Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(6): 14893–14916.
DOI: 10.3934/math.2025668
Received: 28 February 2025
Revised: 23 June 2025
Accepted: 24 June 2025
Published: 30 June 2025

Research article

A structured RMIL conjugate gradient-based strategy for nonlinear least squares with applications in image restoration problems

Rabiu Bashir Yunus[1,*], **Ahmed R. El-Saeed**[2], **Nooraini Zainuddin**[1] **and Hanita Daud**[1]

[1] Department of Fundamental and Applied Sciences, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, Bandar Seri Iskandar 32610, Perak Darul Ridzuan, Malaysia

[2] Department of Mathematics and Statistics, Faculty of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia

* **Correspondence:** Email: rabiu_22000788@utp.edu.my.

Abstract: Numerous scientific and technical domains have found use for nonlinear least squares (NLS). Conventional approaches to NLS problem solving often suffer from computational inefficiencies and high memory requirements, particularly when applied to large-scale systems. This paper presents the structured conjugate gradient coefficient using a structured secant-like approximation to solve NLS problems. The approach develops a structured vector approximation that captures the vector-matrix relationship using Taylor series expansions of the Hessian of the goal function. It is possible to incorporate more Hessian information into the traditional search direction, since this approximation satisfies a quasi-Newton condition. Furthermore, given conventional assumptions, a global convergence study is performed. Benchmark NLS tasks are used for numerical studies to assess the method's performance against alternative approaches. The results demonstrate the promise and success of the approach. Lastly, problems with image restoration are addressed using the suggested technique.

Keywords: structured; nonlinear; least squares; conjugate gradient; image restoration; optimization; sufficient descent
Mathematics Subject Classification: 65K05, 90C52, 90C53, 90C56

1. Introduction

Nonlinear least squares (NLS) is a mathematical optimization technique that is used to solve problems where the relationship between variables is nonlinear. In many real-world applications, NLS plays a crucial role, as it enables the fitting of complex models to the observed data, thereby estimating parameters to minimize the difference between the predicted values and the observed data.

It has been widely adopted as a mathematical optimization tool in science and engineering. Applications span various areas, including machine learning; it is commonly applied in training models such as linear and logistic regressions, where it helps minimize the loss function without computing matrix inverses [1]. In artificial intelligence (AI), including training AI models such as neural networks, conjugate gradient (CG) can be used as an alternative to gradient descent for faster convergence in smaller or batch-based problems. In addition, NLS has been applied in reinforcement learning, especially in policy optimization methods that rely on second-order information [2]. It helps improve accuracy in tasks such as sparse signal reconstruction [3], robotic motion control [4], and computer vision [5]. NLS problems can be described as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(x), \tag{1.1}$$

where the definition of the function $f(x)$ is

$$f(x) = \frac{1}{2}\|u(x)\|^2 = \frac{1}{2}\sum_{i=1}^{m}(u_i(x))^2, x \in \mathbb{R}^n. \tag{1.2}$$

Given $u(x)$ as $(u_1(x), u_2(x), \cdots, u_m(x))^T$, where each individual residual $u_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, 2, \cdots, and\ m$ is a smooth function, we define $J(x) \in \mathbb{R}^{m\times n}$ as the Jacobian matrix associated with the residual function $u(x)$. Furthermore, we use $\nabla f(x)$ to represent the gradient of the objective function and $\nabla^2 f(x)$ to represent the Hessian of the same objective function [6,7]. The Jacobian of the residual $u(x)$ is given by the following:

$$J(x) = \begin{bmatrix} \frac{\partial u_1}{\partial x_1}(x) & \frac{\partial u_1}{\partial x_2}(x) & \cdots & \frac{\partial u_1}{\partial x_n}(x) \\ \frac{\partial u_2}{\partial x_1}(x) & \frac{\partial u_2}{\partial x_2}(x) & \cdots & \frac{\partial u_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_m}{\partial x_1}(x) & \frac{\partial u_m}{\partial x_2}(x) & \cdots & \frac{\partial u_m}{\partial x_n}(x) \end{bmatrix}. \tag{1.3}$$

The gradient of $f(x)$ is given by

$$g(x) = \sum_{i=1}^{m} u_i(x)\nabla u_i(x) = J(x)^T u(x), \tag{1.4}$$

and the Hessian is given by

$$G(x) = \underbrace{\sum_{i=1}^{m} \nabla u_i(x)\nabla u_i(x)^T}_{\text{First part}} + \underbrace{\sum_{i=1}^{m} u_i(x)\nabla^2 u_i(x)}_{\text{Second part}} = J(x)^T J(x) + T(x). \tag{1.5}$$

where

$$T(x) = \sum_{i=1}^{m} u_i(x)\nabla^2 u_i(x). \tag{1.6}$$

It is crucial to note that the Hessian matrix $G(x)$ in the second term of (1.5) is a symmetric $n \times n$ matrix. The Jacobian matrix and its transpose, which are obtained from the first partial derivatives of

$f(x)$ (the linear part), make up the first component of this matrix, which is made up of two separate parts. The tensor terms, which represent the nonlinear component, are captured by the second component, $T(x)$. The sum of products involving $u_i(x)$ and $\nabla^2 u_i(x)$, which contain m and $n \times n$ components, respectively, produces these tensor terms $T(x)$ [8]. The first and second-order approaches are the two general groups into which NLS techniques fall. The computations of first-order methods, including the Gauss-Newton method, the Levenberg-Marquardt method, and the Steepest Descent (SD) method, are based on the first derivative of the objective function. Conversely, second-order approaches either use estimates of the Hessian matrix or incorporate the Hessian matrix of the objective function. The Newton method, the quasi-Newton method (QN), and the trust-region approach (TR) are examples of second-order techniques [8,9].

Furthermore, NLS problems are a specific type of unconstrained optimization problem [9]. As a result, using general unconstrained minimization techniques, a series of iterates can be produced through the following:

$$x_{k+1} = x_k + \alpha_k d_k. \tag{1.7}$$

In the direction of the search vector d_k, the step sizes $\alpha_k > 0,\ k \geq 0$ are obtained using a line search scheme.

The search direction d_k is generated using the CG method by the following formula:

$$d_k = \begin{cases} -g_k & if \quad k = 0 \\ -g_k + \beta_k d_{k-1} & if \quad k \geq 1 \end{cases} \tag{1.8}$$

where β_k is the CG coefficient or the CG parameter, and $g_k = \nabla f(x_k)^T$ denotes the gradient of the function f. However, the selection of β_k significantly affects the computation of CG methods, thus prompting extensive research into the effective choices of the parameters. For specific examples, refer to [10]. Furthermore, the direction d_k described in Eq (1.8) is usually required to satisfy the sufficient descent condition, which stipulates that there must be a constant $c > 0$ such that

$$g_k^T d_k \leq -c\|g_k\|^2.$$

Gradient-based methods are fundamental tools for solving optimization problems, particularly in the context of minimizing nonlinear functions. These methods play a critical role in applications such as signal processing and system identification due to their computational efficiency and scalability. Recent advances have further enhanced their robustness and convergence properties in practical settings. For example, improved gradient-based techniques have been successfully applied in adaptive control systems [11] and high-dimensional optimization problems that involve partial differential equations [12]. Hestenes and Stiefel [13] (HS), Polak and Ribiere (1969) [14], Polyak [15] (PRP), and Liu and Storey [16] (LS) introduced many classical nonlinear conjugate gradient algorithms that made use of the following β_k updating formula:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad \beta_k^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad \beta_k^{LS} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}, \tag{1.9}$$

where $y_k = g_{k+1} - g_k$, and $\|.\|$ denotes the L-2 norm. Although these classical conjugate gradient formulas are computationally efficient, they may occasionally fail to ensure global convergence for the arbitrary functions of Hager and Zhang [17]. For example, Powell highlighted problems with the PRP formula, which cycles without reaching an optimum, even when line search methods are employed by Yao et al. [18]. The second category of classical methods includes Fletcher and Reeves (FR) [19], Fletcher's conjugate descent (CD) [20], and Dai and Yuan (DY) [21], each associated with the following CG parameters:

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \quad \beta_k^{CD} = \frac{\|g_{k+1}\|^2}{-d_k^T g_k}, \quad \beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k}. \tag{1.10}$$

When employing an exact line search, the Rivaie, Mustafa, Ismail, and Leong (RMIL) formula exhibits strong global convergence properties [22]. However, its numerical performance is notably inferior to that of the PRP formula. Furthermore, the authors in [22] stated that β_k^{RMIL} satisfies the following condition:

$$0 < \beta_k \leq \frac{\|g_k\|^2}{\|d_{k-1}\|^2}, \tag{1.11}$$

which they utilized to establish the global convergence of their algorithm. Wu et al. [23] extended this concept to develop a variant of RMIL, defined as

$$\beta_k^{VRMIL} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|d_{k-1}\|} g_k^T d_{k-1}}{\max\{\mu \|g_k\| \|d_{k-1}\|, \|d_{k-1}\|^2\}}, \tag{1.12}$$

and demonstrated that the proposed formula satisfies the well-known sufficient descent condition regardless of the line search strategy employed. Moreover, they proved that the method achieves global convergence under mild assumptions. However, Dai [24] revealed that the RMIL method does not always satisfy condition (1.11), and proposed an improved version, RMIL+, which is given by the following:

$$\beta_k^{RMIL+} = \begin{cases} \dfrac{g_k^T y_{k-1}}{\|d_{k-1}\|^2}, & \text{if} \quad 0 \leq g_k^T g_{k-1} \leq \|g_k\|^2, \\ 0, & \text{otherwise.} \end{cases} \tag{1.13}$$

To enhance the computational efficiency of RMIL+, Yousif and Saleh [25] modified this formulation by replacing $g_k^T g_{k-1}$ with its absolute value, thereby defining the new CG coefficient as follows:

$$\beta_k^{RMIL*} = \begin{cases} \dfrac{g_k^T y_{k-1}}{\|d_{k-1}\|^2}, & \text{if} \quad 0 \leq |g_k^T g_{k-1}| \leq \|g_k\|^2, \\ 0, & \text{otherwise.} \end{cases} \tag{1.14}$$

This modification effectively doubles the permissible interval length of (1.14) compared to (1.13), thus enhancing the computational efficiency of the method. The authors further established the global convergence of (1.14) under the exact line search conditions. As a result, several related studies [26] have explored variants of RMIL that conform to conditions (1.11). Recently, Zhang and Ding [27] introduced adaptive methods based on the integer order and proposed a fractional order gradient optimization approach, which resulted in a novel fractional order stochastic gradient filtering algorithm with improved performance. Structured quasi-Newton (SQN) methods have gained significant attention for efficiently solving nonlinear optimization problems, particularly when partial information about the Hessian matrix is available. Zhang et al. [28] proposed a new SQN algorithm that incorporates partial Hessian information to improve the quality of the approximation while maintaining a low computational cost. Similarly, Chen et al. [29] developed a modified quasi-Newton method tailored to structured optimization problems, using known Hessian structures to improve convergence behavior and solution accuracy. This important contribution offers new research directions for both quadratic and non-linear optimization problems. Earlier foundational work by Dennis et al. [30] established a convergence theory for the structured Broyden-Fletcher-Goldfarb-Shanno (BFGS) method and demonstrated its effectiveness for NLS problems by aligning secant updates with the problem structure. Extending this line of research, Amini et al. [31] introduced a structured secant method based on partial information from the Hessian. This study proposes a structured CG-based method based on the RMIL approach, thereby integrating second-order information to ensure a downward search direction. The following is a summary of the main contributions of this study:

- Building upon the RMIL framework, we developed a structured RMIL CG algorithm with a modified version of the structured secant equation;
- The suggested search direction satisfies the sufficient descent condition;
- Under specific assumptions, we established the global convergence of the method, facilitated by a nonmonotone line search strategy;
- Numerical experiments were conducted to assess the effectiveness of the suggested approach in comparison to existing methods.

The article is structured as follows: Section 2 outlines the proposed methodology and the corresponding algorithm; Section 3 analyzes the global convergence properties of the proposed method; Section 4 presents numerical experiments to evaluate the computational efficiency of the proposed algorithm compared to other methods; and in Section 5, we discuss the application of the proposed method to image restoration problems.

2. The structured RMIL conjugate gradient-based method

This section presents the conceptual framework underlying the proposed structured RMIL CG-based method. It begins with the derivation of the structured secant formula, followed by a discussion of the motivation behind the formulation of the SRMILCG parameter. In addition, an algorithm is provided that outlines the steps involved in the implementation of the method.

2.1. Structured secant formula derivation

Initially, we approximate the product of the matrix $T(x)$ and the vector v. This operation can be understood as an approximation of the influence of $\nabla^2 u_i(x)$ on a given vector such as $v \in \mathbb{R}^n$. This enables the derivation of essential data from $T(x)$.

Consequently, the required approximation can be obtained by post-multiplying $T(x) = \sum_{i=1}^{m} u_i(x_k)\nabla^2 u_i(x_k)$ in (1.6) by s_{k-1}, that is,

$$T(x_k)s_{k-1} = \sum_{i=1}^{m} u_i(x_k)\nabla^2 u_i(x_k)s_{k-1}. \tag{2.1}$$

To simplify the notation, we introduce the following definition:

$$\begin{aligned} T_k &= T(x_k), \\ u_k^i &= u_i(x_k), \\ \nabla^2 u_k^i &= \nabla^2 u_i(x_k) = H_k^i, \\ &\text{and gradients of components-wise} \\ &u_k^i \text{ by } g_k^i \text{ for } i = 1, 2, \ldots, m. \end{aligned}$$

Our goal is to estimate $\nabla^2 u_k^i s_{k-1}$. This can be accomplished by using the expansion of the Taylor series of u_k^i and g_k^i as outlined below:

$$u_{k-1}^i \approx u_k^i + g_k^{iT} s_{k-1} + \frac{1}{2!} s_{k-1}^T H_k^i s_{k-1}. \tag{2.2}$$

Similarly, we obtain the following:

$$g_{k-1}^i \approx g_k^i - H_k^i s_{k-1}. \tag{2.3}$$

To achieve the desired result, we first pre-multiply (2.3) by s_{k-1}.

$$s_{k-1}^T g_{k-1}^i \approx s_{k-1}^T g_k^i - s_{k-1}^T H_k^i s_{k-1}. \tag{2.4}$$

Since both (2.2) and (2.3) contain the term $s_{k-1}^T H_k^i s_{k-1}$ in their expression, we can rewrite Eqs (2.2) and (2.4) in terms of this expression, then equate and simplify the result to obtain the following:

$$s_{k-1}^T H_k^i s_{k-1} \approx 2(u_{k-1}^i - u_k^i) + (g_k^i + g_{k-1}^i)^T s_{k-1} + s_{k-1}^T (g_k^i - g_{k-1}^i).$$

If a simple diagonal approximation is considered for $H_{k,i}$, then

$$s_{k-1}^T H_k^i s_{k-1} \approx \zeta I \|s_{k-1}\|^2 \approx 2(u_{k-1}^i - u_k^i) + (g_k^i + g_{k-1}^i)^T s_{k-1} + s_{k-1}^T (g_k^i - g_{k-1}^i).$$

Therefore, the approximation of $H_{k,i} s_{k-1}$ is as follows:

$$H_k^i s_{k-1} \approx \frac{[2(u_{k-1}^i - u_k^i) + (g_k^i + g_{k-1}^i)^T s_{k-1} + s_{k-1}^T (g_k^i - g_{k-1}^i)]}{\|s_{k-1}\|^2} s_{k-1}. \tag{2.5}$$

Consequently, adding up all of i after inserting (2.5) in (1.3) yields the following:

$$H_k s_{k-1} \approx \frac{[2(u_{k-1} - u_k) + (J_k + J_{k-1})^T s_{k-1} + s_{k-1}^T (J_k - J_{k-1})]}{\|s_{k-1}\|^2} s_{k-1} = z_{k-1}.$$

Thus

$$\vartheta_{k-1} = u_k^T \left[2(u_{k-1} - u_k) + (J_k + J_{k-1})^T s_{k-1} + s_{k-1}^T (J_k - J_{k-1}) \right]. \tag{2.6}$$

We assume that the Hessian in (1.6) is approximated in a manner that satisfies a structured secant criterion, as given by the following:

$$G_k s_{k-1} \approx \bar{z}_{k-1}, \tag{2.7}$$

As a result, we derive the following:

$$\bar{z}_{k-1} = J_k^T J_k s_{k-1} + \frac{\vartheta_{k-1}}{\|s_{k-1}\|^2} s_{k-1}. \tag{2.8}$$

2.2. Motivation and the proposed SRMILCG formula

Motivated by the research conducted by [22],

$$\beta_k^{RMIL} = \frac{g_k^T y_{k-1}}{d_{k-1}^T (d_{k-1} - g_k)}. \tag{2.9}$$

We present the structured RMIL method, and y_{k-1} is replaced by $\bar{z}_{k-1}$ in the numerator, while the denominator is simplified as $d_{k-1}^T (d_{k-1} - g_k) = \|d_{k-1}\|^2 - d_{k-1}^T g_k$. For an exact line search, $d_{k-1}^T g_k = 0$. Therefore, the formula is given by the following:

$$\beta_k^{SRMILCG} = \frac{g_k^T \bar{z}_{k-1}}{\|d_{k-1}\|^2}, \tag{2.10}$$

where $\bar{z}_{k-1} = J_k^T J_k s_{k-1} + \frac{\vartheta_{k-1}}{\|s_{k-1}\|^2} s_{k-1}$. The proposed direction is defined by the following:

$$d_k = -g_k + \beta_k^{SRMILCG} d_{k-1}, \forall\, k \geq 1. \tag{2.11}$$

Additionally, this study utilizes the nonmonotone line search strategy introduced by Zhang and Hager [32] to determine the α_k. In particular, when the search direction d_k exhibits a significant descent, the nonmonotone line search criteria listed below are utilized to determine α_k:

$$f(x_k + \alpha_k d_k) \leq Q_k + \delta \alpha_k g_k^T d_k, \tag{2.12}$$

where,

$$\begin{cases} Q_0 = f(x_0), \\ Q_{k+1} = \frac{\eta_k P_k Q_k + f(x_{k+1})}{P_{k+1}}, \\ P_0 = 1, \\ P_{k+1} = \eta_k P_k + 1. \end{cases} \tag{2.13}$$

Equations (2.12) and (2.13) describe a nonmonotone line search adopted from the work of Zhang and Hager [32]. The parameter δ in (2.13) is a small positive constant used as a sufficient decrease parameter; in our case, we used $\delta = 0.01$. Additionally, η_k in (2.13) is a parameter typically between 0 and 1 that controls how much the method allows the function values to temporarily increase.

Remark 2.1. *The sequence P_k is located between $f(x_k)$ and ψ_k, where*

$$\psi_k := \frac{1}{k+1} \sum_{i=0}^{k} f(x_i), \; k \geq 0. \tag{2.14}$$

Remark (2.1) clarifies that P_k is not arbitrarily chosen, but strategically bounded between the value of the current function and the average of the previous ones. This ensures a controlled nonmonotonic descent, which is central to the convergence of Zhang's nonmonotone line search method [33]. Moreover, the steps for the SRMILCG approach are presented in Algorithm 1.

Algorithm 1 Structured RMIL CG-based method (SRMILCG)

```
1: **Input:** Initial point $x_0 \in \mathbb{R}^n$, tolerance $\epsilon > 0$, parameters $\delta, \sigma \in (0, 1)$,
2:         bounds $0 \leq \eta_{\min} \leq \eta_{\max} \leq 1$, and maximum iterations $k_{\max} \in \mathbb{N}$
3: **Initialize:** $k := 0$
4: Compute gradient $g_k := \nabla f(x_k)$
5: **if** $||g_k|| \leq \epsilon$ **then**
6:     **return** $x_k$
7: **end if**
8: Set search direction $d_k := -g_k$
9: **while** $k < k_{\max}$ **do**
10:     Compute step size $\alpha_k$ using the non-monotone line search (2.12)
11:     Update iterate: $x_{k+1} := x_k + \alpha_k d_k$
12:     Compute gradient: $g_{k+1} := \nabla f(x_{k+1})$
13:     **if** $||g_{k+1}|| \leq \epsilon$ **then**
14:         **return** $x_{k+1}$
15:     **end if**
16:     Compute $\beta_k^{\text{SRMILCG}}$ using (2.10), with $\bar{z}_{k-1}$ from (2.8)
17:     Update direction: $d_{k+1} := -g_{k+1} + \beta_k^{\text{SRMILCG}} d_k$
18:     Set $k := k + 1$
19: **end while**
20: **return** $x_k$
```

3. Convergence analysis

Under specific assumptions, this section provides the convergence analysis of the proposed SRMIL methods, using Eq (2.10). Throughout this discussion, it is assumed that $g_k \neq 0$ for all k. The following common presumptions regarding the objective function are considered.

Assumption 1. The level set is as follows:

$$\ell = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}.$$

It is contained within a bounded region. Specifically, there exists a positive constant $\bar{\Gamma}$ such that

$$||x|| \leq \bar{\Gamma}, \quad \forall x \in \ell.$$

Assumption 2. The residual $u(x)$, and its derivative, that is, the Jacobian matrix, $J(x)$, are Lipschitz continuous in some open neighborhood N of ℓ, that is, $||J(x) - J(y)|| \leq L||x - y||$, $||u(x) - u(y)|| \leq b_1||x - y||$, $||g(x) - g(y)|| \leq c_1||x - y||$, $||J(x)|| \leq a_2$, and $||u(x)|| \leq b_2$, $\forall x, y \in \ell$, where L, b_1, c_1, a_2, and b_2 are positive constants, and $||J(x)||$ denotes the matrix norm induced by the Euclidean vector norm (i.e., the spectral norm). We now present the following important lemma.

Lemma 3.1. *Suppose that the first and second assumptions are satisfied. Let Algorithm 1 generate the sequences $\{x_k\}$ and $\{d_k\}$. Consequently, a constant $M > 0$ exists such that for each $k \geq 0$, the following is true:*

$$||\bar{z}_{k-1}|| \leq M||s_{k-1}||. \tag{3.1}$$

Proof. From Eq (2.8), we have the following:

$$\bar{z}_{k-1} = J_k^T J_k s_{k-1} + \frac{\vartheta_{k-1}}{||s_{k-1}||^2} s_{k-1}$$

where,

$$\vartheta_{k-1} = u_k^T \left[2(u_{k-1} - u_k) + (J_k + J_{k-1})^T s_{k-1} + s_{k-1}^T (J_k - J_{k-1})\right].$$

The given expression can be reformulated as follows:

$$\vartheta_{k-1} = u_k^T \left[s_{k-1}^T (J_k - J_{k-1}) + [J_k s_{k-1} - (u_k - u_{k-1}) + J_{k-1} s_{k-1} - (u_k - u_{k-1})]\right],$$

$$\begin{aligned}
|\vartheta_{k-1}| &= |u_k^T \left[s_{k-1}^T (J_k - J_{k-1}) + [J_k s_{k-1} - (u_k - u_{k-1}) + J_{k-1} s_{k-1} - (u_k - u_{k-1})]\right]| \\
&\leq ||u_k||||s_{k-1}(J_k - J_{k-1})|| + ||\left[-(J_k s_{k-1} - (u_k - u_{k-1})) + (J_{k-1} s_{k-1} - (u_k - u_{k-1}))\right]|| \\
&\leq b_2 ||J_k - J_{k-1}||||s_{k-1}|| + ||J_k s_{k-1} - (r_k - r_{k-1})|| + ||J_{k-1} s_{k-1} - (u_k - u_{k-1})|| \\
&\leq L b_2 ||s_{k-1}||^2 + \int_0^1 ||J_k - J(x_k + t(x_{k-1} - x_k))|| \, dt ||x_{k-1} - x_k|| \\
&\quad + \int_0^1 ||J_{k-1} - J(x_{k-1} + t(x_k - x_{k-1}))|| \, dt ||x_k - x_{k-1}|| \\
&\leq L b_2 ||s_{k-1}||^2 + \left[\frac{L}{2}||s_{k-1}||^2 + \frac{L}{2}||s_{k-1}||^2\right] \\
&= (L b_2 + L)||s_{k-1}||^2.
\end{aligned}$$

Hence,

$$\begin{aligned}
||z_{k-1}^{A3TSCG}|| &= ||J_k^T J_k s_{k-1} + \frac{\vartheta_{k-1}}{||s_{k-1}||^2} s_{k-1}|| \\
&\leq a_2^2 ||s_{k-1}|| + \frac{(L b_2 + L_1)||s_{k-1}||^3}{||s_{k-1}||^3} \\
&= (a_2^2 + L b_2 + L)||s_{k-1}||.
\end{aligned}$$

Therefore, by setting, $M := (a_2^2 + L b_2 + L)$, the inequality holds. □

The lemma that follows shows that, the sufficient descent requirement is met by the direction d_k, independent of the line search requirement.

Lemma 3.2. *Let* (2.10) *be used to define $\beta_k^{SRMILCG}$, and let Algorithm 1 be used to construct the sequence x_k. Eq* (2.11) *defines the search direction d_k, which exhibits the following inequality for every $k \geq 0$ when $c > 0$:*

$$g_k^T d_k \leq -c||g_k||^2. \tag{3.2}$$

Proof. It follows for $k = 0$ that

$$g_0^T d_0 = -g_0^T g_0 = -\|g_0\|^2.$$

If $k \geq 1$, then it follows that

$$\begin{aligned} g_k^T d_k =& g_k^T(-g_k + \beta_k^{SRMILCG} d_{k-1}), \\ =& -g_k^T g_k + \frac{g_k^T \bar{z}_{k-1}}{\|d_{k-1}\|^2} g_k^T d_{k-1}, \\ \leq& -\|g_k\|^2 + \frac{|g_k^T \bar{z}_{k-1}|}{\|d_{k-1}\|^2} |g_k^T d_{k-1}|, \\ \leq& -\|g_k\|^2 + \frac{\|g_k\|\|\bar{z}_{k-1}\|}{\|d_{k-1}\|^2} \|g_k\|\|d_{k-1}\|, \\ \leq& -\left(1 - \frac{M\|s_{k-1}\|}{\|d_{k-1}\|}\right)\|g_k\|^2. \end{aligned}$$

Thus, by defining c as

$$c := \left(1 - \frac{M\bar{\Gamma}}{\gamma}\right),$$

we obtain $g_k^T d_k \leq -c\|g_k\|^2$. The proof is now complete. □

Lemma 3.3. *Assume that Algorithm 1 defines the sequences $\{x_k\}$ and $\{d_k\}$. For every $k \geq 1$, there is $\kappa > 0$ such that*

$$\|d_k\| \leq \kappa\|g_k\|. \tag{3.3}$$

Proof. If $k = 0$, then $\|d_0\| = \|-g_0\| = \|g_0\|$.

If $k \geq 1$, then it follows that

$$\begin{aligned} \|d_k\| &= \left\|-g_k + \beta_k^{SRMILCG} d_{k-1}\right\|, \\ &\leq \|g_k\| + |\beta_k^{SRMILCG}|\|d_{k-1}\|, \end{aligned} \tag{3.4}$$

Now, by taking the modulus of the parameter β_k in Eq (2.10), we have the following:

$$|\beta_k^{SRMILCG}| = \left|\frac{g_k^T \bar{z}_{k-1}}{\|d_{k-1}\|^2}\right| \leq \frac{\|g_k\|\|\bar{z}_{k-1}\|}{\|d_{k-1}\|^2} \leq \frac{M\|g_k\|\|s_{k-1}\|}{\|d_{k-1}\|^2}. \tag{3.5}$$

Substituting (3.5) into (3.4), we get the following:

$$\begin{aligned} \|d_k\| &\leq \|g_k\| + \frac{M\|g_k\|\|s_{k-1}\|}{\|d_{k-1}\|^2}\|d_{k-1}\| \\ &\leq \|g_k\| + \frac{M\|g_k\|\|s_{k-1}\|}{\|d_{k-1}\|} \\ &= \left(1 + \frac{M\|s_{k-1}\|}{\|d_{k-1}\|}\right)\|g_k\|. \end{aligned}$$

Therefore, by defining κ as

$$\kappa := \left(1 + \frac{M\bar{\Gamma}}{\gamma}\right),$$

we obtain $\|d_k\| \leq \kappa$. □

Lemma 3.4. *Suppose Assumption 1 is satisfied; if the iterative sequence $\{x_k\}$ is generated by Algorithm 1, then it can be established that $f_k \leq Q_k$ for every value of k.*

Proof. Let $t \geq 0$ and define $\Phi : \mathbb{R} \to \mathbb{R}$ by

$$\Phi_k(t) = \frac{tQ_{k-1} + f(x_k)}{t+1}. \tag{3.6}$$

Differentiating Eq (3.6) with respect to t yields the following:

$$\frac{d\Phi_k(t)}{dt} = \frac{Q_{k-1} - f(x_k)}{(t+1)^2}. \tag{3.7}$$

By the relation $g_k^T d_k \leq -c\|g_k\|^2$, $\forall k$, we have from (2.12) that

$$f(x_k) = f(x_{k-1} + \alpha_k d_{k-1}) \leq Q_{k-1} + \delta\alpha_k(g_{k-1}^T d_{k-1}) \leq Q_{k-1} - \kappa_1\delta\alpha_k\|g_{k-1}\|^2 \leq Q_{k-1}.$$

This implies that for all $t \geq 0$, $\frac{d\Phi_k(t)}{dt} \geq 0$, which means that $\Phi_k(t)$ is not decreasing. Therefore, Eq (3.6) satisfies $f(x_k) = \Phi_k(0) \leq \Phi_k(t)$, for all $t \geq 0$. Specifically, by taking $t = \eta_{k-1}P_{k-1}$, we obtain the following:

$$f(x_k) = \Phi_k(0) \leq \Phi_k(\eta_{k-1}P_{k-1}) = \frac{\eta_{k-1}P_{k-1}Q_{k-1} + f(x_k)}{\eta_{k-1}P_{k-1} + 1} \tag{3.8}$$

$$= \frac{\eta_{k-1}P_{k-1}Q_{k-1} + f(x_k)}{P_k} \quad \text{(from (2.13))} \tag{3.9}$$

$$= Q_k. \tag{3.10}$$

Therefore, the lower bound of Q_k is determined. Next, we demonstrate $Q_k \leq \psi_k$ by induction. Let $k = 0$; from equation ((2.12)) we have $Q_0 = \psi_0 = f(x_0)$. Now, assume $Q_j \leq \psi_j$ for all $0 \leq j < k$. Given that $\eta_k \in [0, 1]$ and $P_0 = 1$ by Eq (2.12), we obtain the following:

$$P_{j+1} = 1 + \sum_{i=0}^{j} \prod_{l=0}^{i} \eta_{j-l} + j + 2. \tag{3.11}$$

Combining (3.6), (3.8) and (3.11), we have the following:

$$\begin{aligned} Q_k &= \Phi_k(P_k - 1) \\ &= \Phi_k(\eta_{k-1}P_{k-1}) \\ &= \Phi_k\left(\sum_{i=0}^{k-1} \prod_{n=0}^{i} \eta_{k-n-1}\right) \\ &\leq \Phi_k(k). \end{aligned}$$

Using the induction step, we obtain the following:

$$\Phi(k) = \frac{kQ_{k-1} + f(x_k)}{k+1} \leq \frac{k\psi_{k-1} + f(x_k)}{k+1} = \psi_k.$$

Hence, it holds that $Q_k \leq \psi_k$. □

Lemma 3.5. *Suppose that Assumption 2 is satisfied; if the sequence $\{x_k\}$ is generated by SRMILCG, it can be deduced that*

$$\alpha_k \geq \left(\frac{2(1-\delta)}{c_1\zeta}\right)\frac{|g_k^T d_k|}{\|d_k\|^2}. \tag{3.12}$$

Remark 3.6. *If the starting condition $P_0 = 1$ and taking into account that $\eta_k \in [0,1]$, if $Q_j \leq \psi_j$ for all $0 \leq j < k$, then we obtain the following:*

$$P_{j+1} = 1 + \sum_{i=0}^{m}\prod_{m=0}^{i}\eta_{j-m} \leq j+2. \tag{3.13}$$

Theorem 3.7. *Suppose that Eq (1.1) defines the function $f(x)$, and Assumptions 1 and 2 are satisfied. The resulting sequence $\{x_k\}$ generated by SRMILCG algorithms is encompassed by the level set ℓ, and*

$$\lim_{k\to\infty}\inf\|g_k\| = 0. \tag{3.14}$$

Moreover, if η_{max} is less than 1, then

$$\lim_{k\to\infty}\|g_k\| = 0. \tag{3.15}$$

Proof. To begin with, we present the fact that

$$f_{k+1} \leq Q_k - \beta\|g_k\|^2, \tag{3.16}$$

where,

$$\beta = \frac{2\delta(1-\delta)c^2}{c_1\zeta\kappa^2}. \tag{3.17}$$

Taking the line search in the Eq (2.12), we have the following:

$$f_{k+1} \leq Q_k + \delta\alpha_k g_k^T d_k. \tag{3.18}$$

Additionally, by the inequality in Eq (3.13), we obtain the following:

$$f_{k+1} \leq Q_k - \left(\frac{2\delta(1-\delta)}{c_1\zeta}\right)\left(\frac{|g_k^T d_k|}{\|d_k\|^2}\right). \tag{3.19}$$

Based on the sufficient descent property from Lemma 3.2 and the bound property from Lemma 3.3, we have the following:

$$f_{k+1} \leq Q_k - \left(\frac{2\delta(1-\delta)c^2}{c_1\zeta\kappa^2}\right)\|g_k\|^2. \tag{3.20}$$

Combining Eqs (2.13) and (3.16), we have the following:

$$Q_k = \frac{\eta_k P_k Q_k + f_{k+1}}{P_{k+1}} \leq \frac{\eta_k P_k Q_k + Q_k - \beta\|g_k\|^2}{P_{k+1}} \tag{3.21}$$

$$\begin{aligned} &= \frac{Q_k(\eta_k P_{k+1} + 1) - \beta\|g_k\|^2}{P_{k+1}} \\ &= \frac{Q_k P_{k+1} - \beta\|g_k\|^2}{P_{k+1}} = Q_k - \frac{\beta\|g_k\|^2}{P_{k+1}}. \end{aligned} \tag{3.22}$$

As f is bounded from below, and $f_k \leq Q_k$ for every k, we can deduce that Q_k is also bounded from below. Therefore, (3.22) implies that

$$\sum_{k=1}^{\infty} \frac{\|g_k\|^2}{P_{k+1}} < \infty. \tag{3.23}$$

Since $P_{k+1} \leq k + 2$ by Eq (3.13), Eq (3.22) would not hold if $\|g_k\|$ were bounded away from 0. Accordingly, if $\eta_{max} < 1$, then by (3.13),

$$\begin{aligned} P_{k+1} &= 1 + \sum_{j=0}^{k} \prod_{i=0}^{j} \eta_{k-i} \leq 1 + \sum_{j=0}^{k} \eta_{max}^{j+1} \leq \sum_{j=0}^{k} \eta_{max}^{j} \\ &= \frac{1}{1 - \eta_{max}}. \end{aligned} \tag{3.24}$$

Therefore, we can deduce that (3.14) directly entails (3.15). Thus, the proof is concluded. □

Table 1. Test functions, initial points, and references.

No.	Function name	Starting point	Ref.
A1	Variably dimensioned	$(1 - \frac{1}{n}, 1 - \frac{2}{n}, \ldots, 0)^T$	[34]
A2	Problem 206	$(2, 2, \ldots, 2)^T$	[35]
A3	Trigonometric function	$(\frac{1}{n}, \frac{1}{n}, \ldots)^T$	[36]
A4	Penalty function 1	$(3, 3, \ldots, 3)^T$	[34]
A5	Discrete boundary-value	$(\frac{1}{n+1}, \frac{1}{n+1} - 1, \ldots, \frac{1}{n+1} - 1)^T$	[36]
A6	Linear full rank	$(1, 1, \ldots, 1)^T$	[36]
A7	Problem 202	$(2, 2, \ldots, 2)^T$	[35]
A8	Problem 212	$(0.5, 0.5, \ldots, 0.5)^T$	[35]
A9	Raydan 1	$(\frac{1}{n}, \frac{2}{n}, \ldots, 1)^T$	[34]
A10	Raydan 2	$(\frac{1}{10n}, \frac{2}{10n}, \ldots, \frac{1}{10n})^T$	[34]
A11	Sine function 2	$(1, 1, \ldots, 1)^T$	[37]
A12	Exponential function 1	$(\frac{n}{n-1}, \frac{n}{n-1}, \ldots, \frac{n}{n-1})^T$	[34]
A13	Singular Function 2	$(1, 1, \ldots, 1)^T$	[34]
A14	Ext. Freudenstein & Roth function	$(6, 3, 6, 3, \ldots, 6, 3)^T$	[34]
A15	Function 21	$(1, 1, \ldots, 1)^T$	[34]
A16	Broyden Tridiagonal Function	$(-1, -1, \ldots, -1)^T$	[36]
A17	Exponential function 2	$(\frac{1}{n^2}, \frac{1}{n^2}, \ldots, \frac{1}{n^2})^T$	[34]
A18	Extended Himmelblau	$(1, \frac{1}{n}, 1, \frac{1}{n}, \ldots, 1, \frac{1}{n})^T$	[38]
A19	Function 27	$(100, \frac{1}{n^2}, \ldots, \frac{1}{n^2})^T$	[34]
A20	Triglog function	$(1, 1, \ldots, 1)^T$	[38]
A21	Ext. Powell Singular function	$(1.5E - 4, \ldots, 1.5E - 4)^T$	[34]
A22	Brown almost linear function	$(0.5, 0.5, \ldots, 0.5)^T$	[36]
A23	Zerojacobian Function	$if\ i = 1, \frac{100(n-100)}{n}, if\ i \geq 2, \frac{(n-1000)(n-500)}{(60n)^2}$	[34]

4. Numerical results

This section evaluates the computational efficiency of the proposed algorithm by comparing it with existing methods such as RMIL [22], CG_Descent [39], and the VRMIL method [23]. The comparison parameters include the number of iterations (Iter), the number of function evaluations (Feval), the CPU time (Cpu), and the value of the function (Value_F). Table 1 provides specifics on these benchmark test functions used in the experiment.

The algorithms in this experiment were all implemented in MATLAB R2022a and ran on a system that had an Intel® CoreTM i7-3537U CPU (2.00 GHz) and 8 GB of RAM. The program was set to terminate if the stopping criterion $\|g_k\| \leq 10^{-5}$ was met or if any of the following conditions occurred:

- The iteration count surpassed 1000; and
- The total function evaluations surpassed 5000.

The numerical results of the experiments can be found at `https://github.com/Rabiu-Bashir/Numerical-Results-SRMILCG-Method`. The results are summarized in the graphs using the performance profile introduced by [40]. The authors in [40] developed a methodology to evaluate and contrast a solver's performance s over a collection of problems a. This approach considers n_s solvers and n_a problems as described below, and calculates the computing cost q for any combination of solver and problem:

$$q_{a,s} = \text{Performance measure for solver } s \text{ on a problem } a.$$

Based on computational cost $q_{a,s}$, [40] further developed a metric to compare the efficiency of the solver. The performance ratio, which serves as a reference point, is defined as follows:

$$r_{a,s} = \frac{q_{a,s}}{\min\{q_{a,s} : s \in S\}}.$$

Furthermore, the distribution function can be found using the following:

$$\rho_s(\tau) = \frac{1}{n_a}\text{size}\{a \in A : \log_2(r_{a,s}) \leq \tau\}.$$

This methodology generates performance profile graphs for each solver $s \in S$ using the available data. These graphs, called performance profiles, show the percentage $\rho_s(\tau)$ of problems that a particular solver solves. This methodology generates solutions within a factor τ of the best performance using the currently available data. For a given τ, a solver is said to be more efficient if it produces a greater $\rho_s(\tau)$. Therefore, the method that maintains the best performance profile curve for all values of τ is the most efficient.

In this study, numerical experiments are used to generate performance profile charts for (Iter), (Feval), (Cpu), and (VALUE_F), corresponding to Figures 1–4, alongside the comparisons. In particular, Figure 1 shows that for $\tau \geq 2$, SRMILCG successfully solves approximately 95% of the test problems with fewer iterations, whereas CG_Descent, VRMIL, and RMIL handle around 90%, 73%, and 51%, respectively.

Similarly, in Figure 2, for $\tau \geq 2$, SRMILCG solves almost 95% of the test problems with the fewest function evaluations, outperforming CG_Descent (89%), VRMIL (63%), and RMIL (50%). In terms of the computational efficiency, SRMILCG takes the least amount of CPU time to solve about 98% of the test issues for $\tau \geq 2$, compared to 94% for CG_Descent, 72% for VRMIL, and 64% for RMIL, as shown in Figure 3. Furthermore, Figure 4 reveals that the SRMILCG algorithms consistently outperform CG_Descent, VRMIL, and RMIL, thus providing a more precise approximation of the solution while substantially reducing errors. A key takeaway from these results is that while all algorithms competitively perform at lower values of τ, the SRMILCG method consistently outperforms the others across all metrics as τ increases. This indicates the proposed approach's resilience and competitiveness for the problem set under consideration.

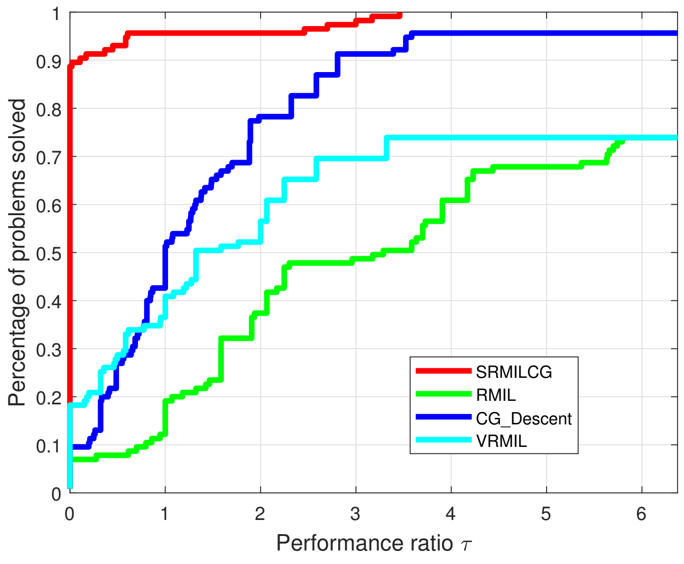

Figure 1. Profiles of performance by Iter.

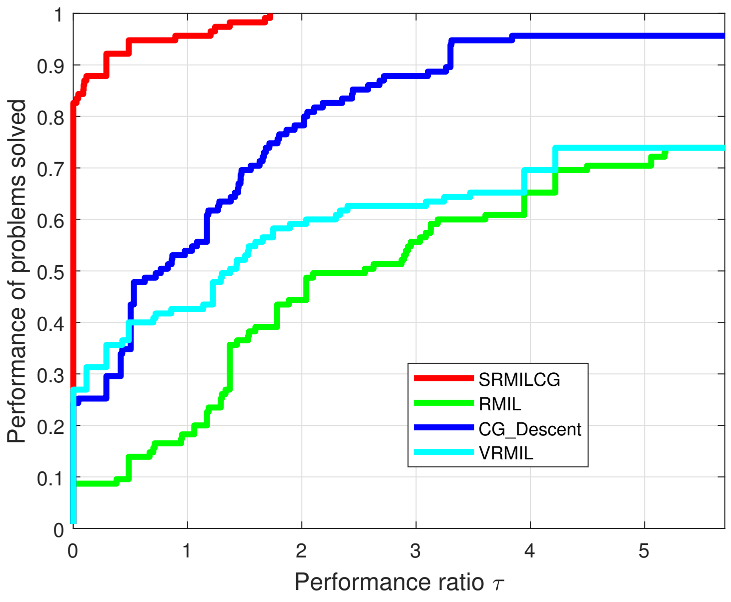

Figure 2. Profiles of performance by Feval.

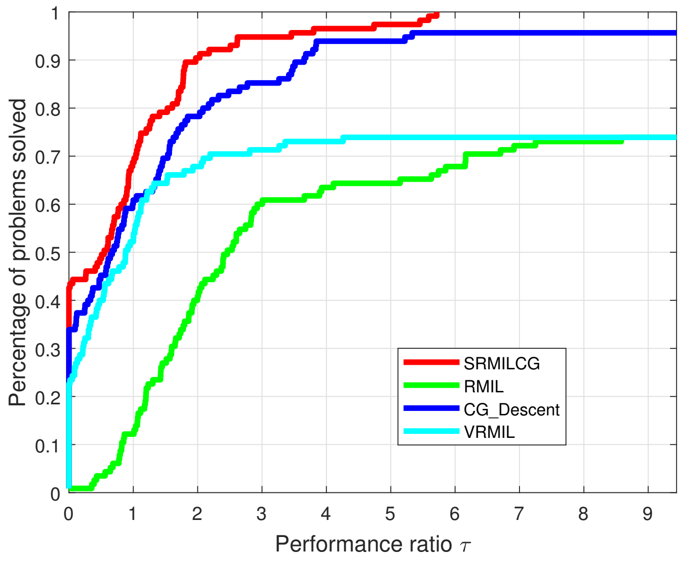

Figure 3. Profiles of performance by CPU time.

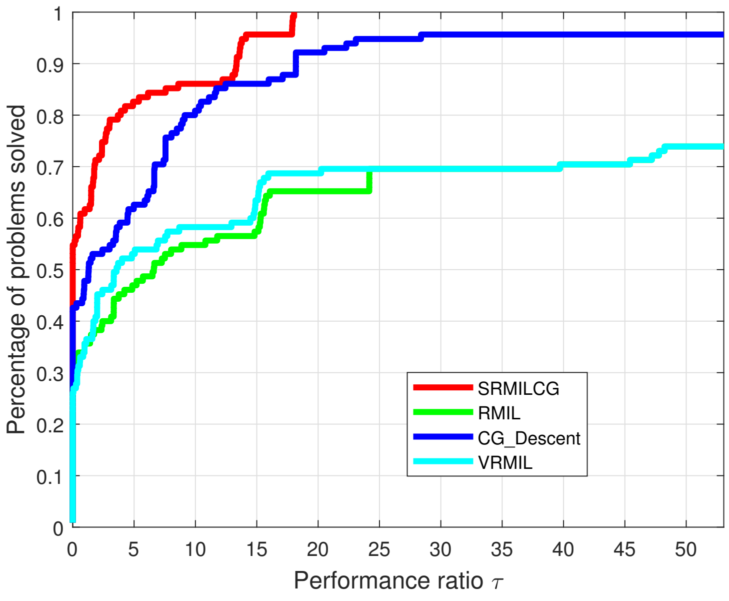

Figure 4. Profiles of performance by norm of the function.

5. Application in image restoration problems

Image restoration is the recovery of an image that is degraded due to noise, blur, or other distortions. The goal is to reconstruct the original image as accurately as possible by reducing noise, correcting distortions, and enhancing details. We apply the proposed SRMILCG method to solve image restoration problems. Raymond et al. [41] implemented a two-phase approach to recover images affected by impulse noise. To identify noisy pixels in the first stage, a median filter was applied.

Examine a picture X of size $M \times N$, in which the location of the pixels of X is represented by the index set $A = \{1, 2, \dots, M\} \times \{1, 2, \dots, N\}$. Let $|N|$ indicate the cardinality of the noise-affected pixels found in the first phase, and let $N \subset A$ represent the indices corresponding to those pixels. The set of four nearest neighbors for each pixel at position $(m, n) \in A$ is defined as $B_{m,n}$:

$$B_{m,n} = \{(m, n-1), (m, n+1), (m-1, n), (m+1, n)\}.$$

Furthermore, let $y_{m,n}$ represent the observed pixel intensity at location (m, n).

In the second stage, the following nonsmooth minimization issue is solved to restore noisy pixels:

$$\min_q \sum_{(m,n)\in N} \left[|q_{m,n} - y_{m,n}| + \frac{\beta}{2} \left(2 \cdot Z^1_{m,n} + Z^2_{m,n} \right) \right], \tag{5.1}$$

where

$$Z^1_{m,n} = \sum_{(a,b)\in B_{m,n} \setminus N} \psi_\alpha(q_{m,n} - y_{a,b}), \quad Z^2_{m,n} = \sum_{(a,b)\in B_{m,n} \cap N} \psi_\alpha(q_{m,n} - q_{a,b}),$$

with $\psi_\alpha(t) = \sqrt{t^2 + \alpha}$ representing an edge-preserving function with the parameter $\alpha > 0$. It plays a crucial role in regularization, especially when the goal is to remove noise while retaining important features, such as edges. The vector $q = [q_{m,n}]_{(m,n)\in N}$ is a length-based column vector. $|N|$ is arranged in lexicographic order. Finding the exact solution to the non-smooth minimization problem (5.1) is computationally demanding and time-consuming. The nonsmooth term was eliminated by Cai et al. [42], which yielded the following smooth unconstrained optimization that is shown below:

$$\min_q F_\alpha(q) := \sum_{(m,n)\in N} \left(2 \sum_{(a,b)\in B_{m,n} \setminus N} \psi_\alpha(q_{m,n} - y_{a,b}) + \sum_{(a,b)\in B_{m,n} \cap N} \psi_\alpha(q_{m,n} - q_{a,b}) \right). \tag{5.2}$$

The scale of (5.2) increases with an increasing noise ratio. Using the CG scheme to address the problem (5.2) mentioned above, the authors in [43] found that damaged images can be successfully restored.

Now, we reduce the salt-and-pepper noise, which is a particular kind of impulse noise, using the two-phase method. To find the noisy pixels in the first stage, an adaptive median filter is used [44]. In the second stage, we solve (5.2) using SRMILCG and evaluate its performance against the CG_DESCENT [39], and the RMIL of [22]. Furthermore, it should be noted that the nonmonotone line search (2.12) is used in all approaches to obtain the step length α_k. Man (512×512), Lena (512×512), Boat (512×512), and Hill (512×512) are the test photographs that were used. The following halting criterion was used by all of the methods under comparison:

$$\text{Itr} > 300 \quad \text{or} \quad \frac{|F_\alpha(q_k) - F_\alpha(q_{k-1})|}{|F_\alpha(q_k)|} \le 10^{-4}. \tag{5.3}$$

The experiments were performed in MATLAB R2022a on a personal computer featuring an Intel® Core™ i7-3537U processor (2.00 GHz) with 8 GB of RAM. We used the well-known peak signal-to-noise ratio (PSNR) to assess the quality of image restoration. For further information, refer to [45]. PSNR is defined as follows:

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\frac{1}{MN}\sum_{m,n}(x^o_{m,n} - x^*_{m,n})^2}\right), \tag{5.4}$$

where $x^o_{m,n}$ and $x^*_{m,n}$ denote the pixel values of the original and restored images, respectively.

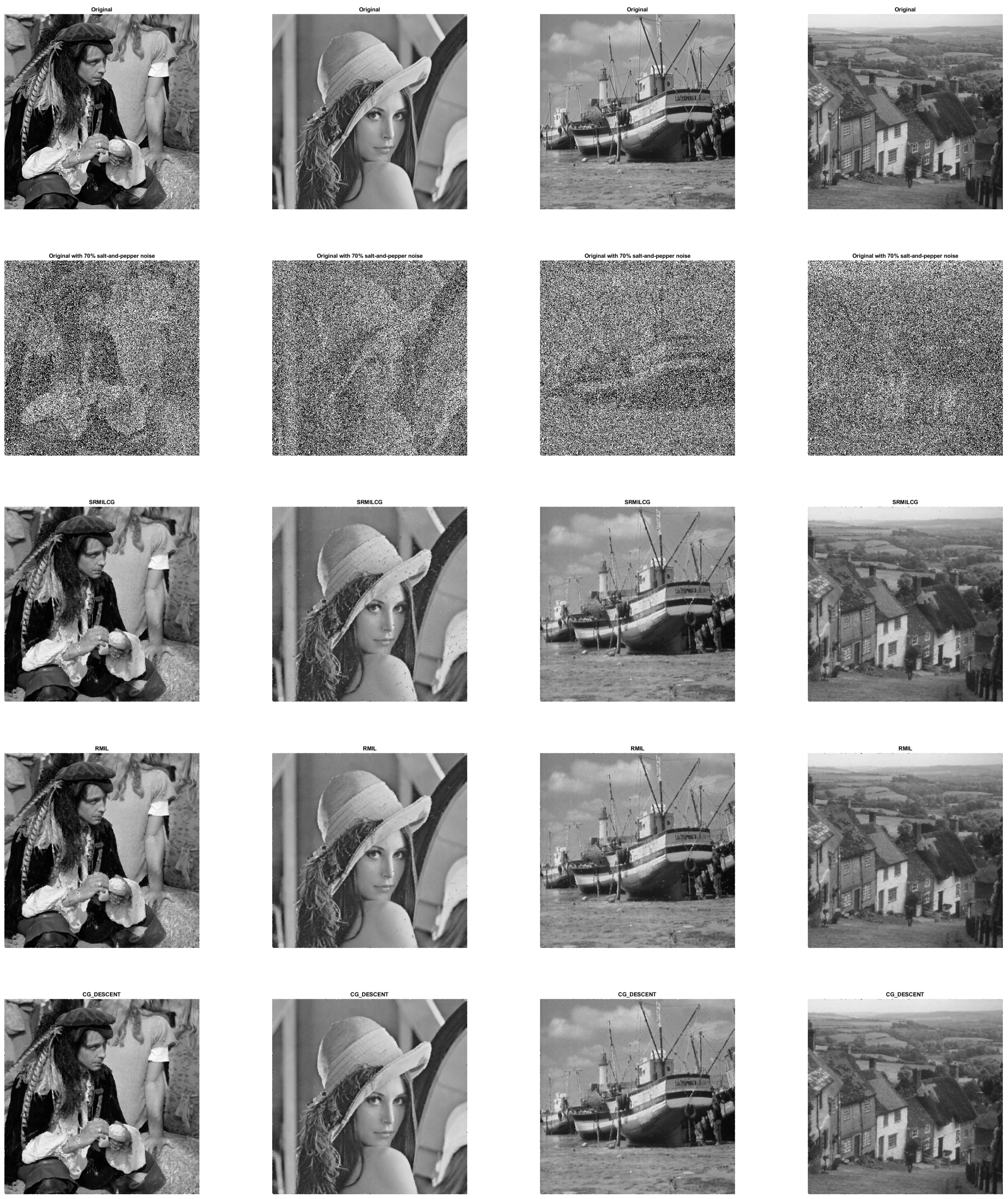
Original
Original
Original
Original
Original with 70% salt-and-pepper noise
Original with 70% salt-and-pepper noise
Original with 70% salt-and-pepper noise
Original with 70% salt-and-pepper noise
SRMILCG
SRMILCG
SRMILCG
SRMILCG
RMIL
RMIL
RMIL
RMIL
CG_DESCENT
CG_DESCENT
CG_DESCENT
CG_DESCENT

Figure 5. The original image (1st row) with (2nd row) 70% salt-and-pepper noise, the restored images by SRMILCG(3rd row), RMIL (4th row),and CG_DESCENT (5th row).

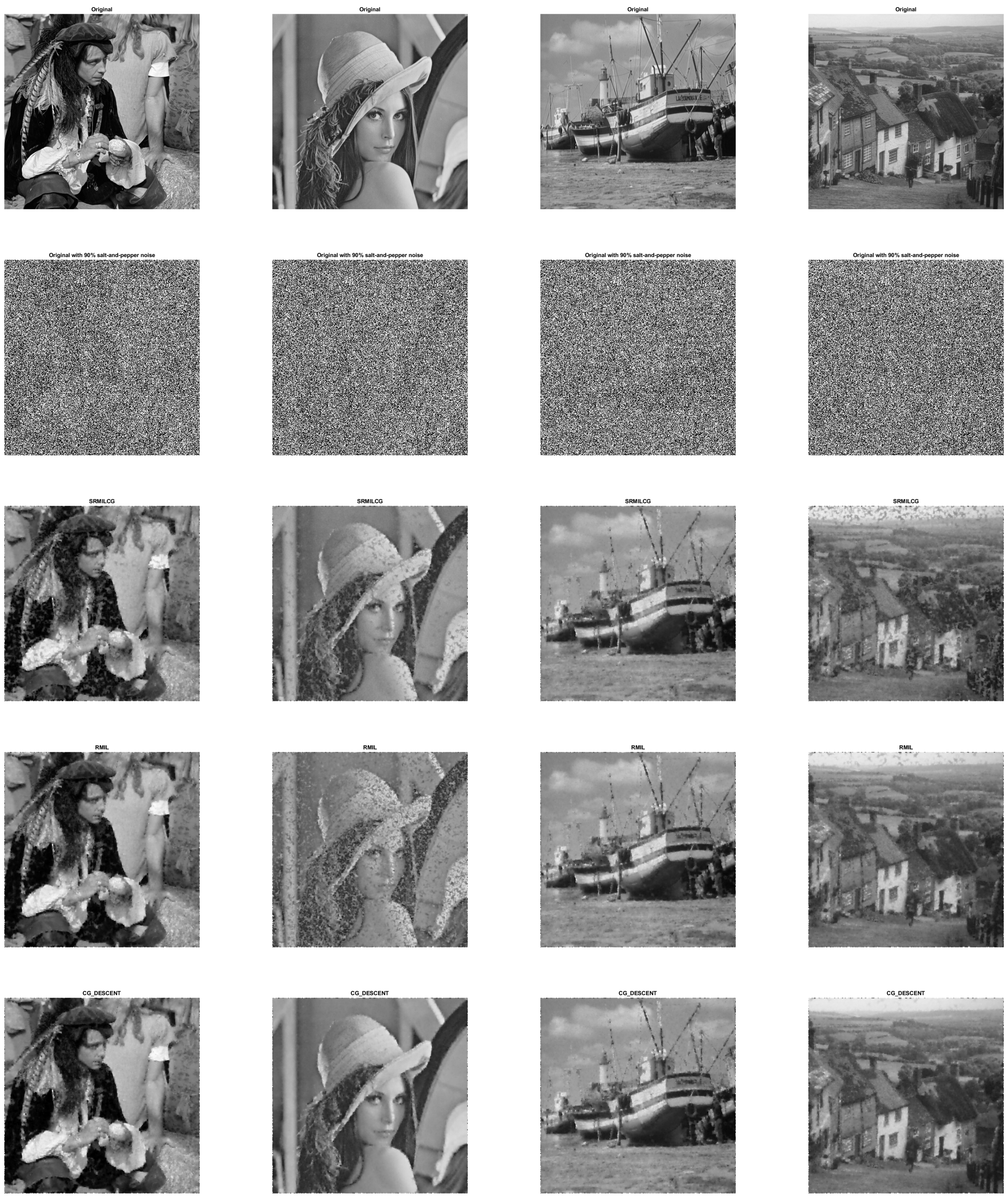
Original
Original
Original
Original
Original with 90% salt-and-pepper noise
Original with 90% salt-and-pepper noise
Original with 90% salt-and-pepper noise
Original with 90% salt-and-pepper noise
SRMILCG
SRMILCG
SRMILCG
SRMILCG
RMIL
RMIL
RMIL
RMIL
CG_DESCENT
CG_DESCENT
CG_DESCENT
CG_DESCENT

Figure 6. The original image (1st row) with (2nd row) 90% salt-and-pepper noise, the restored images by SRMILCG(3rd row), RMIL (4th row), and CG_DESCENT (5th row).

Table 2 provides a summary of the iterations of the restored images (Itr), CPU time (Cpu), and PSNR values. To conserve space, we only show the original and restored images for the three algorithms at 70% and 90% salt-and-pepper noise levels. The corresponding visual results are shown in Figures 5 and 6, respectively.

As observed in Table 2, the proposed SRMILCG method typically achieves convergence in fewer iterations and with lower CPU times compared to the other two algorithms. Furthermore, the PSNR values of images restored using SRMILCG are generally higher, with only a few exceptions. Overall, SRMILCG demonstrates a superior performance over RMIL and CG_DESCENT in the tested images.

Table 2. Numerical comparisons with different noise ratios.

Image	**Noise ratio**	**SRMILCG**	**RMIL**	**CG_DESCENT**
		Itr / Cpu / PSNR	**Itr / Cpu / PSNR**	**Itr / Cpu / PSNR**
Man	0.30	15 / 11.10 / 31.50	18 / 7.28 / 31.50	20 / 8.70 / 31.54
	0.50	15 / 21.97 / 28.62	25 / 13.80 / 29.10	23 / 13.12 / 29.11
	0.70	23 / 20.69 / 26.14	29 / 21.39 / 26.09	32 / 22.61 / 26.25
	0.90	24 / 21.81 / 20.78	38 / 44.70 / 22.05	42 / 37.06 / 22.28
Lena	0.30	15 / 10.30 / 37.00	16 / 7.72 / 36.91	20 / 8.62 / 36.93
	0.50	18 / 23.02 / 34.12	22 / 14.22 / 34.33	20 / 14.20 / 34.36
	0.70	15 / 20.75 / 29.03	19 / 19.29 / 30.55	31 / 23.53 / 31.15
	0.90	22 / 22.22 / 23.64	15 / 22.54 / 19.68	43 / 37.82 / 26.24
Boat	0.30	9 / 7.82 / 32.69	15 / 7.32 / 33.54	18 / 8.27 / 33.64
	0.50	25 / 21.55 / 31.15	23 / 14.99 / 31.12	23 / 16.48 / 31.11
	0.70	26 / 23.87 / 28.24	23 / 19.73 / 27.89	26 / 20.42 / 28.26
	0.90	30 / 26.24 / 23.98	43 / 49.22 / 23.97	42 / 42.28 / 24.04
Hill	0.30	17 / 13.21 / 34.88	16 / 7.78 / 34.87	14 / 7.07 / 34.93
	0.50	22 / 18.60 / 32.62	22 / 12.92 / 32.60	20 / 12.72 / 32.58
	0.70	26 / 21.70 / 29.70	34 / 19.90 / 29.70	29 / 20.87 / 29.77
	0.90	25 / 32.91 / 22.69	34 / 31.17 / 25.03	46 / 38.16 / 25.55

6. Conclusions

In this work, we proposed a structured CG-based method based on the RMIL formula. Independent of the line search strategy employed, the proposed method satisfied the sufficient descent condition. We established its global convergence under nonmonotone line search conditions and evaluated its efficiency on several NLS benchmark problems, as well as image restoration tasks drawn from the existing literature. The results demonstrated the competitiveness of the proposed approach compared to the RMIL, CG_DESCENT, and VRMIL methods, particularly in terms of the iteration count, function evaluations, CPU time, and the norm of the residual. Future research should focus on developing more advanced structured conjugate gradient algorithms capable of efficiently solving high-dimensional NLS problems in fields such as optimal control and machine learning.

Author contributions

Rabiu Bashir Yunus: Conceptualization, Methodology, Software, Writing-original draft; Ahmed R. El-Saeed: Investigation, Funding acquisition, Validation, Review & editing; Nooraini Zainuddin: Methodology, Supervision, Visualization; Hanita Daud: Investigation, Resources, Supervision, Validation. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Funding statements

This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-DDRSP2501).

Conflict of interest

The authors declare that they have no competing interests to disclose.

References

1. H. Kim, C. Wang, H. Byun, W. Hu, S. Kim, Q. Jiao, et al., Variable three-term conjugate gradient method for training artificial neural networks, *Neural Networks*, **159** (2023), 125–136. https://doi.org/10.1016/j.neunet.2022.12.001
2. N. Sato, K. Izumi, H. Iiduka, Scaled conjugate gradient method for nonconvex optimization in deep neural networks, *J. Mach. Learn. Res.*, **25** (2024), 1–37.
3. S. J. Wright, R. D. Nowak, M. A. T. Figueiredo, Sparse reconstruction by separable approximation, *IEEE Trans. Signal Process.*, **57** (2009), 2479–2493. https://doi.org/10.1109/TSP.2009.2016892
4. M. M. Yahaya, P. Kumam, A. M. Awwal, P. Chaipunya, S. Aji, and S. Salisu, A New Generalized Quasi-Newton Algorithm Based on Structured Diagonal Hessian Approximation for Solving Nonlinear Least-Squares Problems With Application to 3DOF Planar Robot Arm Manipulator, *IEEE Access*, **10** (2022), 10816–10826. https://doi.org/10.1109/ACCESS.2022.3144875
5. W. Xu, N. Zheng, K. Hayami, Jacobian-free implicit inner-iteration preconditioner for nonlinear least squares problems, *J. Sci. Comput.*, **68** (2016), 1055–1081. https://doi.org/10.1007/s10915-016-0167-z
6. H. Zhang, A. R. Conn, K. Scheinberg, A derivative-free algorithm for least-squares minimization, *SIAM J. Optim.*, **20** (2010), 3555–3576. https://doi.org/10.1137/09075531X
7. H. Mohammad, M. Y. Waziri, S. A. Santos, A brief survey of methods for solving nonlinear least-squares problems, *Numer. Algebra Control Optim.*, **9** (2019), 1–13. https://doi.org/10.3934/naco.2019001
8. K. Madsen, H. B. Nielsen, O. Tingleff, *Methods for Non-linear Least Squares Problems*, Lyngby: Informatics and Mathematical Modelling Technical University of Denmark, 2004.
9. R. B. Yunus, N. Zainuddin, H. Daud, R. Kannan, M. M. Yahaya, A. Al-Yaari, An improved accelerated 3-term conjugate gradient algorithm with second-order Hessian approximation for nonlinear least-squares optimization, *J. Math. Comput. Sci.*, **36** (2025), 263–274. https://doi.org/10.22436/jmcs.036.03.02

10. R. B. Yunus, N. Zainuddin, H. Daud, R. Kannan, S. A. Abdul Karim, M. M. Yahaya, A modified structured spectral HS method for nonlinear least squares problems and applications in robot arm control, *Mathematics*, **11** (2023), 3215. https://doi.org/10.3390/math11143215

11. F. Ding, L. Xu, X. Zhang, Y. Zhou, X. Luan, Recursive identification methods for general stochastic systems with colored noises using the hierarchical identification principle and the filtering identification idea, *Annu. Rev.. Control*, **57** (2024), 100942. https://doi.org/10.1016/j.arcontrol.2024.100942

12. F. Ding, Least squares parameter estimation and multi-innovation least squares methods for linear fitting problems from noisy data, *J. Comput. Appl. Math.*, **426** (2023), 115107. https://doi.org/10.1016/j.cam.2023.115107

13. M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand.*, **49** (1952), 409–436. https://doi.org/10.6028/jres.049.044

14. E. Polak, G. Ribiere, Note on the convergence of conjugate direction methods, *ESAIM Math. Model. Numer. Anal.*, **3** (1969), 35–43. https://doi.org/10.1007/BF01078750

15. B. T. Polyak, The conjugate gradient method in extremal problems, *USSR Comput. Math. Math. Phys.*, **9** (1969), 94–112. https://doi.org/10.1016/0041-5553(69)90035-4

16. Y. Liu, C. Storey, Efficient generalized conjugate gradient algorithms. Part 1: Theory, *J. Optim. Theory Appl.*, **69** (1991), 129–137. https://doi.org/10.1007/BF00940464

17. W. W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, *Pacific J. Optim.*, **2** (2006), 35–58.

18. S. Yao, W. Zengxin, H. Hai, A note about WYL's conjugate gradient method and its applications, *Appl. Math. Comput.*, **191** (2006), 381–388.

19. R. Fletcher, C. M. Reeves, Function minimization by conjugate gradients, *Comput. J.*, **7** (1964), 149–154. https://doi.org/10.1093/comjnl/7.2.149

20. R. Fletcher, *Practical Methods of Optimization*, Chichester: John Wiley & Sons, 2013.

21. Y. H. Dai, Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optim.*, **10** (1999), 177–182. https://doi.org/10.1137/S1052623497318992

22. M. Rivaie, M. Mamat, L. W. June, I. Mohd, A new class of nonlinear conjugate gradient coefficients with global convergence properties, *Appl. Math. Comput.*, **218** (2012), 11323–11332. https://doi.org/10.1016/j.amc.2012.05.030

23. X. Wu, H. Shao, P. Liu, Y. Zhang, Y. Zhuo, An efficient conjugate gradient-based algorithm for unconstrained optimization and its projection extension to large-scale constrained nonlinear equations with applications in signal recovery and image denoising problems. *J. Comput. Appl. Math.*, **422**, (2023), 114879. https://doi.org/10.1016/j.cam.2022.114879

24. Z. Dai, Comments on a new class of nonlinear conjugate gradient coefficients with global convergence properties, *Appl. Math. Comput.*, **276**, (2016), 297–300. https://doi.org/10.1016/j.amc.2015.11.085

25. O. O. O. Yousif, M. A. Saleh, Another modified version of RMIL conjugate gradient method, *Appl. Numer. Math.*, **202** (2024), 120–126.

26. O. O. O. Yousif, The convergence properties of RMIL+ conjugate gradient method under the strong Wolfe line search, *Appl. Math. Comput.*, **367** (2020), 124777. https://doi.org/10.1016/j.amc.2019.124777

27. X. Zhang, F. Ding, Optimal adaptive filtering algorithm by using the fractional-order derivative, *IEEE Signal Process. Lett.*, **29** (2021), 399–403. https://doi.org/10.1109/LSP.2021.3136504

28. J. Z. Zhang, Y. Xue, K. Zhang, A structured secant method based on a new quasi-Newton equation for nonlinear least squares problems, *BIT Numer. Math.*, **43** (2003), 217–229. https://doi.org/10.1023/A:1023665409152

29. L. H. Chen, N. Y. Deng, J. Z. Zhang, A modified quasi-Newton method for structured optimization with partial information on the Hessian, *Comput. Optim. Appl.*, **35** (2006), 5–18. https://doi.org/10.1007/s10589-006-6440-6

30. J. E. Dennis, H. J. Martinez, R. A. Tapia, Convergence theory for the structured BFGS secant method with an application to nonlinear least squares, *J. Optim. Theory Appl.*, **61** (1989), 161–178. https://doi.org/10.1007/BF00962795

31. E. Amini, A. G. Rizi, A new structured quasi-Newton algorithm using partial information on Hessian, *J. Comput. Appl. Math.*, **234** (2010), 805–811. https://doi.org/10.1016/j.cam.2010.01.044

32. H. Zhang, W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM J. Optim.*, **14** (2004), 1043–1056. https://doi.org/10.1137/S1052623403428208

33. R. B. Yunus, N. Zainuddin, H. Daud, R. Kannan, M. M. Yahaya, S. A. A. Karim, New CG-based algorithms with second-order curvature information for NLS problems and a 4DOF arm robot model, *IEEE Access*, **12** (2024), 61086–61103. https://doi.org/10.1109/ACCESS.2024.3393555

34. W. L. Cruz, J. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Math. Comput.*, **75** (2006), 1429–1448. https://doi.org/10.1090/S0025-5718-06-01840-0

35. L. Lukšan, J. Vlček, Test problems for unconstrained optimization, Technical Report 897, Academy of Sciences of the Czech Republic, Institute of Computer Science, 2003. Available from: `http://www.cs.cas.cz/luksan/test.html`.

36. J. J. Moré, B. S. Garbow, K. E. Hillstrom, Testing unconstrained optimization software, *ACM Trans. Math. Softw. (TOMS)*, **7** (1981), 17–41.

37. J. Liu, S. Li, A projection method for convex constrained monotone nonlinear equations with applications, *Comput. Math. Appl.*, **70** (2015), 2442–2453. https://doi.org/10.1016/j.camwa.2015.09.014

38. M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimisation problems, *Int. J. Math. Model. Numer. Optim.*, **4** (2013), 150–194.

39. W. W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, *Pac. J. Optim.*, **2** (2006), 35–58.

40. E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2002), 201–213. https://doi.org/10.1007/s101070100263

41. R. H. Chan, C. W. Ho, M. Nikolova, Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization, *IEEE Trans. Image Process.*, **14** (2005), 1479–1485. https://doi.org/10.1109/TIP.2005.852196

42. J.-F. Cai, R. Chan, B. Morini, Minimization of an edge-preserving regularization functional by conjugate gradient type methods, *Image Processing Based on Partial Differential Equations: Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems*, 2007, 109–122.

43. X. Jiang, W. Liao, J. Yin, J. Jian, A new family of hybrid three-term conjugate gradient methods with applications in image restoration, *Numer. Algorithms*, **91** (2022), 161–191. https://doi.org/10.1007/s11075-022-01258-2

44. H. Hwang, R. A. Haddad, Adaptive median filters: new algorithms and results, *IEEE Trans. Image Process.*, **4** (1995), 499–502. https://doi.org/10.1109/83.370679

45. A. C. Bovik, *Handbook of Image and Video Processing*, Academic Press, 2010.