Mathematics

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

AIMS Mathematics, 9(12): 36167–36190.
DOI: 10.3934/math.20241717
Received: 16 September 2024
Revised: 05 November 2024
Accepted: 11 November 2024
Published: 27 December 2024

Research article

A hybrid approach to conjugate gradient algorithms for nonlinear systems of equations with applications in signal restoration

Xuejie Ma[1] **and Songhua Wang**[2,*]

[1] School of Artificial Intelligence, Guangzhou Huashang College, Guangzhou 511300, China
[2] School of Mathematics, Physics and Statistics, Baise University, Baise 533099, China

* **Correspondence:** Email: wang6600789@bsuc.edu.cn.

Abstract: This paper proposes a novel hybrid PRP-HS-LS-type conjugate gradient algorithm for solving constrained nonlinear systems of equations. The proposed algorithm presents several significant advancements and key features: (i) the conjugate parameter is constructed by utilizing the hybrid technique; (ii) the search direction, designed with the conjugate parameter, possesses sufficient descent and trust region properties without the need for a line search mechanism; (iii) the global convergence is rigorously established under general assumptions, notably without the requirement of the Lipschitz continuity condition; (vi) numerical experiments demonstrate the algorithm's efficiency, particularly in solving large-scale constrained nonlinear systems of equations and addressing the sparse signal restoration problem.

Keywords: constrained nonlinear systems of equations; conjugate gradient method; large-scale; global convergence; signal reconstruction
Mathematics Subject Classification: 65K05, 90C56

1. Introduction

1.1. Background overview

In numerous practical applications, such as neural network optimization [1, 2], image segmentation [3], signal processing [4–7], matrix equations [8, 9], and chemical equilibrium analysis [10], nonlinear systems of equations with convex constraints play an essential role. These applications often involve solving problems where the goal is to find solutions satisfying the convex constraints. Therefore, the research into efficient numerical methods for solving nonlinear systems of equations with convex constraints is not only of significant theoretical interest but also holds profound implications for practical applications. In this paper, we focus on the study of nonlinear systems of equations with convex constraints, which can be formulated as follows:

$$h(x) = 0, \quad x \in H, \tag{1.1}$$

where $H \subseteq \mathbb{R}^n$is a non-empty closed convex subset, and $h : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous monotone mapping that may not necessarily be smooth. Throughout this paper, we denote $\|\cdot\|$ by the Euclidean vector norm.

At present, numerical iterative methods for solving nonlinear systems of equations can be broadly categorized into two major types. The first type includes methods that rely on the Jacobian matrix or its approximations, such as Newton's methods [11, 12], quasi-Newton methods [13], trust-region methods [14], and Levenberg–Marquardt methods [15, 16]. These methods are known for their ability to exhibit rapid local convergence, especially when an appropriate initial point is selected. However, the effectiveness of these methods depends on the computation and storage of the Jacobian matrix or its approximation value. This dependency can lead to algorithm failure or inefficiency, particularly in cases where the Jacobian matrix is difficult to obtain or when nonlinear systems of equations exhibits non-smooth characteristics. The second type includes methods that do not rely on the Jacobian matrix or its approximations, such as conjugate gradient (CG) methods [17–19], spectral gradient methods [20], and other derived first-order gradient-based methods [21]. These methods are known for being structurally simple and requiring only first-order derivative information and no matrix storage, which makes them particularly well-suited for solving large-scale nonlinear systems of equations. Due to their simplicity and scalability, these methods have gained popularity among researchers, especially in applications where computational resources are limited or where nonlinear systems of equations are too large for traditional Jacobian-based methods to be practical.

In recent years, the CG method has garnered significant attention from researchers in the field of optimization, becoming a focal point of study. As an efficient iterative method for solving linear systems of equations and unconstrained optimization problems, the CG method has demonstrated notable advantages, particularly in handling large-scale problems. The versatility of the CG method has been further enhanced through its combination with projection techniques, leading to the development of CG projection methods. For example, Liu et al. [22] proposed a new derivative-free spectral Dai–Yuan (DY) type projection method aimed at solving the problem (1.1). Their study, under reasonable assumptions, not only established the global convergence of the proposed method but also demonstrated its linear convergence rate. Besides, Hu et al. [23] developed a modified Hestenes–Stiefel (HS) type CG projection method. This method integrated the steepest descent approach with the traditional CG method and was applied to solve image restoration problems. Ma et al. [24] designed a modified inertial three-term CG projection method for solving the problem (1.1). This method integrated the inertial extrapolation step into the search direction, and was implemented for the sparse signal and image restoration problems.

The CG projection method updates its search direction using the most recent gradient information and the previous search direction. While this method generally performs well, it can encounter challenges such as slow convergence or unstable convergence performance in certain situations. To address these issues, the three-term CG projection method introduces a new approach for updating the search direction. This approach incorporates not only the current gradient information but also the gradient information from the previous two iterations. The key idea is to leverage a richer set of historical gradient information to enhance the search direction, with the aim of accelerating convergence and improving solution accuracy. Without loss of generality, the iterative formula for the three-term CG projection method is given by $x_{k+1} = x_k + \alpha_k d_k$, where α_k is the step size determined by a line search mechanism. The search direction d_k is defined as:

$$d_k = \begin{cases} -h_0, & k = 0, \\ -h_k + \beta_k d_{k-1} + \theta_k y_{k-1}, & k \geq 1, \end{cases}$$

where β_k is a conjugate parameter, θ_k is a scalar parameter, $h_k \triangleq h(x_k)$, and $y_{k-1} = h_k - h_{k-1}$. The parameters β_k and θ_k play a critical role in the performance of the CG projection method, influencing the algorithm's convergence behavior. Traditional CG methods such as Polak–Ribière–Polyak (PRP), Hestenes–Stiefel (HS), and Liu–Storey (LS) are well-known in the field, with the conjugate parameter expressions given by:

$$\beta_k^{\mathrm{PRP}} = \frac{h_k^{\mathrm{T}} y_{k-1}}{\|h_{k-1}\|^2}, \quad \beta_k^{\mathrm{HS}} = \frac{h_k^{\mathrm{T}} y_{k-1}}{d_k^{\mathrm{T}} y_{k-1}}, \quad \beta_k^{\mathrm{LS}} = \frac{h_k^{\mathrm{T}} y_{k-1}}{-h_{k-1}^{\mathrm{T}} d_{k-1}}. \tag{1.2}$$

Given the outstanding numerical performance of the PRP and HS CG methods, Yin et al. [25] introduced a hybrid three-term CG projection method for solving the problem (1.1), with a particular focus on compressed sensing problems. Additionally, Gao et al. [26] designed an adaptive three-term CG search direction and validated the effectiveness of the proposed method through extensive numerical experiments. Their results demonstrated that the adaptive approach is highly competitive, particularly in solving sparse signal problems.

1.2. Formulation of search direction

To provide a smoother and more coherent introduction to the ideas and algorithms presented in this paper, we begin by reviewing the relevant contributions of the works [27, 28]. Existing three-term CG methods, specifically developed to address unconstrained optimization problems, have successfully demonstrated their remarkable efficiency and wide-ranging applicability in this field. From the well-established HS CG method, Li et al. [27] presented an innovative three-term HS-type CG method, which was designed for solving unconstrained optimization problems. Their conjugate and scalar parameters are defined as follows:

$$\beta_k^{\mathrm{THS}} = \frac{h_k^{\mathrm{T}} y_{k-1}}{d_{k-1}^{\mathrm{T}} y_{k-1}} - \frac{\|y_{k-1}\|^2 h_k^{\mathrm{T}} d_{k-1}}{(d_{k-1}^{\mathrm{T}} y_{k-1})^2}, \quad \theta_k^{\mathrm{THS}} = t_k \frac{h_k^{\mathrm{T}} d_{k-1}}{d_{k-1}^{\mathrm{T}} y_{k-1}}, \tag{1.3}$$

where $t_k = \min\left\{\hat{t}, \max\left\{0, y_{k-1}^{\mathrm{T}}(y_{k-1} - s_{k-1})/\|y_{k-1}\|^2\right\}\right\}$, $s_{k-1} = x_k - x_{k-1}$, and $0 < \hat{t} < 1$. In addition, Li et al. [28] proposed a three-term PRP-type CG method, also aimed at unconstrained optimization problems. In their approach, the corresponding conjugate and scalar parameters are modified by substituting $\|h_{k-1}\|^2$ with $d_{k-1}^{\mathrm{T}} y_{k-1}$, yielding the following parameters:

$$\beta_k^{\mathrm{TPRP}} = \frac{h_k^{\mathrm{T}} y_{k-1}}{\|h_{k-1}\|^2} - \frac{\|y_{k-1}\|^2 h_k^{\mathrm{T}} d_{k-1}}{\|h_{k-1}\|^4}, \quad \theta_k^{\mathrm{TPRP}} = t_k \frac{h_k^{\mathrm{T}} d_{k-1}}{\|h_{k-1}\|^2}. \tag{1.4}$$

By analyzing the denominators in (1.2), (1.3), and (1.4), we introduce a new notation $\delta_k = \mu\|d_{k-1}\|\|y_{k-1}\| + \max\left\{\|h_{k-1}\|^2, d_{k-1}^{\mathrm{T}} y_{k-1}, -h_{k-1}^{\mathrm{T}} d_{k-1}\right\}$ with $\mu > 0$. Together with the construction forms of (1.3) and (1.4), we design a hybrid modified PRP-HS-LS-type search direction as follows:

$$d_k = \begin{cases} -h_0, & k = 0, \\ -h_k + \beta_k^{\mathrm{MPHL}} d_{k-1} + \theta_k^{\mathrm{MPHL}} y_{k-1}, & k \geq 1, \end{cases} \tag{1.5}$$

where $\beta_k^{\mathrm{MPHL}} = h_k^{\mathrm{T}} y_{k-1}/\delta_k - \|y_{k-1}\|^2 h_k^{\mathrm{T}} d_{k-1}/\delta_k^2$ and $\theta_k^{\mathrm{MPL}} = t_k h_k^{\mathrm{T}} d_{k-1}/\delta_k$. This hybrid method is carefully constructed to ensure that the conjugate parameter maintains the desirable properties of the PRP, HS, and LS parameters. Additionally, it is designed to maintain sufficient descent and trust region properties without the need for a line search mechanism.

2. Description of the proposed algorithm

In this section, we will elaborate on the line search mechanism, the projection operator, and the detailed procedure of the proposed algorithm.

First, the line search mechanism [7, 29] is employed to identify a suitable step size that ensures convergence towards the optimal solution. Specifically, the trial point $z_k = x_k + \alpha_k d_k$ is evaluated, where the step size $\alpha_k = \beta\rho^{i_k}$. Here, β is a predetermined scaling factor that initializes the search, and ρ is a contraction factor that helps refine the step size iteratively. The integer i_k is the smallest nonnegative integer i such that the following condition holds:

$$-\, h(x_k + \beta\rho^i d_k)^{\mathrm{T}} d_k \geq \sigma\beta\rho^i \|h(x_k + \beta\rho^i d_k)\| \|d_k\|^2, \tag{2.1}$$

where the parameter σ is typically chosen between 0 and 1, and it determines the strength of the sufficient decrease condition.

Second, the projection operator [7, 24] is a fundamental tool in the design and theoretical analysis of various algorithms, particularly those addressing convex constrained optimization problems. The projection operator $P_H[\cdot]$, which maps a point from the space $\mathbb{R}^n$ onto its non-empty closed convex subset H, is defined as:

$$P_H[x] = \arg\min_y \{\|x - y\|, y \in H\}, \quad x \in \mathbb{R}^n.$$

This operator plays a critical role in maintaining feasibility within the constraint set H. Moreover, $P_H[\cdot]$ possesses the non-expansive property, i.e., $\|P_H(x) - P_H(y)\| \leq \|x - y\|$ for all $x, y \in \mathbb{R}^n$.

Finally, based on the concepts discussed above, we propose a hybrid modified PRP-HS-LS-type CG projection algorithm (Abbr. Algorithm MPHL). The step-by-step procedure of the proposed algorithm for solving the problem (1.1) is outlined below.

Algorithm MPHL

```
Input: An initial guess $x_0 \in \mathbb{R}^n$ and parameters $\beta > 0, 0 < \rho < 1, \sigma, \hat{t}, \mu, \epsilon > 0$, with $k := 0$.
while $||h_k|| > \epsilon$ do
    Compute $\beta_k^{\text{MPHL}}$ and $\theta_k^{\text{MPHL}}$, and determine the search direction $d_k$ by (1.5).
    Compute the step size $\alpha_k$ by (2.1) and calculate the trail point $z_k$.
    if $z_k \in H$ and $||h(z_k)|| < \epsilon$ then
        Set $x_k := z_k$, and break.
    else
        Update the next iteration by
            $x_{k+1} = P_H\left[x_k - \gamma\chi_k h(z_k)\right], \ \chi_k = \frac{h(z_k)^{\text{T}}(x_k - z_k)}{||h(z_k)||^2}, \gamma \in (0, 2). \qquad (2.2)$
    end if
    Increment $k := k + 1$.
end while
Output: Solution $x_k$.
```

3. Analysis of algorithmic properties

3.1. Properties of search direction

The following lemma demonstrates that the search direction d_k satisfies two key properties: sufficient descent and trust region characteristics.

Lemma 1. *If Algorithm MPHL generates a sequence $\{d_k\}$, then the search direction d_k satisfies the following conditions:*

$$h_k^T d_k \leq -a_1 ||h_k||^2, \tag{3.1}$$

$$a_1 ||h_k|| \leq ||d_k|| \leq a_2 ||h_k||, \tag{3.2}$$

where $a_1 = 1 - (1 + \hat{t})^2/4$ and $a_2 = 1 + 1/\mu + 1/\mu^2 + \hat{t}/\mu$.

Proof. For $k = 0$, (1.5) yields that (3.1) and (3.2) are obviously satisfied. For $k \geq 1$, multiplying both sides of (1.5) by h_k^{T}, we obtain:

$$\begin{aligned}
h_k^{\text{T}} d_k &= -||h_k||^2 + \beta_k^{\text{MPHL}} h_k^{\text{T}} d_{k-1} + \theta_k^{\text{MPHL}} h_k^{\text{T}} y_{k-1} \\
&= -||h_k||^2 + \frac{(1+t_k) h_k^{\text{T}} y_{k-1} h_k^{\text{T}} d_{k-1}}{\delta_k} - \frac{||y_{k-1}||^2 (h_k^{\text{T}} d_{k-1})^2}{\delta_k^2} \\
&= -||h_k||^2 + 2\left(\frac{1+t_k}{2} h_k^{\text{T}}\right) \frac{y_{k-1} h_k^{\text{T}} d_{k-1}}{\delta_k} - \frac{||y_{k-1}||^2 (h_k^{\text{T}} d_{k-1})^2}{\delta_k^2} \\
&\leq -||h_k||^2 + \frac{(1+t_k)^2 ||h_k||^2}{4} + \frac{||y_{k-1}||^2 (h_k^{\text{T}} d_{k-1})^2}{\delta_k^2} - \frac{||y_{k-1}||^2 (h_k^{\text{T}} d_{k-1})^2}{\delta_k^2} \\
&= -a_1 ||h_k||^2,
\end{aligned}$$

where the inequality holds with the relationship $2a^{\text{T}} b \leq ||a||^2 + ||b||^2$. This shows that (3.1) holds.

Next, using the Cauchy-Schwarz inequality, (3.1) can be simplified to obtain $\|d_k\| \geq a_1\|h_k\|$. From the definition of β_k^{MPHL}, we have:

$$\left|\beta_k^{MPHL}\right| \leq \frac{\|h_k\|\|y_{k-1}\|}{\delta_k} + \frac{\|y_{k-1}\|^2\|h_k\|\|d_{k-1}\|}{\delta_k^2} \leq \frac{\|h_k\|\|y_{k-1}\|}{\mu\|d_{k-1}\|\|y_{k-1}\|} + \frac{\|y_{k-1}\|^2\|h_k\|\|d_{k-1}\|}{(\mu\|d_{k-1}\|\|y_{k-1}\|)^2} = \left(\frac{1}{\mu} + \frac{1}{\mu^2}\right)\frac{\|h_k\|}{\|d_{k-1}\|}.$$

Besides, from the definition of θ_k^{MPHL}, we have:

$$\left|\theta_k^{\text{MPHL}}\right| \leq \frac{\hat{t}\|h_k\|\|d_{k-1}\|}{\mu\|d_{k-1}\|\|y_{k-1}\|} = \frac{\hat{t}\|h_k\|}{\mu\|y_{k-1}\|}.$$

Together with the above inequalities, (1.5) yields:

$$\begin{aligned}\|d_k\| &= \left\|-h_k + \beta_k^{\text{MPHL}} d_{k-1} + \theta_k^{\text{TPRP}} y_{k-1}\right\| \\ &\leq \|h_k\| + \left|\beta_k^{\text{MPHL}}\right| \|d_{k-1}\| + \left|\theta_k^{\text{TPRP}}\right| \|y_{k-1}\| \\ &\leq \|h_k\| + \left(\tfrac{1}{\mu} + \tfrac{1}{\mu^2}\right)\tfrac{\|h_k\|}{\|d_{k-1}\|}\|d_{k-1}\| + \tfrac{\hat{t}\|h_k\|}{\mu\|y_{k-1}\|}\|y_{k-1}\| \\ &= a_2\|h_k\|.\end{aligned}$$

This shows that (3.2) holds. □

3.2. Properties of convergence

To systematically analyze the global convergence of Algorithm MPHL, we first introduce some general assumptions. Throughout the analysis, we assume that the sequences $\{x_k\}$ and $\{z_k\}$ generated by Algorithm MPHL are infinite. Otherwise, the last iteration is the solution of the problem (1.1). To provide a comprehensive understanding of the global convergence, we begin by describing these general assumptions:

(S1) The solution set of the problem (1.1) is non-empty, which is critical for ensuring that the optimization problem is well-posed and that the existence of solutions can be guaranteed under typical scenarios encountered in nonlinear systems.

(S2) The function $h(x)$ is monotone, meaning that $(h(x) - h(y))^{\text{T}}(x - y) \geq 0$ holds for any $x, y \in \mathbb{R}^n$.

The following lemma plays a pivotal role in establishing the global convergence of Algorithm MPHL, providing the foundation for the subsequent analysis and results.

Lemma 2. *Let sequences $\{x_k\}$, $\{z_k\}$, and $\{d_k\}$ be generated by Algorithm MPHL. Then the following statements are true:*

(i) *There exists a step size α_k such that the line search mechanism* (2.1) *is satisfied for all k;*

(ii) *The sequence $\{x_k\}$ is bounded;*

(iii) *As the iteration number k approaches infinity, we have $\lim_{k\to\infty} \alpha_k\|d_k\| = 0$, meaning that the product of the step size α_k and the norm of the search direction d_k tends to zero.*

Proof. First, we aim to prove that statement (i) holds. We proceed by using a proof by contradiction. Assume that the line search mechanism (2.1) does not hold. In other words, there exists a constant $\tilde{k} > 0$ such that the following inequality is satisfied for any integer $i > 0$:

$$-h(x_{\tilde{k}} + \beta\rho^i d_{\tilde{k}})^{\text{T}} d_{\tilde{k}} < \sigma\beta\rho^i\|h(x_{\tilde{k}} + \beta\rho^i d_{\tilde{k}})\|\|d_{\tilde{k}}\|^2.$$

By leveraging the continuity of h, we can analyze the behavior of the inequality as $i \to \infty$. Specifically, we note that as i increases, the term $\beta\rho^i d_{\bar{k}}$ tends to dominate the argument of the function h, leading to a more pronounced influence of h at points increasingly distant from $x_{\bar{k}}$. This suggests that the values of $h(x_{\bar{k}} + \beta\rho^i d_{\bar{k}})$ will converge to a certain limit determined by the continuity of h and the direction of $d_{\bar{k}}$. Taking the limit as $i \to \infty$, we obtain the following expression:

$$-h(x_{\bar{k}})^{\mathrm{T}} d_{\bar{k}} \le 0.$$

On the other hand, from a previously established result given by (3.1), we know that:

$$-h(x_{\bar{k}})^{\mathrm{T}} d_{\bar{k}} \ge a_1 \|h_{\bar{k}}\|^2,$$

Therefore, we conclude that statement (i) is indeed valid.

Next, we aim to prove that statement (ii) holds. To facilitate the analysis, we begin by two key inequalities. From Assumption (S2), we have the following:

$$h(z_k)^{\mathrm{T}}(x_k - x_*) = h(z_k)^{\mathrm{T}}(x_k - z_k + z_k - x_*) \ge h(z_k)^{\mathrm{T}}(x_k - z_k), \tag{3.3}$$

where x_* denotes an optimal solution such that $h(x_*) = 0$. From the definition of z_k and the condition given by (2.1), we have:

$$h(z_k)^{\mathrm{T}}(x_k - z_k) = -\alpha_k h(z_k)^{\mathrm{T}} d_k \ge \sigma\alpha_k^2 \|d_k\|^2 = \sigma\|x_k - z_k\|^2. \tag{3.4}$$

Now, by utilizing the projection operator $P_H[x]$, the definition of χ_k, and combining (3.3), and (3.4), we can derive the following inequality:

$$\begin{aligned} \|x_{k+1} - x_*\|^2 &\le \|x_k - \gamma\chi_k h(z_k) - x_*\|^2 \\ &= \|x_k - x_*\|^2 - 2\gamma\chi_k h(z_k)^{\mathrm{T}}(x_k - x_*) + \gamma^2\chi_k^2\|h(z_k)\|^2 \\ &\le \|x_k - x_*\|^2 - \gamma(2-\gamma)\chi_k^2\|h(z_k)\|^2 \\ &\le \|x_k - x_*\|^2 - \gamma(2-\gamma)\sigma^2\|x_k - z_k\|^4, \end{aligned} \tag{3.5}$$

where the above equality holds for the sum of squares. The inequality (3.5) shows that the norm $\{\|x_k - x_*\|\}$ decreases at each iteration. Thus, we can conclude that the sequence $\{x_k - x_*\}$ is decreasing and converging to 0, implying that $\{x_k\}$ is bounded and converges to the optimal solution x_*. Therefore, we conclude that statement (ii) is indeed valid.

Finally, we proceed to prove that statement (iii) holds. Starting with the inequality in (3.5), we reorganize it to derive the following expression:

$$\gamma(2-\gamma)\sigma^2 \sum_{k=0}^{\infty} \|x_k - z_k\|^4 \le \sum_{k=0}^{\infty} \left(\|x_k - x_*\|^2 - \|x_{k+1} - x_*\|^2\right) \le \|x_0 - x_*\|^2.$$

This inequality implies that the series $\sum_{k=0}^{\infty} \|x_k - z_k\|^4$ is bounded and convergent. Since $\|x_k - z_k\|^4 \ge 0$, we can conclude that $\lim_{k\to\infty} \|x_k - z_k\|^4 = 0$. This, in turn, implies that $\lim_{k\to\infty} \|x_k - z_k\| = 0$. From the definition of z_k, we know that $z_k = x_k + \alpha_k d_k$. Hence, the convergence of $\|x_k - z_k\|$ to 0 implies $\lim_{k\to\infty} \alpha_k\|d_k\| = 0$. □

The global convergence properties of Algorithm MPHL are outlined in the following theorem.

Theorem 1. *Let the sequence $\{h_k\}$ be generated by Algorithm MPHL. Then, the sequence satisfies the following convergence property:*

$$\lim_{k\to\infty} \|h_k\| = 0.$$

Proof. We begin by proceeding with a proof by contradiction. For the sake of contradiction, assume that there exists a positive constant ϑ such that $\|h_k\| > \vartheta$ holds for all $k \geq 0$. This assumption, combined with the result from (3.2), implies that $\|d_k\| \geq a_1\vartheta$. Moreover, due to the continuity of the function $h(x)$ and the boundedness of the sequence $\{x_k\}$, it follows that the sequence $\{h_k\}$ is also bounded. This means that there exists another positive constant ι such that $\|h_k\| \leq \iota$. Utilizing this, together with (3.2), we can deduce that $\|d_k\| \leq a_2\iota$. Consequently, the sequence $\{d_k\}$ is bounded, which implies that $\lim\limits_{k\to\infty} \alpha_k = 0$ with Lemma 2 (iii).

Since both sequences $\{x_k\}$ and $\{d_k\}$ are bounded, there exists an infinite index set Γ such that:

$$\lim_{j\to\infty, j\in\Gamma} x_{k_j} = \acute{x} \qquad \lim_{j\to\infty, j\in\Gamma} d_{k_j} = \acute{d}.$$

Next, we consider the line search mechanism (2.1), which gives the inequality:

$$-h(x_{k_j} + \rho^{-1}\alpha_{k_j} d_{k_j})^{\mathrm{T}} d_{k_j} < \sigma\rho^{-1}\alpha_{k_j}\|h(x_{k_j} + \rho^{-1}\alpha_{k_j} d_{k_j})\|\|d_{k_j}\|^2.$$

Taking the limit as $j \to \infty$, we arrive at $-h(\acute{x})^{\mathrm{T}}\acute{d} \leq 0$. Furthermore, from the inequality derived in (3.1), we have:

$$h_{k_j}^{\mathrm{T}} d_{k_j} \leq -a_1\|h_{k_j}\|^2.$$

Taking the limit as $j \to \infty$, we conclude that $-h(\acute{x})^{\mathrm{T}}\acute{d} \geq a_1\|h(\acute{x})\| > 0$. This directly contradicts the earlier result that $-h(\acute{x})^{\mathrm{T}}\acute{d} \leq 0$. Therefore, the assumption that $\|h_k\| > \vartheta$ for all $k \geq 0$ must be false, and we conclude that $\lim\limits_{k\to\infty} \|h_k\| = 0$. □

4. Numerical experiment for constrained nonlinear systems of equations

4.1. General setups

This section focuses on applying the Algorithm MPHL to a set of widely recognized test problems and comparing its performance with three existing algorithms: Algorithm PSGM [22], Algorithm PDY [29], and Algorithm PCG [30]. The experiments were conducted on a system running Ubuntu 20.04.2 LTS 64-bit, powered by an Intel(R) Xeon(R) Gold 5115 2.40GHz CPU. This standardized computational environment ensures the reliability and consistency of the results, enabling a fair comparison between the algorithms.

The parameter settings for Algorithm PSGM, Algorithm PDY, and Algorithm PCG are configured according to their respective references. For Algorithm MPHL, the parameters are set as follows: $\beta = 1$, $\rho = 0.74$, $\sigma = 10^{-4}$, $\gamma = 1.3$, $\hat{t} = 10^3$, $\mu = 2$, $\epsilon = 10^{-6}$. The test problems are formulated in the form $h(x) = (h_1(x), h_2(x), \ldots, h_n(x))^{\mathrm{T}}$ with the variable $x = (x_1, x_2, \ldots, x_n)$. For each test problem, the dimension n is set to [10,000 50,000 100,000 150,000 200,000]. The initial points are selected as follows: $x_1 = (1, 1, \ldots, 1)^{\mathrm{T}}$, $x_2 = (0.1, 0.1, \ldots, 0.1)^{\mathrm{T}}$, $x_3 = (\frac{1}{2}, \frac{1}{2^2}, \ldots, \frac{1}{2^n})^{\mathrm{T}}$, $x_4 = (2, 2, \ldots, 2)^{\mathrm{T}}$, $x_5 = (1, \frac{1}{2}, \ldots, \frac{1}{n})^{\mathrm{T}}$, $x_6 = (\frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n})^{\mathrm{T}}$, and $x_7 = (\frac{n-1}{n}, \frac{n-2}{n}, \ldots, \frac{n-n}{n})^{\mathrm{T}}$. Specifically, each algorithm is terminated when $\|h_k\| \leq \epsilon$ or the number of iterations exceeded 2000. The following test problems are considered:

Problem 1. Set

$$\begin{aligned} h_1(x) &= e^{x_1} - 1, \\ h_i(x) &= e^{x_i} + x_i - 1, \quad \text{for } i = 2, 3, \ldots, n, \end{aligned}$$

and the constraint set $H = \mathbb{R}^n_+$.

Problem 2. Set

$$h_i(x) = 2x_i - \sin(|x_i|), \quad \text{for } i = 1, 2, \ldots, n,$$

and the constraint set $H = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i \leq n, \ x_i > -1, \ i = 1, 2, \ldots, n\}$.

Problem 3. Set

$$h_i(x) = (e^{x_i})^2 + 3\sin(x_i)\cos(x_i) - 1, \quad \text{for } i = 1, 2, \ldots, n,$$

and the constraint set $H = \mathbb{R}^n_+$.

Problem 4. Set

$$h_i(x) = \frac{1}{n}e^{x_i} - 1, \quad \text{for } i = 1, 2, \ldots, n,$$

and the constraint set $H = \mathbb{R}^n_+$.

Problem 5. Set

$$h_i(x) = x_i - 2\sin(|x_i - 1|), \quad \text{for } i = 1, 2, \ldots, n,$$

and the constraint set $H = \mathbb{R}^n_+$.

Problem 6. Set

$$h_i(x) = \ln(|x_i| + 1) - \frac{x_i}{n}, \quad \text{for } i = 1, 2, \ldots, n,$$

and the constraint set $H = \mathbb{R}^n_+$.

Problem 7. Set

$$h_i(x) = 2x_i - \sin(|x_i|), \quad \text{for } i = 1, 2, \ldots, n,$$

and the constraint set $H = \mathbb{R}^n_+$.

4.2. Numerical reports

The numerical results of the test problems, solved by these four algorithms, are presented in Tables 1–7. In these tables, “Init(n)” denotes the initial points and the dimension multiplied by 1000. “CPUT” denotes the CPU time in seconds, “NF” denotes the number of function evaluations, and “NIter” denotes the number of iterations. These tables demonstrate that all four algorithms are capable of solving the test problems under different initial points and dimensions. The results clearly show that each of the four algorithms successfully solved the test problems. To visually assess the performance of the four algorithms, we employed the performance profiles technique [31] to plot performance curves based on CPUT, NF, and NIter. In these plots, the higher the curve, the better the corresponding algorithm performs. As shown in Figures 1–3, Algorithm MPHL outperformed the others in a significant portion of the test cases, winning approximately 52%, 51%, and 48% of the test problems in terms of CPUT, NF, and NIter, respectively. The performance curves of Algorithm

MPHL generally lie above those of Algorithm PSGM, Algorithm PDY, and Algorithm PCG, indicating its superior efficiency and effectiveness in solving these test problems.

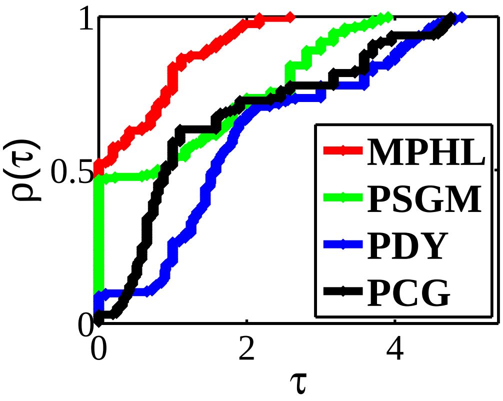

Figure 1. Performance profiles on NIter.

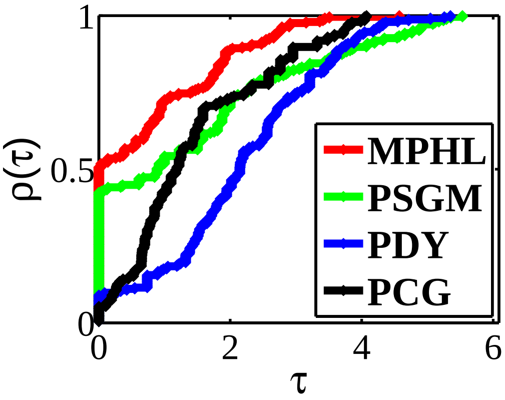

Figure 2. Performance profiles on NF.

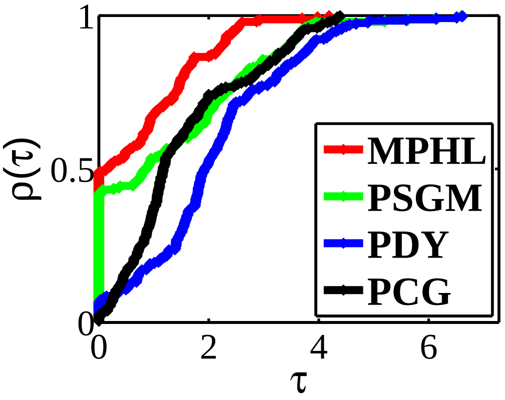

Figure 3. Performance profiles on CPUT.

Table 1. Numerical results for Problem 1.

Inti(n)	MPHL	PSGM	PDY	PCG
	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter
x1(10)	1.29e-02/7/1	9.05e-03/20/7	1.48e-02/73/18	1.89e-02/93/23
x2(10)	1.71e-03/6/1	3.30e-03/17/6	1.10e-02/65/16	1.75e-02/99/26
x3(10)	7.31e-04/4/1	2.86e-03/15/6	6.85e-03/37/12	5.32e-03/24/9
x4(10)	1.21e-03/8/1	5.28e-03/27/8	1.44e-02/77/18	1.97e-02/106/27
x5(10)	1.86e-03/11/2	4.25e-03/22/7	1.10e-02/65/16	2.06e-02/110/31
x6(10)	6.55e-03/40/8	3.74e-03/20/7	1.21e-02/69/17	1.62e-02/93/23
x7(10)	6.73e-03/40/8	4.02e-03/20/7	1.15e-02/69/17	1.77e-02/101/26
x1(50)	5.05e-03/7/1	1.94e-02/26/8	5.31e-02/79/19	6.74e-02/97/24
x2(50)	3.50e-03/6/1	1.40e-02/17/6	4.40e-02/65/16	7.09e-02/101/26
x3(50)	2.50e-03/4/1	9.78e-03/15/6	2.37e-02/37/12	1.78e-02/24/9
x4(50)	4.85e-03/8/1	2.82e-02/41/10	6.65e-02/102/22	7.61e-02/108/27
x5(50)	6.48e-03/11/2	1.52e-02/22/7	4.23e-02/65/16	8.06e-02/110/31
x6(50)	2.56e-02/40/8	1.47e-02/21/7	4.96e-02/73/18	6.58e-02/97/24
x7(50)	2.46e-02/40/8	1.46e-02/21/7	4.93e-02/73/18	6.82e-02/99/25
x1(100)	1.36e-02/7/1	5.37e-02/29/8	1.25e-01/86/20	1.60e-01/97/24
x2(100)	9.47e-03/6/1	3.27e-02/17/6	1.14e-01/69/17	1.57e-01/95/24
x3(100)	7.34e-03/4/1	2.85e-02/15/6	5.51e-02/37/12	4.11e-02/24/9
x4(100)	1.18e-02/8/1	7.70e-02/59/12	1.65e-01/120/25	1.65e-01/105/26
x5(100)	1.82e-02/11/2	3.49e-02/22/7	9.49e-02/65/16	1.71e-01/110/31
x6(100)	5.65e-02/40/8	3.35e-02/24/8	1.17e-01/79/19	1.57e-01/97/24
x7(100)	5.55e-02/40/8	4.01e-02/24/8	1.12e-01/79/19	1.41e-01/100/25
x1(150)	2.96e-02/7/1	1.86e-01/38/9	4.44e-01/97/22	4.96e-01/101/25
x2(150)	2.34e-02/6/1	8.58e-02/17/6	3.30e-01/69/17	4.90e-01/99/25
x3(150)	1.77e-02/4/1	7.90e-02/15/6	1.84e-01/37/12	1.40e-01/24/9
x4(150)	2.95e-02/8/1	3.10e-01/73/13	6.31e-01/144/28	5.46e-01/109/27
x5(150)	4.58e-02/11/2	1.04e-01/22/7	3.06e-01/65/16	5.84e-01/110/31
x6(150)	1.77e-01/40/8	1.21e-01/25/8	3.70e-01/79/19	4.82e-01/97/24
x7(150)	1.79e-01/40/8	1.21e-01/25/8	3.73e-01/79/19	4.82e-01/97/24
x1(200)	3.51e-02/7/1	2.71e-01/45/10	6.46e-01/103/23	6.93e-01/101/25
x2(200)	3.34e-02/6/1	1.16e-01/17/6	4.34e-01/69/17	6.73e-01/99/25
x3(200)	2.33e-02/4/1	1.01e-01/15/6	2.45e-01/37/12	1.87e-01/24/9
x4(200)	3.99e-02/8/1	5.16e-01/93/15	9.41e-01/160/30	7.14e-01/106/26
x5(200)	6.20e-02/11/2	1.40e-01/22/7	4.12e-01/65/16	7.56e-01/110/31
x6(200)	2.34e-01/40/8	1.71e-01/28/8	5.08e-01/83/20	6.59e-01/101/25
x7(200)	2.30e-01/40/8	1.65e-01/28/8	5.09e-01/83/20	6.58e-01/101/25

Table 2. Numerical results for Problem 2.

Inti(n)	MPHL	PSGM	PDY	PCG
	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter
x1(10)	4.81e-03/34/8	2.40e-03/19/7	2.44e-03/19/6	5.56e-03/36/10
x2(10)	5.01e-03/33/8	2.16e-03/16/6	1.13e-02/68/17	6.95e-03/36/10
x3(10)	8.54e-03/62/13	2.57e-03/16/6	8.70e-03/72/18	4.86e-03/39/11
x4(10)	4.90e-03/31/8	1.80e-01/24/7	9.16e-02/81/20	6.05e-03/38/11
x5(10)	9.07e-03/67/15	3.29e-03/20/8	9.95e-03/72/18	5.96e-03/48/14
x6(10)	7.79e-03/58/12	2.53e-03/18/7	8.97e-03/76/19	7.77e-03/54/15
x7(10)	7.23e-03/58/12	2.37e-03/18/7	9.62e-03/76/19	8.26e-03/54/15
x1(50)	1.62e-02/34/8	1.06e-02/19/7	3.37e-02/77/19	1.87e-02/40/11
x2(50)	1.71e-02/37/9	7.60e-03/16/6	3.03e-02/72/18	1.82e-02/39/11
x3(50)	2.58e-02/59/13	7.77e-03/16/6	3.23e-02/72/18	1.93e-02/39/11
x4(50)	1.72e-02/35/9	8.43e-01/28/7	8.30e-01/82/20	2.03e-02/41/12
x5(50)	2.80e-02/68/14	1.02e-02/20/8	3.04e-02/72/18	2.36e-02/50/14
x6(50)	2.87e-02/67/14	9.81e-03/18/7	3.33e-02/80/20	2.46e-02/54/15
x7(50)	2.54e-02/63/13	9.15e-03/18/7	3.57e-02/80/20	2.84e-02/54/15
x1(100)	2.73e-02/34/8	1.56e-02/21/7	5.63e-02/77/19	3.32e-02/43/12
x2(100)	2.86e-02/37/9	1.68e-02/19/7	5.88e-02/77/19	3.22e-02/39/11
x3(100)	4.70e-02/63/14	1.71e-02/20/7	5.82e-02/77/19	3.13e-02/39/11
x4(100)	2.93e-02/35/9	2.67e+00/33/8	2.67e+00/86/21	3.27e-02/41/12
x5(100)	5.29e-02/69/15	1.88e-02/23/8	5.68e-02/77/19	3.97e-02/53/15
x6(100)	4.71e-02/67/14	1.61e-02/19/7	5.73e-02/77/19	4.12e-02/54/15
x7(100)	4.71e-02/67/14	1.70e-02/19/7	5.68e-02/77/19	4.02e-02/54/15
x1(150)	7.15e-02/34/8	4.41e-02/23/7	1.53e-01/81/20	9.95e-02/43/12
x2(150)	7.75e-02/37/9	4.18e-02/20/7	1.47e-01/77/19	9.32e-02/39/11
x3(150)	1.19e-01/60/13	4.24e-02/21/7	1.53e-01/81/20	9.25e-02/39/11
x4(150)	7.86e-02/35/9	5.49e+00/36/8	5.54e+00/88/21	1.01e-01/41/12
x5(150)	1.39e-01/68/15	7.11e-02/41/10	1.58e-01/81/20	1.31e-01/53/15
x6(150)	1.34e-01/67/14	4.04e-02/19/7	1.48e-01/77/19	1.29e-01/54/15
x7(150)	1.30e-01/67/14	4.42e-02/19/7	1.50e-01/77/19	1.28e-01/54/15
x1(200)	1.00e-01/38/9	5.80e-02/23/7	2.15e-01/87/21	1.30e-01/43/12
x2(200)	9.95e-02/37/9	5.26e-02/21/7	1.98e-01/81/20	1.22e-01/39/11
x3(200)	1.51e-01/59/13	5.70e-02/22/7	2.09e-01/86/21	1.20e-01/39/11
x4(200)	9.68e-02/35/9	9.33e+00/37/8	9.49e+00/93/22	1.31e-01/41/12
x5(200)	1.64e-01/63/14	6.12e-02/24/8	2.14e-01/86/21	1.64e-01/53/15
x6(200)	1.67e-01/67/14	5.29e-02/20/7	1.92e-01/77/19	1.66e-01/54/15
x7(200)	1.67e-01/67/14	5.19e-02/20/7	1.95e-01/77/19	1.66e-01/54/15

Table 3. Numerical results for Problem 3.

Inti(n)	MPHL	PSGM	PDY	PCG
	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter
x1(10)	3.83e-03/6/1	3.48e-03/25/7	7.98e-03/66/13	7.49e-03/67/13
x2(10)	1.08e-03/9/1	2.73e-03/21/6	7.02e-03/61/12	6.52e-03/56/11
x3(10)	4.17e-04/3/1	3.54e-04/3/1	3.41e-04/3/1	3.55e-04/3/1
x4(10)	6.28e-04/4/1	4.96e-03/37/9	9.76e-03/80/15	1.59e-03/11/2
x5(10)	1.85e-03/19/2	3.45e-03/29/8	9.22e-03/84/16	7.17e-03/57/11
x6(10)	4.44e-02/240/19	1.03e-02/85/31	8.82e-03/83/16	7.99e-03/78/15
x7(10)	1.15e-02/132/16	1.00e-02/86/29	8.75e-03/83/16	8.30e-03/78/15
x1(50)	2.80e-03/6/1	1.54e-02/30/8	3.56e-02/81/15	2.91e-02/67/13
x2(50)	3.10e-03/9/1	1.11e-02/21/6	2.87e-02/66/13	2.68e-02/61/12
x3(50)	1.39e-03/3/1	1.19e-03/3/1	1.21e-03/3/1	1.46e-03/3/1
x4(50)	2.10e-03/4/1	2.27e-02/43/10	4.76e-02/105/18	5.63e-03/11/2
x5(50)	6.71e-03/19/2	1.22e-02/29/8	3.57e-02/84/16	2.39e-02/57/11
x6(50)	6.34e-02/201/15	1.18e-02/27/8	3.41e-02/78/15	3.42e-02/78/15
x7(50)	8.85e-02/293/23	1.21e-02/27/8	3.35e-02/78/15	3.30e-02/78/15
x1(100)	6.80e-03/6/1	2.91e-02/35/8	6.57e-02/81/15	5.09e-02/67/13
x2(100)	5.98e-03/9/1	1.89e-02/21/6	5.09e-02/66/13	4.37e-02/61/12
x3(100)	3.00e-03/3/1	2.50e-03/3/1	3.22e-03/3/1	3.07e-03/3/1
x4(100)	5.27e-03/4/1	4.82e-02/53/11	1.08e-01/132/21	1.04e-02/11/2
x5(100)	1.15e-02/19/2	2.22e-02/29/8	6.27e-02/84/16	4.03e-02/57/11
x6(100)	4.14e-01/298/15	2.27e-02/29/8	6.19e-02/78/15	6.13e-02/78/15
x7(100)	1.09e-01/200/15	2.48e-02/29/8	5.96e-02/78/15	5.77e-02/78/15
x1(150)	1.18e-02/6/1	7.72e-02/40/9	1.71e-01/94/17	1.47e-01/72/14
x2(150)	1.29e-02/9/1	4.32e-02/21/6	1.21e-01/66/13	1.24e-01/61/12
x3(150)	6.34e-03/3/1	4.55e-03/3/1	5.72e-03/3/1	7.47e-03/3/1
x4(150)	9.20e-03/4/1	1.23e-01/68/13	2.73e-01/153/23	2.39e-02/11/2
x5(150)	2.90e-02/19/2	5.44e-02/29/8	1.55e-01/84/16	1.14e-01/57/11
x6(150)	7.60e-01/319/24	5.86e-02/30/8	1.53e-01/83/16	1.69e-01/83/16
x7(150)	4.19e-01/341/22	5.51e-02/30/8	1.50e-01/83/16	1.66e-01/83/16
x1(200)	1.42e-02/6/1	9.94e-02/44/9	2.54e-01/112/19	1.85e-01/72/14
x2(200)	1.67e-02/9/1	5.61e-02/21/6	1.56e-01/66/13	1.58e-01/61/12
x3(200)	7.95e-03/3/1	6.22e-03/3/1	6.66e-03/3/1	8.94e-03/3/1
x4(200)	1.22e-02/4/1	1.85e-01/80/14	3.66e-01/163/24	3.00e-02/11/2
x5(200)	3.35e-02/19/2	6.82e-02/29/8	1.89e-01/84/16	1.33e-01/57/11
x6(200)	1.17e+00/738/48	7.43e-02/31/8	2.24e-01/97/18	2.13e-01/83/16
x7(200)	4.37e-01/276/19	7.18e-02/31/8	2.22e-01/97/18	2.09e-01/83/16

Table 4. Numerical results for Problem 4.

Inti(n)	MPHL	PSGM	PDY	PCG
	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter
x1(10)	5.82e-03/29/14	1.07e-02/117/14	1.11e-02/84/27	6.53e-03/40/18
x2(10)	5.52e-03/29/14	1.26e-02/140/15	1.17e-02/87/28	8.87e-03/50/22
x3(10)	8.45e-03/46/20	1.28e-02/146/15	1.18e-02/87/28	7.33e-03/45/20
x4(10)	5.40e-03/27/13	9.04e-03/97/13	1.02e-02/77/25	7.36e-03/46/20
x5(10)	7.03e-03/42/17	1.33e-02/150/17	1.16e-02/87/28	7.36e-03/47/21
x6(10)	8.39e-03/46/20	1.16e-02/130/15	1.09e-02/84/27	7.51e-03/45/20
x7(10)	8.00e-03/46/20	1.21e-02/130/15	1.19e-02/84/27	7.35e-03/45/20
x1(50)	2.41e-02/33/16	1.52e-01/479/30	1.02e-01/219/52	3.47e-02/52/23
x2(50)	2.32e-02/33/16	1.74e-01/554/31	1.10e-01/235/55	3.60e-02/54/24
x3(50)	2.82e-02/42/18	1.81e-01/573/33	1.12e-01/239/56	3.62e-02/54/24
x4(50)	2.20e-02/31/15	1.29e-01/403/26	9.46e-02/199/48	3.42e-02/50/22
x5(50)	3.29e-02/49/21	1.79e-01/573/33	1.17e-01/239/56	3.68e-02/54/24
x6(50)	2.99e-02/47/19	1.68e-01/530/33	1.06e-01/228/54	3.59e-02/54/24
x7(50)	3.03e-02/47/19	1.76e-01/530/33	1.06e-01/228/54	3.63e-02/54/24
x1(100)	3.97e-02/31/15	4.66e-01/851/42	2.02e-01/264/60	5.53e-02/52/23
x2(100)	4.40e-02/35/17	5.26e-01/950/45	2.19e-01/284/64	6.03e-02/54/24
x3(100)	5.17e-02/41/19	5.32e-01/952/46	2.17e-01/284/64	6.01e-02/56/25
x4(100)	3.83e-02/31/15	4.14e-01/736/39	1.92e-01/244/56	5.35e-02/50/22
x5(100)	5.33e-02/47/20	5.39e-01/956/48	2.29e-01/284/64	6.37e-02/56/25
x6(100)	5.51e-02/49/20	5.09e-01/897/45	2.11e-01/274/62	6.23e-02/59/26
x7(100)	5.63e-02/49/20	5.01e-01/897/45	2.14e-01/274/62	6.56e-02/59/26
x1(150)	1.14e-01/33/16	1.40e+00/1140/50	5.44e-01/281/63	1.85e-01/54/24
x2(150)	1.12e-01/33/16	1.58e+00/1291/54	5.84e-01/301/67	1.88e-01/56/25
x3(150)	1.35e-01/44/19	1.64e+00/1341/60	5.85e-01/301/67	1.83e-01/53/24
x4(150)	1.14e-01/33/16	1.25e+00/1011/48	5.09e-01/261/59	1.76e-01/52/23
x5(150)	1.42e-01/46/19	1.67e+00/1346/63	5.85e-01/301/67	1.91e-01/56/25
x6(150)	1.55e-01/49/21	1.54e+00/1234/56	6.21e-01/295/66	2.11e-01/56/25
x7(150)	1.70e-01/49/21	1.58e+00/1234/56	6.14e-01/295/66	2.11e-01/56/25
x1(200)	1.69e-01/35/17	2.29e+00/1432/61	1.32e+00/534/100	2.45e-01/51/23
x2(200)	1.67e-01/35/17	2.59e+00/1621/66	1.40e+00/577/107	2.55e-01/53/24
x3(200)	2.02e-01/45/20	2.59e+00/1625/66	1.41e+00/582/108	2.66e-01/56/25
x4(200)	1.58e-01/33/16	2.02e+00/1249/56	1.19e+00/486/92	2.37e-01/49/22
x5(200)	2.07e-01/48/20	2.58e+00/1625/66	1.41e+00/582/108	2.63e-01/56/25
x6(200)	2.15e-01/52/21	2.46e+00/1548/66	1.36e+00/558/104	2.60e-01/56/25
x7(200)	2.18e-01/52/21	2.47e+00/1548/66	1.36e+00/558/104	2.60e-01/56/25

Table 5. Numerical results for Problem 5.

Inti(n)	MPHL	PSGM	PDY	PCG
	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter
x1(10)	9.00e-03/67/13	3.54e-03/18/6	5.10e-03/41/10	7.26e-03/41/10
x2(10)	8.47e-03/67/13	2.83e-03/19/6	1.52e-02/125/25	5.33e-03/45/11
x3(10)	1.39e-02/127/22	2.69e-03/19/6	1.35e-02/125/25	5.40e-03/45/11
x4(10)	6.76e-03/50/11	3.86e-03/38/7	6.39e-03/51/13	5.78e-03/46/12
x5(10)	1.74e-02/121/21	4.00e-03/21/7	1.44e-02/125/25	5.65e-03/45/11
x6(10)	1.81e-02/133/23	3.16e-03/21/7	1.59e-02/120/24	5.14e-03/41/10
x7(10)	1.55e-02/134/23	2.68e-03/21/7	1.32e-02/120/24	4.78e-03/41/10
x1(50)	2.80e-02/67/13	8.72e-03/18/6	2.11e-02/45/11	1.87e-02/41/10
x2(50)	2.76e-02/67/13	9.55e-03/20/6	5.25e-02/130/26	2.00e-02/45/11
x3(50)	5.02e-02/133/23	1.01e-02/20/6	5.50e-02/130/26	1.96e-02/45/11
x4(50)	2.33e-02/55/12	1.77e-02/46/8	2.64e-02/56/14	2.35e-02/50/13
x5(50)	5.70e-02/145/25	1.23e-02/24/8	5.66e-02/130/26	2.17e-02/45/11
x6(50)	6.02e-02/158/27	1.03e-02/21/7	5.03e-02/125/25	2.01e-02/45/11
x7(50)	5.61e-02/144/25	1.12e-02/21/7	5.18e-02/125/25	2.04e-02/45/11
x1(100)	4.91e-02/72/14	1.52e-02/19/6	3.60e-02/45/11	3.17e-02/41/10
x2(100)	4.69e-02/67/13	1.68e-02/21/6	4.47e-02/55/13	3.45e-02/45/11
x3(100)	9.30e-02/145/25	2.10e-02/26/7	1.01e-01/142/28	3.46e-02/45/11
x4(100)	4.10e-02/55/12	5.26e-02/91/9	4.85e-02/63/15	4.02e-02/50/13
x5(100)	8.30e-02/127/22	2.52e-02/28/8	1.12e-01/142/28	3.29e-02/45/11
x6(100)	1.03e-01/157/27	1.72e-02/21/7	9.35e-02/130/26	3.37e-02/45/11
x7(100)	8.95e-02/138/24	1.91e-02/21/7	9.33e-02/130/26	3.38e-02/45/11
x1(150)	1.38e-01/72/14	4.67e-02/22/7	8.85e-02/45/11	1.02e-01/45/11
x2(150)	1.35e-01/72/14	4.96e-02/26/7	2.62e-01/142/28	1.07e-01/45/11
x3(150)	2.24e-01/127/22	5.60e-02/31/8	2.55e-01/142/28	1.15e-01/49/12
x4(150)	1.17e-01/55/12	1.01e-01/63/10	1.42e-01/73/17	1.17e-01/50/13
x5(150)	2.82e-01/156/27	5.72e-02/31/8	2.69e-01/142/28	1.10e-01/49/12
x6(150)	2.73e-01/151/26	4.62e-02/22/7	2.35e-01/130/26	1.03e-01/45/11
x7(150)	2.48e-01/139/24	4.59e-02/22/7	2.34e-01/130/26	1.02e-01/45/11
x1(200)	1.73e-01/72/14	5.76e-02/22/7	1.25e-01/51/12	1.28e-01/45/11
x2(200)	1.71e-01/72/14	7.15e-02/31/8	3.22e-01/142/28	1.27e-01/45/11
x3(200)	3.11e-01/136/24	7.53e-02/34/8	3.35e-01/147/29	1.41e-01/49/12
x4(200)	1.39e-01/55/12	1.46e-01/80/11	3.89e-01/171/34	1.48e-01/50/13
x5(200)	3.76e-01/168/29	7.58e-02/34/8	3.62e-01/147/29	1.40e-01/49/12
x6(200)	3.26e-01/146/25	6.53e-02/25/8	2.56e-01/111/23	1.28e-01/45/11
x7(200)	3.11e-01/139/24	6.43e-02/25/8	2.59e-01/111/23	1.28e-01/45/11

Table 6. Numerical results for Problem 6.

Inti(n)	MPHL	PSGM	PDY	PCG
	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter
x1(10)	1.26e-03/5/2	8.11e-04/5/2	1.04e-03/5/2	1.87e-03/9/4
x2(10)	4.73e-04/3/1	2.35e-03/13/6	8.69e-04/5/2	8.99e-04/5/2
x3(10)	4.83e-04/3/1	1.48e-03/11/5	6.15e-04/5/2	1.19e-03/7/3
x4(10)	1.45e-03/7/3	1.51e-03/9/3	2.20e-03/13/4	2.26e-03/11/5
x5(10)	4.06e-03/19/9	2.11e-03/15/7	9.00e-04/5/2	4.46e-03/23/11
x6(10)	3.35e-03/17/8	2.55e-03/17/8	1.54e-03/9/4	4.54e-03/26/12
x7(10)	3.33e-03/17/8	3.76e-03/17/8	1.78e-03/9/4	4.76e-03/26/12
x1(50)	3.51e-03/5/2	5.59e-03/10/3	7.61e-03/13/4	7.31e-03/9/4
x2(50)	1.68e-03/3/1	9.88e-03/15/7	3.38e-03/5/2	3.67e-03/5/2
x3(50)	1.34e-03/3/1	5.17e-03/11/5	2.31e-03/5/2	4.16e-03/7/3
x4(50)	5.44e-03/7/3	1.51e-02/33/5	1.30e-02/23/6	8.56e-03/11/5
x5(50)	1.51e-02/19/9	8.58e-03/15/7	3.15e-03/5/2	1.82e-02/25/12
x6(50)	1.28e-02/17/8	9.79e-03/17/8	6.09e-03/9/4	2.27e-02/31/15
x7(50)	1.18e-02/17/8	1.06e-02/17/8	6.09e-03/9/4	2.27e-02/31/15
x1(100)	6.82e-03/5/2	1.88e-02/19/4	2.37e-02/23/6	1.32e-02/9/4
x2(100)	3.87e-03/3/1	1.64e-02/15/7	7.05e-03/5/2	6.41e-03/5/2
x3(100)	3.32e-03/3/1	8.66e-03/11/5	4.85e-03/5/2	7.66e-03/7/3
x4(100)	9.83e-03/7/3	4.79e-02/58/8	4.46e-02/44/10	1.67e-02/11/5
x5(100)	2.67e-02/19/9	1.49e-02/15/7	5.83e-03/5/2	3.20e-02/25/12
x6(100)	2.33e-02/17/8	1.96e-02/18/7	1.03e-02/10/3	3.81e-02/31/15
x7(100)	2.36e-02/17/8	1.77e-02/18/7	1.15e-02/10/3	4.02e-02/31/15
x1(150)	1.57e-02/5/2	4.97e-02/27/5	5.39e-02/23/6	3.27e-02/9/4
x2(150)	7.74e-03/3/1	3.97e-02/15/7	1.31e-02/5/2	1.69e-02/5/2
x3(150)	6.87e-03/3/1	2.48e-02/11/5	1.15e-02/5/2	2.25e-02/7/3
x4(150)	2.34e-02/7/3	1.24e-01/76/9	1.16e-01/55/12	4.06e-02/11/5
x5(150)	7.44e-02/19/9	3.80e-02/15/7	1.38e-02/5/2	9.39e-02/25/12
x6(150)	6.02e-02/17/8	5.58e-02/24/9	3.15e-02/13/4	1.14e-01/31/15
x7(150)	5.89e-02/17/8	5.37e-02/24/9	3.13e-02/13/4	1.16e-01/31/15
x1(200)	1.97e-02/5/2	8.34e-02/36/6	6.76e-02/23/6	4.10e-02/9/4
x2(200)	1.08e-02/3/1	5.24e-02/15/7	1.75e-02/5/2	2.16e-02/5/2
x3(200)	9.28e-03/3/1	3.23e-02/11/5	1.50e-02/5/2	2.82e-02/7/3
x4(200)	3.08e-02/7/3	2.06e-01/99/11	1.65e-01/61/13	5.13e-02/11/5
x5(200)	8.44e-02/19/9	4.59e-02/15/7	1.71e-02/5/2	1.16e-01/25/12
x6(200)	7.45e-02/17/8	7.16e-02/24/9	4.17e-02/13/4	1.44e-01/31/15
x7(200)	7.55e-02/17/8	6.93e-02/24/9	4.08e-02/13/4	1.45e-01/31/15

Table 7. Numerical results for Problem 7.

Inti(n)	MPHL	PSGM	PDY	PCG
	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter	CPUT/NF/NIter
x1(10)	6.77e-04/4/1	2.63e-03/16/7	7.65e-03/52/17	4.56e-03/31/12
x2(10)	5.96e-04/4/1	2.63e-03/16/7	6.01e-03/46/15	6.64e-03/29/11
x3(10)	4.26e-04/4/1	1.96e-03/14/6	4.85e-03/37/12	3.22e-03/24/9
x4(10)	5.83e-04/5/1	2.87e-03/23/8	8.58e-03/60/19	7.28e-03/34/13
x5(10)	5.62e-03/31/12	2.43e-03/16/7	6.40e-03/40/13	3.94e-03/26/10
x6(10)	5.51e-03/34/13	2.20e-03/16/7	6.86e-03/52/17	4.65e-03/31/12
x7(10)	5.90e-03/34/13	2.30e-03/16/7	6.69e-03/52/17	4.38e-03/31/12
x1(50)	1.97e-03/4/1	1.23e-02/22/8	2.92e-02/60/19	1.89e-02/31/12
x2(50)	1.69e-03/4/1	1.01e-02/16/7	2.55e-02/49/16	1.84e-02/31/12
x3(50)	1.41e-03/4/1	5.53e-03/14/6	1.46e-02/37/12	1.20e-02/24/9
x4(50)	2.19e-03/5/1	1.71e-02/39/10	3.73e-02/77/22	2.08e-02/34/13
x5(50)	1.84e-02/31/12	7.07e-03/16/7	2.09e-02/40/13	1.50e-02/26/10
x6(50)	2.07e-02/34/13	1.08e-02/19/8	2.55e-02/52/17	1.97e-02/34/13
x7(50)	2.09e-02/34/13	9.77e-03/19/8	2.59e-02/52/17	1.95e-02/34/13
x1(100)	4.28e-03/4/1	2.31e-02/28/9	5.79e-02/65/20	3.04e-02/31/12
x2(100)	3.55e-03/4/1	1.48e-02/16/7	4.37e-02/49/16	3.05e-02/31/12
x3(100)	3.02e-03/4/1	9.40e-03/14/6	2.62e-02/37/12	1.93e-02/24/9
x4(100)	4.09e-03/5/1	4.09e-02/60/12	7.24e-02/91/24	3.33e-02/34/13
x5(100)	3.10e-02/31/12	1.15e-02/16/7	3.38e-02/40/13	2.59e-02/26/10
x6(100)	3.40e-02/34/13	1.78e-02/21/8	4.88e-02/60/19	3.44e-02/36/14
x7(100)	3.50e-02/34/13	1.91e-02/21/8	5.08e-02/60/19	3.49e-02/36/14
x1(150)	8.52e-03/4/1	6.37e-02/34/9	1.62e-01/77/22	9.48e-02/31/12
x2(150)	8.49e-03/4/1	3.93e-02/16/7	1.11e-01/49/16	9.67e-02/31/12
x3(150)	7.68e-03/4/1	2.93e-02/14/6	7.29e-02/37/12	6.51e-02/24/9
x4(150)	9.60e-03/5/1	1.17e-01/74/13	2.08e-01/103/26	1.00e-01/34/13
x5(150)	8.68e-02/31/12	3.37e-02/16/7	8.95e-02/40/13	7.67e-02/26/10
x6(150)	9.75e-02/34/13	4.76e-02/23/8	1.36e-01/60/19	1.09e-01/36/14
x7(150)	9.45e-02/34/13	4.97e-02/23/8	1.31e-01/60/19	1.07e-01/36/14
x1(200)	1.05e-02/4/1	8.25e-02/37/9	2.09e-01/77/22	1.20e-01/31/12
x2(200)	1.00e-02/4/1	4.77e-02/16/7	1.40e-01/49/16	1.19e-01/31/12
x3(200)	8.83e-03/4/1	3.64e-02/14/6	9.53e-02/37/12	8.19e-02/24/9
x4(200)	1.14e-02/5/1	1.67e-01/85/14	2.92e-01/117/28	1.29e-01/34/13
x5(200)	1.11e-01/31/12	4.33e-02/16/7	1.15e-01/40/13	1.03e-01/26/10
x6(200)	1.23e-01/34/13	6.60e-02/23/8	1.89e-01/65/20	1.42e-01/36/14
x7(200)	1.25e-01/34/13	6.26e-02/23/8	1.84e-01/65/20	1.41e-01/36/14

5. Numerical experiment for sparse signal restoration

5.1. General setups

A key challenge in compressed sensing is recovering sparse signals from under-determined systems of linear equations [32]. The objective is to identify sparse solutions that meet a specific set of linear constraints. We consider the under-determined systems of linear equations as follows:

$$\theta = \Theta\tilde{x},$$

where θ is an observed signal, $\Theta \in R^{m\times n}$ is a linear mapping, and $\tilde{x}$ is an original sparse signal. It is inherently difficult to restore the sparse signal $\tilde{x}$ from the observed signal θ due to the ill-posed nature of the linear system, which often lacks a unique solution. To tackle this challenge, regularization techniques are often employed, which impose additional constraints to promote sparsity in the solution. Specifically, the problem can be formulated as the minimization of a combined $\ell_1 - \ell_2$ norm:

$$\min_x \frac{1}{2}\|\theta - \Theta x\|_2^2 + \varpi\|x\|_1, \tag{5.1}$$

where $\varpi > 0$ is the regularization parameter, and $\|x\|_1$ and $\|x\|_2$ represent the ℓ_1 and ℓ_2 norms, respectively. Clearly, the problem (5.1) is a convex, unconstrained optimization problem. This convexity is crucial as it ensures the existence of a global minimum, which can be efficiently identified using various optimization algorithms. The regularization parameter ϖ plays a key role in balancing the trade-off between sparsity and the signal.

The process of solving the problem (5.1) begins with reformulating the problem into a convex quadratic form [33]. This involves expressing any $x \in R^n$ as a vector composed of two non-negative components, such that

$$x = u - v, \quad u \geq 0, \quad v \geq 0, \quad u, v \in R^n,$$

where $u_i = (x_i)_+$ and $v_i = (-x_i)_+$ for all $i = 1, 2, \ldots, n$, with $(\cdot)_+ = \max\{0, x\}$. Together with the problem (5.1), this results in the following optimization problem:

$$\min_{u,v} \frac{1}{2}\|\Theta(u - v) - \theta\|_2^2 + \varpi e_n^{\mathrm{T}} u + \varpi e_n^{\mathrm{T}} v, \tag{5.2}$$

where $e_n = (1, 1, \ldots, 1)^{\mathrm{T}} \in R^n$. For convenient writing, we define the following notations:

$$\eta = \begin{pmatrix} u \\ v \end{pmatrix}, \quad \tau = \varpi e_{2n} + \begin{pmatrix} -\Theta^{\mathrm{T}}\theta \\ \Theta^{\mathrm{T}}\theta \end{pmatrix}, \quad \Gamma = \begin{pmatrix} \Theta^{\mathrm{T}}\Theta & -\Theta^{\mathrm{T}}\Theta \\ -\Theta^{\mathrm{T}}\Theta & \Theta^{\mathrm{T}}\Theta \end{pmatrix}.$$

According to these notations, the problem (5.2) can be reformulated as follows:

$$\min_\eta \frac{1}{2}\eta^{\mathrm{T}}\Gamma\eta + \tau^{\mathrm{T}}\eta, \quad \eta \geq 0. \tag{5.3}$$

Additionally, according to the work [34], the problem (5.3) was further reformulated, and it was demonstrated to be equivalent to the following:

$$h(\eta) = \min\{\eta, \Gamma\eta + \tau\} = 0.$$

As a result, our proposed algorithm can be applied in the sparse signal restoration.

To evaluate the performance of Algorithm MPHL, its numerical results are compared with those of Algorithm PSGM, Algorithm PDY, and Algorithm PCG. The context of this evaluation is sparse signal restoration, which involves recovering an original signal of length n from an observed signal of length m. Note that the length of the observed signal is less than that of the original signal. The performance of the algorithms is characterized using the following metrics: (i) Mean Squared Error (MSE); (ii) NIter; (iii) CPUT; (iv) Objective function value (objFun).

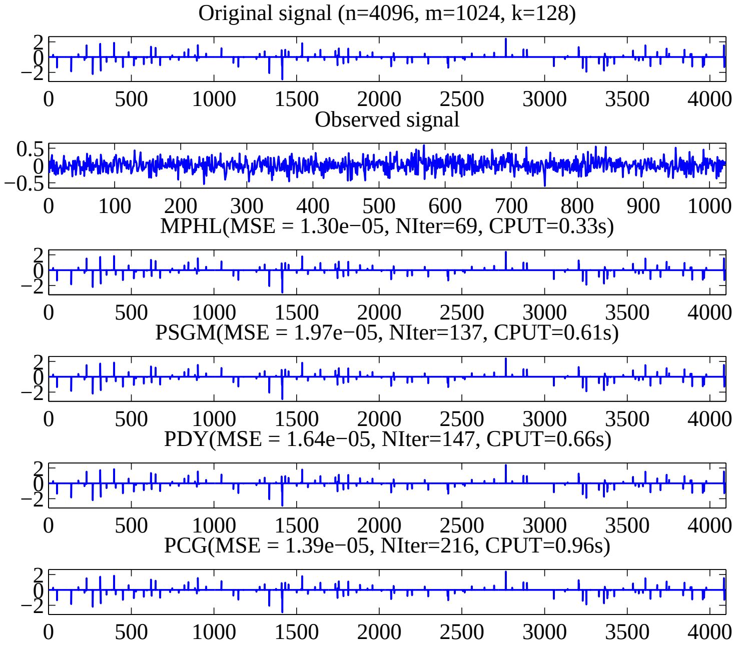

Figure 4. From top to bottom: the original signal, the observed signal, and the recovered signals by four algorithms with the signal size $(n, m, k) = (4096, 1024, 128)$.

5.2. *Numerical reports*

In this experiment, we consider two signal sizes: $(n, m, k) = (2^{12}, 2^{10}, 2^{7})$ and $(n, m, k) = (2^{13}, 2^{11}, 2^{8})$. Figures 4 and 6 sequentially display the original signal, the observed signal, and the recovered signals obtained by each algorithm. The results indicate that all algorithms successfully recover the original signal. However, Algorithm MPHL achieves signal recovery with fewer iterations and shorter CPU time compared to the other algorithms. Further analysis reveals that as the signal size increases, Algorithm MPHL shows only a modest increase in the number of iterations and CPU time. Additionally, Figures 5 and 7 provide a comparative analysis of the convergence results for the four algorithms in terms of MSE, NIter, CPUT, and objFun. These figures illustrate that the MSE values and objective function values for all four algorithms ultimately approach zero, indicating that each algorithm can achieve high-quality restoration results. Notably, Algorithm MPHL demonstrates faster convergence, outperforming the other algorithms in terms of both efficiency and computational cost.

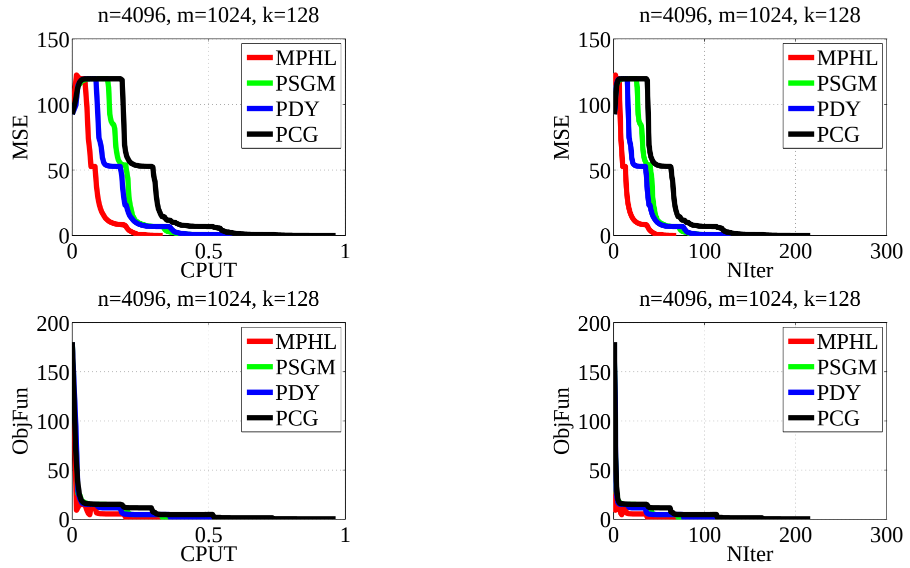

Figure 5. Comparison results of four algorithms with the signal size $(n, m, k) = (4096, 1024, 128)$.

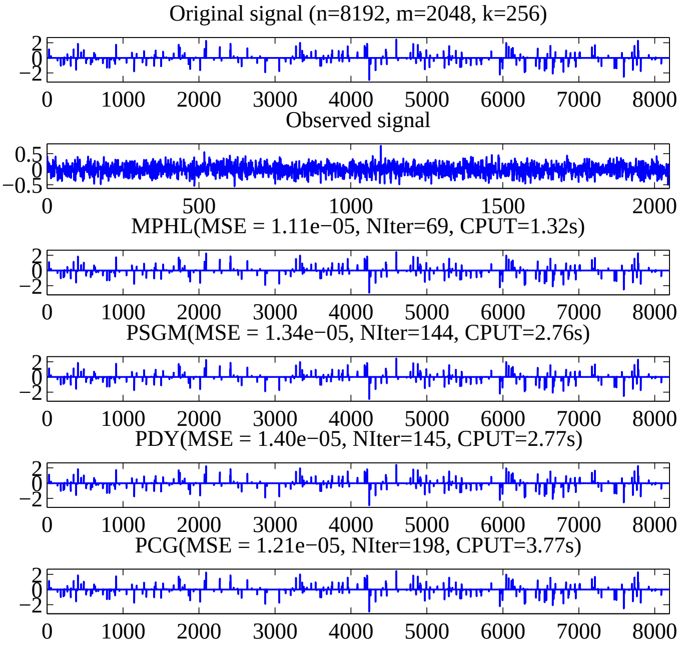

Figure 6. From top to bottom: the original signal, the observed signal, and the recovered signals by four algorithms with the signal size $(n, m, k) = (8192, 2048, 256)$.

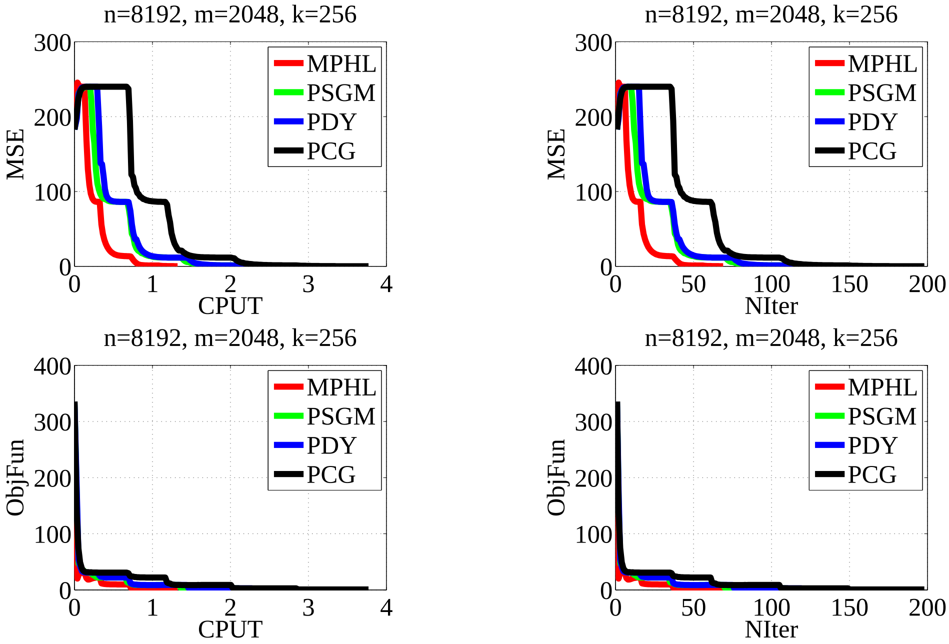

Figure 7. Comparison results of four algorithms with the signal size $(n, m, k) = (8192, 2048, 256)$.

6. Conclusions

In this paper, we proposed a novel hybrid PRP-HS-LS-type conjugate gradient algorithm tailored for solving nonlinear systems of equations with convex constraints. The proposed algorithm incorporates a hybrid technique to design the conjugate parameter, combining the strengths of PRP, HS, and LS conjugate gradient methods to enhance performance and stability. The search direction, formulated by using the hybrid conjugate parameter, inherently satisfies the sufficient descent condition and trust region properties. Remarkably, this is achieved without the need for a line search mechanism. The global convergence of the proposed algorithm is rigorously established under general conditions. Notably, this proof does not rely on the often restrictive Lipschitz continuity assumption, broadening the applicability of the algorithm to a wider class of problems. Extensive numerical experiments demonstrate the algorithm's superior efficiency, particularly in solving large-scale nonlinear systems of equations with convex constraints. Additionally, the proposed algorithm proves highly effective in addressing sparse signal restoration problems. In the future, we will discuss the potential extension of our algorithm to complex variables.

Author contributions

Xuejie Ma: Conceptualization, Investigation, Writing–original draft, Writing–review and editing, Funding acquisition; Songhua Wang: Conceptualization, Funding acquisition, Writing–review and editing. All authors have read and approved the final version of the manuscript for publication.

Use of AI tools declaration

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

Acknowledgments

This work is supported by the Natural Science Foundation in Guangxi Province, PR China (grant number 2024GXNSFAA010478) and the Guangzhou Huashang College Daoshi Project (grant number 2024HSDS15).

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. W. Xue, P. Wan, Q. Li, An online conjugate gradient algorithm for large-scale data analysis in machine learning, *AIMS Mathematics*, **6** (2021), 1515–1537. https://doi.org/10.3934/math.2021092
2. J. Chorowsk, J. M. Zurada, Learning understandable neural networks with nonnegative weight constraints, *IEEE Trans. Neural. Netw. Learn. Syst.*, **26** (2015), 62–69. https://doi.org/10.1109/TNNLS.2014.2310059
3. Y. H. Zheng, B. Jeon, D. H. Xu, Q. M. Wu, H. Zhang, Image segmentation by generalized hierarchical fuzzy C-means algorithm, *J. Intell. Fuzzy Syst.*, **28** (2015), 961–973. https://doi.org/10.3233/IFS-141378
4. Y. Li, C. Li, W. Yang, A new conjugate gradient method with a restart direction and its application in image restoration, *AIMS Mathematics*, **12** (2023), 28791–28807. https://doi.org/10.3934/math.20231475
5. S. Aji, P. Kumam, A. M. Awwal, An efficient DY-type spectral conjugate gradient method for system of nonlinear monotone equations with application in signal recovery, *Aims Mathematics*, **6** (2021), 8078–8106. https://doi.org/10.3934/math.2021469
6. K. Ahmed, M. Y. Waziri, A. S. Halilu, Sparse signal reconstruction via Hager-Zhang-type schemes for constrained system of nonlinear equations, *Optimization*, **73** (2024), 1949–1980. https://doi.org/10.1080/02331934.2023.2187255
7. D. D. Li, S. H. Wang, Y. Li, J. Q. Wu, A projection-based hybrid PRP-DY type conjugate gradient algorithm for constrained nonlinear equations with applications, *Appl. Numer. Math.*, **195** (2024), 105–125. https://doi.org/10.1016/j.apnum.2023.09.009
8. M. Dehghan, A. Shirilord, Three-step iterative methods for numerical solution of systems of nonlinear equations, *Eng. Comput.*, **38** (2022), 1015–1028. https://doi.org/10.1007/s00366-020-01072-1
9. M. Dehghan, G. Karamali, A. Shirilord, An iterative scheme for a class of generalized Sylvester matrix equations, *AUT J. Math. Comut.*, **5** (2024), 195–215.

10. K. Meinijes, A. P. Morgan, A methodology for solving chemical equilibrium systems, *Appl. Math. Comput.*, **22** (1987), 333–361.

11. X. Tong, S. Zhou, A smoothing projected Newton-type method for semismooth equations with bound constraints, *J. Ind. Manag. Optim.*, **1** (2005), 235–250. https://doi.org/10.3934/jimo.2005.1.235

12. M. Dehghan, M. Hajarian, Fourth-order variants of Newton's method without second derivatives for solving non-linear equations, *Eng. Comput.*, **29** (2012), 356–365. https://doi.org/10.1108/02644401211227590

13. C. Wang, Y. Wang, C. Xu, A projection method for a system of nonlinear monotone equations with convex constraints, *Math. Meth. Oper. Res.*, **66** (2007), 33–46. https://doi.org/10.1007/s00186-006-0140-y

14. G. Yuan, Z. Wei, M. Zhang, An active-set projected trust region algorithm for box constrained optimization problems, *J. Syst. Sci. Complex.*, **28** (2015), 1128–1147. https://doi.org/10.1007/s11424-014-2199-5

15. J. Fan, On the levenberg-marquardt methods for convex constrained nonlinear equations, *J. Ind. Manag. Optim.*, **9** (2013), 227–241. https://doi.org/10.3934/jimo.2013.9.227

16. J. Yin, J. Jian, G. Ma, A modified inexact Levenberg-Marquardt method with the descent property for solving nonlinear equations, *Comput. Optim. Appl.*, **87** (2024), 289–322. https://doi.org/10.1007/s10589-023-00513-z

17. Y. Ding, Y. Xiao, J. Li, A class of conjugate gradient methods for convex constrained monotone equations, *Optimization*, **66** (2017), 2309–2328. https://doi.org/10.1080/02331934.2017.1372438

18. M. Sun, J. Liu, New hybrid conjugate gradient projection method for the convex constrained equations, *Calcolo*, **53** (2016), 399–411. https://doi.org/10.1007/s10092-015-0154-z

19. H. Zheng, J. Li, P. Liu, An inertial Fletcher-Reeves-type conjugate gradient projection-based method and its spectral extension for constrained nonlinear equations, *J. Appl. Math. Comput.*, **70** (2024), 2427–2452. https://doi.org/10.1007/s12190-024-02062-y

20. Z. Yu, J. Lin, J. Sun, Y. Xiao, L. Liu, Z. Li, Spectral gradient projection method for monotone nonlinear equations with convex constraints, *Appl. Numer. Math.*, **59** (2009), 2416–2423. https://doi.org/10.1016/j.apnum.2009.04.004

21. M. Dehghan, R. Mohammadi-Arani, Generalized product-type methods based on bi-conjugate gradient (GPBiCG) for solving shifted linear systems, *Comput. Appl. Math.*, **36** (2017), 1591–1606. https://doi.org/10.1007/s40314-016-0315-y

22. J. K. Liu, Y. M. Feng, A derivative-free iterative method for nonlinear monotone equations with convex constraints, *Numer. Algor.*, **82** (2019), 245–262. https://doi.org/10.1007/s11075-018-0603-2

23. W. J. Hu, J. Z. Wu, G. L. Yuan, Some modified Hestenes-Stiefel conjugate gradient algorithms with application in image restoration, *Appl. Numer. Math.*, **158** (2020), 360–376. https://doi.org/10.1016/j.apnum.2020.08.009

24. G. Ma, J. Jin, J. Jian, D. Han, A modified inertial three-term conjugate gradient projection method for constrained nonlinear equations with applications in compressed sensing, *Numer. Algor.*, **92** (2023), 1621–1653. https://doi.org/10.1007/s11075-022-01356-1

25. J. H. Yin, J. B. Jian, X. Z. Jiang, M. X. Liu, L. Wang, A hybrid three-term conjugate gradient projection method for constrained nonlinear monotone equations with applications *Numer. Algor.*, **88** (2021), 389–418.

26. P. T. Gao, C. J. He, Y. Liu, An adaptive family of projection methods for constrained monotone nonlinear equations with applications, *Appl. Math. Comput.*, **359** (2019), 1–16. https://doi.org/10.1016/j.amc.2019.03.064

27. M. Li, A modified Hestense-Stiefel conjugate gradient method close to the memoryless BFGS quasi-Newton method, *Optim. Methods Softw.*, **33** (2018), 336–353. https://doi.org/10.1080/10556788.2017.1325885

28. M. Li, A three term Polak-Ribière-Polyak conjugate gradient method close to the memoryless BFGS quasi-Newton method, *J. Ind. Manag. Optim.*, **16** (2018), 245–260.

29. A. B. Abubakar, P. Kumam, H. Mohammad, A note on the spectral gradient projection method for nonlinear monotone equations with applications, *Comput. Appl. Math.*, **39** (2020), 129. https://doi.org/10.1007/s40314-020-01151-5

30. J. K. Liu, S. J. 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

31. E. D. Dolan, J. Jorge, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2001), 201–213. https://doi.org/10.1007/s101070100263

32. D. D. Li, J. Q. Wu, Y. Li, S. H. Wang, A modified spectral gradient projection-based algorithm for large-scale constrained nonlinear equations with applications in compressive sensing, *J. Comput. Appl. Math.*, **424** (2023), 115006. https://doi.org/10.1016/j.cam.2022.115006

33. M. A. T. Figueiredo, R. D. Nowak, S. J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE J. Sel. Top. Signal Process.*, **1** (2007), 586–597. https://doi.org/10.1109/JSTSP.2007.910281

34. Y. Xiao, Q. Wang, Q. Hu, Non-smooth equations based method for $\ell 1$-norm problems with applications to compressed sensing, *Nonlinear Anal. Theory Methods Appl.*, **74** (2011), 3570–3577.

AIMS Press