_n+\left(c_1v_1+\cdots+c_nv_n+c_{n+1}\right)b=0.$$

Because $b$ and vectors $a_j$, $1\leq j\leq n$, are linearly independent:

$$\text{rank}\left[H(x_j)+g(x_j)b_j^t \ \vdots \ b\right]=\text{rank}\left(H(x_j)\right)+1.$$

□

In this paper, we draw inspiration from the works in [9], [11] and the Broyden method [2] to propose a revised quasi-Newton's method for unconstraint nonlinear optimization. The local convergence of this method, based on the approach presented in [11] is achieved under certain assumptions.

This paper is organized as follows: The next section is dedicated to the explanation of the new method. In Section 3, we elaborate on the local convergence of the proposed method. Several numerical simulations are conducted in Section 4. Section 5 dedicated to conclusion and remarks.

## 2 The Revised Quasi-Newton's Method

To solve Problem (1), we substitute $s=-\triangle x$ in Equation (5). Let us define:

$$g(x_{j-1})=g(x_j)-\frac{H(x_j)\triangle x}{1-b_j^t\triangle x}.$$

Let

$$\beta_j = 1 - b_j^t \triangle x, \quad \triangle g = g(x_j) - g(x_{j-1}). \tag{9}$$

Therefore,

$$\beta_j \triangle g = H(x_j) \triangle x,$$

then

$$\beta_j = \frac{(\triangle g)^t H(x_j) \triangle x}{(\triangle g)^t \triangle g}. \tag{10}$$

Based on (9), we can derive

$$b_j = \frac{(1-\beta_j) a_j}{a_j^t \triangle x}, \tag{11}$$

where $a_j \in \mathbb{R}^n,\ a_j^t \triangle x \neq 0$. Let $a_j = \triangle x$, then

$$b_j = \frac{(1-\beta_j)\triangle x}{(\triangle x)^t \triangle x} = \frac{(\triangle g)^t \left(\triangle g - H(x_j)\triangle x\right)}{(\triangle g)^t \triangle g} \frac{\triangle x}{(\triangle x)^t \triangle x}. \tag{12}$$

using (12), we obtain a rational approximation function

$$(g(x_j)) b_j^t = \frac{(\triangle g)^t \left(\triangle g - H(x_j)\triangle x\right)}{(\triangle g)^t \triangle g} \frac{g(x_j)(\triangle x)^t}{(\triangle x)^t \triangle x}. \tag{13}$$

Following the concept presented by Broyden [2], we approximate the Hessian matrix in each iteration of the aforementioned rational model using matrix $B_j$. Consequently, we propose a revised quasi-Newton's method as follows:

$$\begin{cases} x_{j+1} = x_j - \left(B_j + (g(x_j)) b_j^t\right)^{-1} g(x_j), \\ (g(x_j)) b_j^t = \frac{(\triangle g)^t (\triangle g - B_j \triangle x)}{(\triangle g)^t \triangle g} \frac{g(x_j)(\triangle x)^t}{(\triangle x)^t \triangle x}, \end{cases} \tag{14}$$

where

$$B_j = B_{j-1} + \frac{(\triangle g - B_{j-1}\triangle x)(\triangle x)^t}{(\triangle x)^t (\triangle x)}.$$

The primary problem with (14) is that even if $B_{j-1}$ is symmetric, $\left(B_j + (g(x_j)) b_j^t\right)$ may not be. Thus, we aim to find $\left(B_j + (g(x_j)) b_j^t\right)^+$ that obeys

$$\left[(B_j + (g(x_j)) b_j^t)^+\right]^t = \left(B_j + (g(x_j)) b_j^t\right)^+,$$

as well as (14).

One approach to constructing a symmetric $\left(B_j + (g(x_j)) b_j^t\right)^+$ from $\left(B_j + (g(x_j)) b_j^t\right)$ is as follows:

$$\left(B_j + (g(x_j)) b_j^t\right)^+ = \frac{1}{2}\left[\left(B_j + (g(x_j)) b_j^t\right) + \left(B_j + (g(x_j)) b_j^t\right)^t\right],$$

which represents the symmetric matrix closest to $\left(B_j + (g(x_j)) b_j^t\right)$ [4].

**Algorithm 2** Revised quasi-Newton method

**step 1:** Given an initial point $x_0 \in \mathbb{R}^n$, calculate $B_0 = H(x_0)$ and set $j = 0$ and $b_0 = 0$.

**step 2:** Calculate $g(x_j)$. If $\| g(x_j) \| \leq \varepsilon$, then stop.

**step 3:** Calculate $x_{j+1}$ and $g_j b_j^t$ as follows:

$$\begin{cases} x_{j+1} = x_j - \left[(B_j + (g(x_j))b_j^t)^+\right]^{-1} g_j, \\ g_j b_j^t = \frac{(\triangle g)^t(\triangle g - B_j \triangle x)}{(\triangle g)^t \triangle g} \frac{g(x_j)(\triangle x)^t}{(\triangle x)^t \triangle x}, \end{cases}$$

**step 4:** Update $B_{j+1}$ using the equation:

$$B_{j+1} = B_j + \frac{(\triangle g_j - B_j \triangle x_j)(\triangle x_j)^t}{(\triangle x_j)^t (\triangle x_j)},$$

where $\triangle x_j = x_{j+1} - x_j$ and $\triangle g_j = g_{j+1} - g_j$.

**step 5:** Set $j = j + 1$ and return to step 2.

As we are aware, the conventional quasi-Newton equation only utilizes information from the current point. To obtain a better approximation for the Hessian matrix in (14), we employ the method proposed by [11] to construct a new quasi-Newton equation, assuming a quadratic relationship between the last three iterations of information. Using Equation (2), we can deduce

$$\begin{cases} g(x_j) + B_{j+1}(x_{j+1} - x_j) = g(x_{j+1}), \\ g(x_{j-1}) + B_{j+1}(x_j - x_{j-1}) = g(x_j), \end{cases}$$

hence,

$$\begin{cases} B_{j+1} \triangle x_j = \triangle g_j, \\ B_{j+1} \triangle x_{j-1} = \triangle g_{j-1}, \end{cases} \tag{15}$$

where $\triangle x_j = x_{j+1} - x_j$, $\triangle x_{j-1} = x_j - x_{j-1}$.

Considering (15), we can deduce:

$$B_{j+1}\left(\triangle x_j - \delta_j \triangle x_{j-1}\right) = \triangle g_j - \delta_j \triangle g_{j-1}.$$

Therefore, the new quasi-Newton equation is given by

$$B_{j+1} \widetilde{\triangle x} = \widetilde{\triangle g}, \tag{16}$$

where $B_{j+1}$ approximates $H(x_{j+1})$, and

$$\begin{cases} \widetilde{\triangle x} = \triangle x_j - \delta_j \triangle x_{j-1}, \\ \widetilde{\triangle g} = \triangle g_j - \delta_j \triangle g_{j-1}, \\ \delta_j = \frac{\|\triangle x_j\|^2}{\|\triangle x_{j-1}\|(2\|\triangle x_j\| + \|\triangle x_{j-1}\|)}. \end{cases} \tag{17}$$

When $\delta_j = 0$, the system (17) simplifies to the conventional quasi-Newton equation. With the introduction of the new Equation (17), we can rewrite formula (14) and derive the following algorithm:

$$\begin{cases} (B_j + (g(x_j))b_j^t)^+ = \frac{1}{2}\left[(B_j + (g(x_j))b_j^t) + (B_j + (g(x_j))b_j^t)^t\right], \\ x_{j+1} = x_j - \left[(B_j + (g(x_j))b_j^t)^+\right]^{-1} g(x_j), \\ (g(x_j))b_j^t = \frac{(\triangle g)^t(\triangle g - B_j\triangle x)}{(\triangle g)^t\triangle g}\frac{g(x_j)(\triangle x)^t}{(\triangle x)^t\triangle x}, \end{cases}$$

where

$$B_j = B_{j-1} + \frac{(\widetilde{\triangle} g_{j-1} - B_{j-1}\widetilde{\triangle x}_{j-1})(\widetilde{\triangle x}_{j-1})^t}{(\widetilde{\triangle x}_{j-1})^t(\widetilde{\triangle x}_{j-1})}.$$

**Algorithm 3** New revised quasi-Newton method

**step 1:** Given an initial point $x_0 \in \mathbb{R}^n$, calculate $B_0 = H(x_0)$ and set $j = 0$ and $b_0 = 0$.

**step 2:** Compute $g(x_j)$. If $\| g(x_j) \| \leq \varepsilon$, then stop.

**step 3:** Calculate $x_{j+1}$ and $g_j b_j^t$ using equations:

$$\begin{cases} x_{j+1} = x_j - \left[(B_j + g_j b_j^t)^+\right]^{-1} g_j, \\ g_j b_j^t = \frac{(\triangle g)^t(\triangle g - B_j\triangle x)}{(\triangle g)^t\triangle g}\frac{g(x_j)(\triangle x)^t}{(\triangle x)^t\triangle x}. \end{cases}$$

**step 4:** Compute $\widetilde{\triangle x}_j$, $\widetilde{\triangle g}_j$ using (17) and update $B_j$ as follows:

$$B_{j+1} = \begin{cases} B_j + \frac{(\triangle g_j - B_j\triangle x_j)(\triangle x_j)^t}{(\triangle x_j)^t(\triangle x_j)} & j = 0, \\ B_j + \frac{(\widetilde{\triangle} g_j - B_j\widetilde{\triangle x}_j)(\widetilde{\triangle x}_j)^t}{(\widetilde{\triangle x}_j)^t(\widetilde{\triangle x}_j)} & j \geq 1. \end{cases}$$

**step 5:** Set $j = j + 1$ and return to step 2.

**Remark 2.** The presented algorithm can be utilized for solving convex problems. However, for non-convex problems, the problem must satisfy certain necessary conditions, and the algorithm may require modifications. In [14], the modified Broyden family method for non-convex function has been investigated.

## 3 Local Convergence Analysis

In this section, we prove the local convergence of the aforementioned method for nonlinear unconstrained optimization problems. The notation, $\| \cdot \|$ represents the $l^2$ vector norm or Frobenius matrix norm. Therefore, we require the following assumptions:

**Assumption 2.** Consider Problem (1), where the gradient $g(x)$ is continuously differentiable and bounded in the open convex set $\Omega \subseteq \mathbb{R}^n$. Let $x^*$ be a solution of (1), and assume that the Hessian

matrix $H(x^*)$ is nonsingular. Furthermore, suppose there exists a constants $\alpha > 0$ and $c > 0$ such that for all $x, y \in \Omega$, the following conditions hold:

$$\|H^{-1}(x^*)\| \leq \alpha \quad \text{and} \quad \|g(x) - g(y)\| \leq c\|x - y\|.$$

**Definition 1.** A function $f$ is Lipschitz continuous with a constant $\gamma$ in a set $X$, denoted as $f \in Lip(X)$, if for every $x, y \in X$, the inequality

$$|f(x) - f(y)| \leq \gamma\|x - y\|.$$

holds.

**Lemma 1.** [9, 11] Assuming that $g : \mathbb{R}^n \to \mathbb{R}^n$ satisfies Assumption 2, we have for all $x + y \in \Omega$,

$$\|g(x + y) - g(x) - H(x)\| \leq \frac{c}{2}\|y\|^2.$$

Moreover, there exist $\varepsilon > 0$ and $0 < m < M$ such that

$$m\|x_1 - x_2\| \leq \|g(x_1) - g(x_2)\| \leq M\|x_1 - x_2\|,$$

holds for all $x_1, x_2 \in \Omega$ satisfying $max\{\|x_1 - x^*\|, \|x_2 - x^*\|\} \leq \varepsilon$.

**Lemma 2.** [5] Let $F, G \in \mathbb{R}^{n\times n}$ be matrices and $\|FG\| \leq \|F\|\|G\|$. If $F$ is nonsingular and $\|F^{-1}(G - F)\| < 1$, then $G$ is also nonsingular and

$$\|G^{-1}\| \leq \frac{\|F^{-1}\|}{1 - \|F^{-1}(G - F)\|}.$$

**Lemma 3.** [4, 11] Consider an open convex set $\Omega \subseteq \mathbb{R}^n$ that includes $x_j, x_{j+1}$, with $x_j \neq x^*$. Let $g : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable in $\Omega$, and let $H(x) \in Lip(\Omega)$, $B_j \in \mathbb{R}^{n\times n}$ and $B_{j+1}$ be defined by Algorithm 2. Additionally, assume that Assumption 2 holds. Then, we have

$$\|E_{j+1}\| \leq \|E_j\| + \frac{c}{2}\left(\|e_{j+1}\| + \|e_j\|\right), \tag{18}$$

where $E_k = B_j - H(x^*)$, $k = j, j + 1$; $e_i = x_i - x^*$, $i = j, j + 1$.

**Lemma 4.** [5] Consider an open convex set $\Omega \subseteq \mathbb{R}^n$ that includes $x_{j-1}$, $x_j$ , $x_{j+1}$, with $x_{j-1} \neq x^*$ and $x_j \neq x^*$. Let $g : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable in $\Omega$, $H(x) \in Lip(\Omega)$, $B_j \in \mathbb{R}^{n\times n}$ and let $B_{j+1}$ be defined by Algorithm 3. Furthermore, assume that Assumption 2 holds. If $\|\triangle x_j\| \neq 0$, $\|\triangle x_{j-1}\| \neq 0$, $x^* \in \Omega$, then

$$\|E_{j+1}\| \leq \begin{cases} \|E_j\| + \frac{c}{2}\left(\|e_{j+1}\| + \|e_j\|\right), & j = 0, \\ \|E_j\| + \frac{c}{2}\left(2\|e_{j+1}\| + 3\|e_j\| + \|e_{j-1}\|\right), & j \geq 1, \end{cases} \tag{19}$$

where

$$E_k = B_j - H(x^*), \quad k = j, j + 1,$$

$$e_i = x_i - x^*, \quad i = j - 1, j, j + 1.$$

Using the lemmas mentioned earlier, we can prove the following convergence result.

**Theorem 2.** Consider Problem (1), where the gradient $g(x)$ satisfies Assumption 2, Lemmas 1 and 3. Assume there exists a small positive constant $\varepsilon$ such that $\|x_0 - x^*\| \leq \varepsilon$. If $x^* \in \mathbb{R}^n$ is the solution of (1), the sequence $\{x_j\}$ generated by Algorithm 2 is well-defined and converges to $x^*$.

*Proof.* We aim to prove that

$$\|x_n - x^*\| \leq \varepsilon, \quad \|B_n - H(x^*)\| \leq c_1\delta,$$

when $1 \leq c_1 < 2$. First, we use induction to prove that

$$\|B_n - H(x^*)\| \leq (2 - 2^{-n})\delta, \tag{20}$$

$$\|x_{n+1} - x^*\| \leq \frac{1}{2}\|x_n - x^*\|, \quad n = 0, 1, 2, \ldots. \tag{21}$$

When $n = 0$, Algorithm 2 reduces to classical Newton's method, and the hypothesis holds true. Now, assuming that (20) and (21) hold for $n = 0, 1, 2, \ldots, k-1$. Hence,

$$\|B_{k-1} - H(x^*)\| \leq (2 - 2^{-(k-1)})\delta, \tag{22}$$

$$\|x_k - x^*\| \leq \frac{1}{2}\|x_{k-1} - x^*\|. \tag{23}$$

For $n = k$, we have

$$\|B_k - H(x^*)\| \overset{(18)}{\leq} \|B_{k-1} - H(x^*)\| + \frac{c}{2}\left(\|x_k - x^*\| + \|x_{k-1} - x^*\|\right) \tag{24}$$

$$\overset{(22),(23)}{\leq} (2 - 2^{-(k-1)})\delta + \frac{3}{4}c\|x_{k-1} - x^*\|. \tag{25}$$

From (21) and $\|x_0 - x^*\| = \|e_0\| \leq \varepsilon$, we obtain

$$\|x_{k-1} - x^*\| \leq 2^{-(k-1)}\|x_0 - x^*\| \leq 2^{-(k-1)}\varepsilon, \tag{26}$$

Substituting (26) into (24), we have

$$\|B_k - H(x^*)\| \leq (2 - 2^{-(k-1)})\delta + \frac{3}{4}c \times 2^{-(k-1)}\varepsilon,$$

by considering $\frac{3}{2}c\varepsilon \leq \delta$,

$$\|B_k - H(x^*)\| \leq (2 - 2^{-(k-1)} + 2^{-k})\delta \leq (2 - 2^{-k})\delta, \tag{27}$$

which verifies (20).

To prove (21), we need to demonstrate that $B_k + g(x_k)b_k^t$ is invertible for the iteration to be well-defined. From $\|H^{-1}(x^*)\| \leq \alpha$, we have

$$\|H^{-1}(x^*)\left(B_k + g(x_k)b_k^t - H(x^*)\right)\| \leq \|H^{-1}(x^*)\|\left(\|B_k - H(x^*)\| + \|g(x_k)b_k^t\|\right) \tag{28}$$

$$\leq \alpha\left((2 - 2^{-k})\delta + \|g(x_k)b_k^t\|\right). \tag{29}$$

Considering $\|g(x_k)b_k^t\|$, based on Assumption 2 and Lemma 1, we have

$$\|g(x_k)b_k^t\| \leq \|g(x_k)\|\|b_k\|$$

$$
\begin{aligned}
&\overset{(14)}{=} \|g(x_k)\| \frac{\|(\triangle g_{k-1})^t\,(\triangle g_{k-1} - B(x_k)\triangle x_{k-1})\,\|}{\|(\triangle g_{k-1})^t \triangle g_{k-1}\|} \frac{\|(\triangle x_{k-1})^t\|}{\|(\triangle x_{k-1})^t \triangle x_{k-1}\|} \\
&\leq \|g(x_k)\| \frac{\|g(x_k) - g(x_{k-1}) - B_k \triangle x_{k-1}\|}{\|\triangle g_{k-1}\|\|\triangle x_{k-1}\|} \\
&= \|g(x_k)\| \frac{\|g(x_k) - g(x_{k-1}) - H(x^*)\triangle x_{k-1} + H(x^*)\triangle x_{k-1} - B_k \triangle x_{k-1}\|}{\|\triangle g_{k-1}\|\|\triangle x_{k-1}\|} \\
&\leq \|g(x_k)\| \frac{\frac{1}{2}\|\triangle x_{k-1}\|^2 + \|B_k - H(x^*)\|\|\triangle x_{k-1}\|}{\|\triangle g_{k-1}\|\|\triangle x_{k-1}\|} \\
&\leq \|g(x_k)\| \frac{\frac{1}{2}\|\triangle x_{k-1}\| + \|B_k - H(x^*)\|}{\|\triangle g_{k-1}\|} \\
&\overset{(20)}{\leq} \|g(x_k)\| \left( \frac{1}{2} \frac{\|\triangle x_{k-1}\|}{\|\triangle g_{k-1}\|} + \frac{2\delta}{\|\triangle g_{k-1}\|} \right).
\end{aligned} \tag{30}
$$

Using the induction hypotheses, we have $\|x_{k-1} - x^*\| \leq \varepsilon$. From (21) and $\|e_0\| \leq \varepsilon$, for $n = k-1$, we obtain

$$
\|x_k - x^*\| \leq \frac{1}{2}\|x_{k-1} - x^*\| \leq \left(\frac{1}{2}\right)^k \|x_0 - x^*\| \leq \frac{1}{2}\|x_0 - x^*\| \leq \varepsilon, \tag{31}
$$

thus, $\|x_k - x^*\| \leq \varepsilon$. Based on Assumption 2, for all $x \in \Omega$, $g(x)$ is bounded Therfore, it is clear that $\triangle g_{k-1} = g(x_k) - g(x_{k-1})$ is bounded. Now assume there exists a constant $\xi > 0$ such that $\frac{1}{\|\triangle g_{k-1}\|} \leq \xi$, as per Lemma 1, we have

$$
m\|\triangle x_{k-1}\| \leq \|\triangle g_{k-1}\|.
$$

Hence,

$$
\frac{1}{2}\|\triangle x_{k-1}\| \leq \frac{1}{2m}\|\triangle g_{k-1}\|.
$$

Consequently, according to the last Equation (30),

$$
\begin{aligned}
\|g(x_k)b_k^t\| &\leq \|g(x_k)\| \left(\frac{1}{2m} + 2\delta\xi\right) \\
&= \|g(x_k) - g(x^*)\| \left(\frac{1}{2m} + 2\delta\xi\right) \\
&\overset{(Assumption 2)}{\leq} c\|x_k - x^*\| \left(\frac{1}{2m} + 2\delta\xi\right) \\
&\overset{(31)}{\leq} 2^{-k} c\varepsilon \left(\frac{1}{2m} + 2\delta\xi\right).
\end{aligned}
$$

By selecting appropriate values for $m, \xi, \varepsilon$ and $\delta$ such that

$$
6\alpha\delta \leq 1, \tag{32}
$$

$$
\frac{3}{2}c\varepsilon \leq \delta, \tag{33}
$$

$$
\frac{1}{2m} + 2\delta\xi \leq \frac{3}{4}, \tag{34}
$$

as a result, we can derive, from equations (28), (32), (33), and (34),

$$
\|H^{-1}(x^*)\left(B_k + g(x_k)b_k^t - H(x^*)\right)\| \leq \alpha \left((2 - 2^{-k})\delta + 2^{-k} c\varepsilon \left(\frac{1}{2m} + 2\delta\xi\right)\right)
$$

$$\begin{aligned}
&\leq \alpha\left((2-2^{-k})\delta + 2^{-(k+1)}\delta\right)\\
&\leq \alpha\left(2-2^{-(k+1)}\delta\right)\\
&\leq 2\alpha\delta \leq \frac{1}{3}.
\end{aligned} \tag{35}$$

According to Lemma 2, and with $M = H(x^*)$ and $N = B_k + g(x_k)b_k^t$, we can obtain $\|M^{-1}(N - M)\| < 1$. Consequently, $B_k + g(x_k)b_k^t$ becomes invertible, and

$$\begin{aligned}
\| \left(B_k + g(x_k)b_k^t\right)^{-1} \| &\leq \frac{\|H^{-1}(x^*)\|}{1 - \|H^{-1}(x^*)\left(B_k + g(x_k)b_k^t - H(x^*)\right)\|}\\
&\leq \frac{\|H^{-1}(x^*)\|}{1-\frac{1}{3}}\\
&= \frac{3}{2}\|H^{-1}(x^*)\|\\
&\leq \frac{3}{2}\alpha.
\end{aligned} \tag{36}$$

As a consequence, $x_{k+1}$ is well-defined. Utilizing Algorithm 2, and considering $g(x^*) = 0$, we have

$$\begin{aligned}
x_{k+1} - x^* &= x_k - \left(B_k + g(x_k)b_k^t\right)^{-1} g(x_k) - x^*\\
&= x_k - x^* - \left(B_k + g(x_k)b_k^t\right)^{-1}\left(g(x_k) - g(x^*)\right),
\end{aligned}$$

and

$$\begin{aligned}
e_{k+1} &= e_k - \left(B_k + g(x_k)b_k^t\right)^{-1}\left(g(x_k) - g(x^*)\right),\\
\left(B_k + g(x_k)b_k^t\right)e_{k+1} &= \left(B_k + g(x_k)b_k^t - H(x^*)\right)e_k + \left(-g(x_k) + g(x^*) + H(x^*)e_k\right).
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\|e_{k+1}\| \leq \| &\left(B_k + g(x_k)b_k^t\right)^{-1} \|\\
&\times \left(\| - g(x_k) + g(x^*) + H(x^*)e_k\| + \|B_k + g(x_k)b_k^t - H(x^*)\|\|e_k\|\right).
\end{aligned} \tag{37}$$

Employing Lemma 1, (28) and (36), we can conclude that

$$\|e_{k+1}\| \leq \frac{3}{2}\alpha\left(\frac{c}{2}\|e_k\| + (2 - 2^{-(k+1)})\delta\right)\|e_k\|. \tag{38}$$

Moreover, based on (21) and $\|e_0\| \leq \varepsilon$ ,

$$\frac{c}{2}\|e_k\| \leq \frac{c}{2}\left(2^{-k}\|e_0\|\right) \leq 2^{-(k+1)}c\varepsilon,$$

by the assumptions (33) we can conclude that

$$\frac{c}{2}\|e_k\| \leq 2^{-(k+1)}c\varepsilon \leq \frac{2}{3} \times 2^{-(k+1)}\delta. \tag{39}$$

By employing equations (38) and (39), we obtain

$$\|e_{k+1}\| \leq \frac{3}{2}\alpha\left(\frac{2}{3} \times 2^{-(k+1)}\delta + (2 - 2^{-(k+1)})\delta\right)\|e_k\|$$

$$\leq 3\alpha\delta\|e_k\|$$
$$\overset{(32)}{\leq} \frac{1}{2}\|e_k\|.$$

Thus, we have

$$\|x_{k+1} - x^*\| \leq \frac{1}{2}\|x_k - x^*\| \leq \left(\frac{1}{2}\right)^{k+1}\|x_0 - x^*\| \leq \frac{1}{2}\|x_0 - x^*\| \leq \varepsilon.$$

Consequently, the sequence $\{x_n\}$ converges to $x^*$. □

**Theorem 3.** Suppose in Problem (1), the gradient $g(x)$ satisfies Assumption 2, as well as lemmas 1 and 4. Let us assume the existence of a small positive constant $\varepsilon$ such that $\|x_0 - x^*\| \leq \varepsilon$. If $x^* \in \mathbb{R}^n$ represents the solution of (1), then the sequence $\{x_n\}$ generated by Algorithm 3 is well-defined and converges to $x^*$.

*Proof.* Similar to Theorem 2, our aim is to prove that

$$\|x_n - x^*\| \leq \varepsilon, \quad \|B_n - H(x^*)\| \leq c_1\delta,$$

when $1 \leq c_1 < 2$. First, we establish the proof by induction that

$$\|B_n - H(x^*)\| \leq (2 - 2^{-n})\delta, \tag{40}$$
$$\|x_{n+1} - x^*\| \leq \frac{1}{2}\|x_n - x^*\|, \quad n = 0, 1, 2, \ldots. \tag{41}$$

When $n = 0$, Algorithm 3 reduces to the classical Newton's method, and the hypothesis holds true. Now, let's assume that (40) and (41) hold true for $n = 0, 1, 2, \ldots, k-1$. Therefore,

$$\|B_{k-1} - H(x^*)\| \leq (2 - 2^{-(k-1)})\delta, \tag{42}$$
$$\|x_k - x^*\| \leq \frac{1}{2}\|x_{k-1} - x^*\|. \tag{43}$$

For $n = k$, we can infer from Lemma 4 and the induction hypotheses that

$$\|B_k - H(x^*)\| \leq \|B_{k-1} - H(x^*)\| + \frac{c}{2}\left(2\|e_k\| + 3\|e_{k-1}\| + \|e_{k-2}\|\right)$$
$$\leq (2 - 2^{-(k-1)})\delta + \frac{3}{2}c\|e_{k-2}\|. \tag{44}$$

Based on (41) and $\|x_0 - x^*\| = \|e_0\| \leq \varepsilon$, we have

$$\|e_{k-2}\| \leq 2^{-(k-2)}\|e_0\| \leq 2^{-(k-2)}\varepsilon. \tag{45}$$

By substituting (45) into (44) and assuming that $6c\varepsilon \leq \delta$, we can obtain

$$\|B_k - H(x^*)\| \leq (2 - 2^{-(k-1)})\delta + \frac{3c}{2} \times 2^{-(k-2)}\varepsilon$$
$$\leq (2 - 2^{-(k-1)} + 2^{-k})\delta$$
$$\leq (2 - 2^{-k})\delta,$$

which verifies (40).

To prove (41), it is necessary to demonstrate that $B_k + g(x_k)b_k^t$ is invertible, ensuring that the iteration is well-defined. The proof process is similar to Theorem 2 and the process of proving relation (21). Therefore, we will not delve into the details here, but we can conclude that $x_{k+1}$ is well-defined.

Thus, we can conclude

$$\|x_{k+1} - x^*\| \leq \frac{1}{2}\|x_k - x^*\| \leq \left(\frac{1}{2}\right)^{k+1} \|x_0 - x^*\| \leq \frac{1}{2}\|x_0 - x^*\| \leq \varepsilon,$$

Then, the sequence $\{x_n\}$ converges to $x^*$. □

## 4 Numerical Results

Now, to demonstrate the effectiveness of our method, we have solved several numerical examples, which were extracted from [1]. Additionally, we have compared our proposed method with the Broyden quasi-Newton's method in terms of runtime and the required number of iterations. The numerical results are presented in Table 1, where "problem" denotes the test problem, "objfun" represents the optimal value of the objective function, "dim" indicates the dimension of the test problem, "iter" corresponds to the number of iterations, and "time" denotes the CPU execution time in seconds. Moreover, we have assumed $\varepsilon = 10^{-6}$.

**Table 1:** Numerical results.

| problem | dim | Algorithm 2 | | | Algorithm 3 | | | Broyden method | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | objfun | time | iter | objfun | time | iter | objfun | time | iter |
| 1 | 100 | $2.4489e+03$ | 0.391 | 20 | $2.4489e+03$ | 0.391 | 21 | $2.4492e+03$ | 0.390 | 21 |
| 1 | 1000 | $2.4489e+04$ | 2.358 | 25 | $2.4489e+04$ | 2.316 | 22 | $2.4492e+04$ | 2.290 | 18 |
| 2 | 100 | $6.0786e-07$ | 0.311 | 16 | $6.0787e-07$ | 0.308 | 16 | $6.0786e-07$ | 0.303 | 16 |
| 2 | 1000 | $6.0786e-06$ | 5.023 | 20 | $6.0794e-06$ | 4.943 | 18 | $6.0796e-06$ | 4.909 | 15 |
| 3 | 100 | 99.0015 | 0.689 | 50 | 99.0005 | 0.668 | 18 | 98.7220 | 0.658 | 16 |
| 3 | 1000 | 999.0019 | 12.083 | 50 | 998.0015 | 11. 127 | 20 | 998.7221 | 11.070 | 18 |
| 4 | 100 | $0.148e-05$ | 0.558 | 7 | $0.20e-06$ | 0.448 | 8 | $0.21e-06$ | 0.397 | 8 |
| 4 | 1000 | $5.549e-05$ | 15.392 | 18 | $2.03e-07$ | 13.816 | 17 | $4.09e-05$ | 12.120 | 18 |
| 5 | 100 | $-49.82$ | 0.2332 | 12 | $-50.030$ | 0.259 | 9 | $-50.031$ | 0.1624 | 8 |
| 5 | 1000 | $-499.326$ | 8.9342 | 28 | $-500.302$ | 8.499 | 40 | $-500.312$ | 8.617 | 46 |
| 6 | 100 | 50.562 | 0.329 | 26 | 50.045 | 0.252 | 24 | 50.080 | 0.231 | 17 |
| 6 | 1000 | 525.818 | 1.500 | 50 | 500.945 | 1.328 | 50 | 500.799 | 1.354 | 50 |
| 7 | 100 | 420.747 | 1.979 | 40 | 399.962 | 1.958 | 21 | 399.962 | 1.957 | 14 |
| 7 | 1000 | $4.244e+03$ | 3.093 | 30 | $4.245e+03$ | 3.121 | 27 | $3.964e+03$ | 3.051 | 17 |
| 8 | 100 | $1.0000e-03$ | 0.2469 | 4 | $8.7800e-04$ | 0.254 | 4 | $2.9097e-04$ | 0.231 | 3 |
| 8 | 1000 | $0.8818e-02$ | 2.546 | 9 | $0.5200e-02$ | 3.335 | 8 | $0.1573e-02$ | 3.783 | 8 |
| 9 | 100 | $-50$ | 0.244 | 6 | $-50$ | 0.244 | 6 | $-50$ | 0.241 | 7 |
| 9 | 1000 | $-500$ | 2.546 | 9 | $-500$ | 3.336 | 8 | $-500$ | 3.783 | 8 |
| 10 | 100 | $-3.7711e+06$ | 0.978 | 6 | $-3.7711e+06$ | 0.719 | 6 | $-3.7711e+06$ | 0.741 | 5 |
| 10 | 1000 | $-6.1447e+09$ | 11.657 | 8 | $-6.1447e+09$ | 11.766 | 7 | $-6.1447e+09$ | 10.822 | 8 |

## 5 Conclusion

This paper introduced a novel and practical revised quasi-Newton's method for solving unconstrained minimization problems, and we conducted a convergence analysis. Subsequently, we tested our algorithm on various unconstrained problems of small and medium dimensions. Our algorithm demonstrated an improvement in the step length compared to the Broyden quasi-Newton's method. The numerical results and the comparisons with the Broyden quasi-Newton's method provide evidence of the efficiency and robustness of our algorithm.

**Appendix: Test Problems.**

1. Extended Freudenstein & Roth function:

$$\begin{aligned} f(x) &= \sum_{i=1}^{\frac{n}{2}} \left(-13 + x_{2i-1} + \left((5 - x_{2i})x_{2i} - 2\right) x_{2i}\right)^2 \\ &\quad + \left(-29 + x_{2i-1} + \left((x_{2i} + 1)x_{2i} - 14\right) x_{2i}\right)^2, \\ x_0 &= [0.5, -2, 0.5, -2, \ldots, 0.5, -2]. \end{aligned}$$

2. Extended Tridiagonal one function:

$$\begin{aligned} f(x) &= \sum_{i=1}^{\frac{n}{2}} \left(x_{2i-1} + x_{2i} - 3\right)^2 + \left(x_{2i-1} - x_{2i} + 1\right)^4, \\ x_0 &= [2, 2, \ldots, 2]. \end{aligned}$$

3. Generalized PSC1 function:

$$\begin{aligned} f(x) &= \sum_{i=1}^{n-1} \left(x_i^2 + x_{i+1}^2 + x_i x_{i+1}\right)^2 + \sin^2(x_i) + \cos^2(x_i), \\ x_0 &= [3, 0.1, 3, 0.1, \ldots, 3, 0.1]. \end{aligned}$$

4. Extended Powell function:

$$\begin{aligned} f(x) &= \sum_{i=1}^{\frac{n}{4}} \left(x_{4i-3} + 10x_{4i-2}\right)^2 + 5\left(x_{4i-1} - x_{4i}\right)^2 \\ &\quad + \left(x_{4i-2} - 2x_{4i-1}\right)^4 + 10\left(x_{4i-3} - x_{4i}\right)^4, \\ x_0 &= [3, -1, 0, 1, \ldots, 3, -1, 0, 1]. \end{aligned}$$

5. Extended Maratos function:

$$\begin{aligned} f(x) &= \sum_{i=1}^{\frac{n}{2}} x_{2i-1} + c\left(x_{2i-1}^2 + x_{2i}^2 - 1\right)^2, \\ x_0 &= [1.1, 0.1, \ldots, 1.1, 0.1], \quad c = 100. \end{aligned}$$

6. Extended Cliff function:

$$f(x) = \sum_{i=1}^{\frac{n}{2}} \left( \frac{x_{2i-1} - 3}{100} \right)^2 - (x_{2i-1} - x_{2i}) + exp\left(20(x_{2i-1} - x_{2i})\right),$$
$$x_0 = [0, -1, \ldots, 0, -1].$$

7. Extended quadratic penalty QP1 function:

$$f(x) = \sum_{i=1}^{n-1} \left(x_i^2 - 2\right)^2 + \left( \sum_{i=1}^{n} x_i^2 - 0.5 \right)^2,$$
$$x_0 = [1, 1, \ldots, 1].$$

8. SINQUAD function (CUTE):

$$f(x) = (x_1 - 1)^4 + \sum_{i=2}^{n-1} \left(\sin(x_i - x_n) - x_1^2 + x_i^2\right)^2 + \left(x_n^2 - x_1^2\right)^2,$$
$$x_0 = [0.1, 0.1, \ldots, 0.1].$$

9. RMDFSINE function:

$$f(x) = \sum_{i=1}^{\frac{n}{2}} \sin\left(-0.5x_{2i} + x_{2i-1}^2\right),$$
$$x_0 = [1, 1, \ldots, 1].$$

10. Diagonal nine function:

$$f(x) = \sum_{i=1}^{n-1} (exp(x_i) - ix_i) + 10000x_n^2,$$
$$x_0 = [1, 1, \ldots, 1].$$

**References**

[1] Andrei, N. (2008). "An unconstrained optimization test functions collection", Advanced Modeling and Optimization, 10(1), 147-161.

[2] Broyden, C. (1965). "A class of methods for solving nonlinear simultaneous equations", Mathematics of Computation, 19, 577-593.

[3] Dehghan, T., Niri, M., Hosseini, M., Heydari, M. (2019). "An efficient improvement of the Newton's method for solving non-convex optimization problems", Computational Methods for Differential Equations, 7(1), 69-85.

[4] Dennis, J.E., Schnabel, R.B. (1993). "Numerical methods for unconstrained optimization and nonlinear equations", SIAM, Philadelphia.

[5] Fang, X.W., Ni, Q., Zeng, M.L. (2018). "A modified quasi-Newton's method for nonlinear equations", Journal of Computational and Applied Mathematics, 328, 44-58.

[6] Li, D.H., Fukushima, M., Qi, L., Yamashita, N. (2004). "Regularized Newton methods for convex minimization problems with singular solutions", Computational Optimization and Applications, 28, 131-147.

[7] Narcowich, F.J. (2023). "The Rank of a Matrix", https://www.math.tamu.edu/~fnarc/psfiles/rank2005.pdf.

[8] Polyak, R.A. (2009). "Regularized Newton's method for unconstrained convex optimization", Mathematical Programming, 120, 125-145.

[9] Sa, H., Chen, G.Q., Sui, Y.K., Wu, C.Y. (2016). "A new newton-like method for solving nonlinear equations", Springer Plus., 5, 12-69.

[10] Shen, Ch., Xiongda, Ch., Liang, Y. (2012). "A regularized Newton's method for degenerate unconstrained optimization problems", Optimization Letters, 6, 1913-1933.

[11] Song, W.u., Haijun, W. (2020). "A modified Newton-like method for nonlinear equations", Computational and Applied Mathematics, 39, 238.

[12] Sui, Y., Sa, H., Chen, G. (2014). "An improvement for the rational approximation RALND at accumulated two-point information", Mathematica Numerica Sinica, 36, 51-64.

[13] Ueda, K., Yamashita, N. (2010). "Convergence properties of the regularized Newton's method for the unconstrained non-convex optimization", Applied Mathematics and Optimization, 62, 27-46.

[14] Yuan, G., Wang, Zh., Li, P. (2022). "Global convergence of a modified Broyden family method for non-convex functions", Journal of Industrial and Management Optimization, 18(6), 4393-4407.

[15] Zhou, G., Toh, K.C. (2005). "Superlinear convergence of a Newton-type algorithm for monotone equations", Journal of Optimization Theory and Applications, 125, 205-221.