



Research article

An improved convex constrained conjugate gradient descent method for nonlinear monotone equations with signal recovery applications

Habibu Abdullahi^{1,2}, A. K. Awasthi³, Mohammed Yusuf Waziri^{2,4}, Issam A. R. Moghrabi^{5,*}, Abubakar Sani Halilu^{1,2}, Kabiru Ahmed^{2,4}, Sulaiman M. Ibrahim^{6,7}, Yau Balarabe Musa^{1,2} and Elissa M. Nadia⁸

¹ Department of Mathematics, Sule Lamido University Kafin Hausa, Nigeria

² Numerical Optimization Research Group, Bayero University Kano, Nigeria

³ Department of Mathematics, Lovely Professional University, Phagwara, India

⁴ Department of Mathematical Sciences, Bayero University, Kano, Nigeria

⁵ Information Systems and Technology Department, Kuwait Technical College, Kuwait

⁶ School of Quantitative Sciences, Universiti Utara Malaysia, Sintok, 06010, Kedah, Malaysia

⁷ Faculty of Education and Arts Sohar University, Sohar 311, Oman

⁸ Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Campus Besut, 22200 Terengganu, Malaysia

* **Correspondence:** Email: i.moghrabi@ktech.edu.kw.

Abstract: The conjugate descent (CD) method is a conjugate gradient (CG) variant with remarkable convergence properties. This paper presents a modified CD scheme for solving systems of constrained monotone nonlinear equations. Eigenvalue analysis demonstrates that the proposed search direction matrix is positive definite. By incorporating the proposed algorithm with Solodov and Svaiter's projection method (1999), several convex-constrained monotone nonlinear equations were solved, yielding impressive results. The new algorithm exhibits descent properties and achieves global convergence under appropriate assumptions. Numerical comparisons with recent algorithms in the literature highlight the efficiency and effectiveness of the proposed method. Furthermore, the method is applied to signal recovery experiments in compressive sensing.

Keywords: conjugate descent; eigenvalue analysis; convex constraints; compressive sensing; search direction matrix; positive definite matrix

Mathematics Subject Classification: 65K0519, 90C06, 90C30, 90C47, 90C90

1. Introduction

Consider the following system of equations:

$$\mathcal{H}(x) = 0, \quad x \in \psi, \quad (1.1)$$

where $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a continuous and monotone mapping and $\psi \subseteq \mathbb{R}^n$ is a non-empty closed and convex set. The monotonicity of $\mathcal{H}$ implies that it satisfies the following inequality:

$$(\mathcal{H}(x) - \mathcal{H}(y))^T (x - y) \geq 0 \quad \forall x, y \in \mathbb{R}^n. \quad (1.2)$$

In this paper, $\mathbb{R}^n$ and $\|\cdot\|$ refer to the real n -dimensional space and the Euclidean norm, respectively, while $\mathcal{H}(x_k) = \mathcal{H}_k$.

The system described above arises in various fields of science and engineering, as many scientific formulations naturally lead to systems of nonlinear equations. These systems appear in mathematical research areas such as differential equations, optimization theory, and integral equations. They also serve as subproblems in generalized proximal algorithms [1].

In chemistry, such systems frequently arise in the determination of chemical equilibrium points, while in economics, equilibrium analysis is often conducted in a similar manner [2]. Some variational inequalities are reformulated as monotone systems before being solved [3]. Additionally, these systems have applications in compressed sensing, particularly in signal reconstruction and image de-blurring problems [2, 4–6].

These systems also play a crucial role in solving optimal power flow equations [7]. In radiative transfer processes, they are applied through the discretization of the Chandrasekhar integral equation [2]. Furthermore, they are utilized in financial institutions for forecasting and automation [8].

Numerous methods and techniques have been developed, analyzed, and refined to solve system (1.1). Among these, the most widely used is Newton's method and its improved variants [9]. Newton's and quasi-Newton techniques are commonly employed for solving constrained monotone systems of nonlinear equations [9]. However, despite their popularity, the computational cost and storage requirements associated with the Jacobian matrix (the first derivative of the system) make Newton's method impractical for large-scale systems.

Quasi-Newton methods, in contrast, approximate the Jacobian matrix rather than computing it explicitly [10, 11]. While these methods eliminate the need for exact derivative computations, they require substantial memory for matrix storage and are slower than Newton's method due to limited Jacobian information [10, 11]. The steepest descent is among the popular descent methods for solving system (1.1). Although previously explored, it has largely fallen out of favor due to its slow convergence rate [12].

Given the challenges posed by large-scale systems, researchers have increasingly focused on conjugate gradient (CG) methods. As derivative-free approaches, CG methods are particularly well-suited for solving large-scale systems [13].

The CG methods are iterative schemes that generate a sequence of iterative points via

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \quad (1.3)$$

where α_k is the step length, and d_k is the search direction [14]. For CG methods, the search direction is defined as:

$$d_k = \begin{cases} -\mathcal{H}_k, & \text{if } k = 0, \\ -\mathcal{H}_k + \beta_k s_{k-1}, & \text{if } k \geq 1, \end{cases} \quad (1.4)$$

where $s_{k-1} = x_k - x_{k-1}$, and β_k is the CG parameter.

The CG parameter β_k takes different forms depending on its formulation. For example,

$$\beta_k^{CD} = \frac{\|\mathcal{H}_k\|^2}{-\mathcal{H}_{k-1}^T s_{k-1}}, \quad (1.5)$$

which is known as the conjugate descent (CD) CG parameter [15].

The parameter β_k is considered the most critical component of any CG direction, as different parameters lead to distinct CG directions with unique convergence properties and numerical performance. While some CG parameters are computationally efficient, others may fail to satisfy the inequality below, which is an essential condition for global convergence [2]:

$$\mathcal{H}_k^T d_k \leq -\tau \|\mathcal{H}_k\|^2, \quad \tau > 0. \quad (1.6)$$

The CD parameter β_k^{CD} belongs to the class of classical CG methods, exhibiting good convergence properties but weak numerical performance. To address this limitation, Yang et al. [16] hybridized the Liu-Storey (LS) and CD parameters using Wolfe's line search strategy, resulting in an efficient algorithm. Similarly, Kaelo et al. [17] hybridized the LS, Hestenes-Stiefel (HS), and Dai-Yuan (DY) CG parameters, producing an algorithm that satisfies (1.6). Using the strong Wolfe's line search strategy, the proposed algorithm achieved global convergence.

Snezana and Djordjevic [18] proposed a new hybrid of the CD and LS methods, ensuring that the hybrid parameter satisfies (1.6). Liu et al. [19] presented a scheme that also satisfies (1.6), introducing an LS-type CG parameter for solving the system (1.1) by applying Powell's strategy to the LS-type scheme [20]. Motivated by the method in [21], Wang et al. [22] developed a three-term CG algorithm for solving (1.1). Similarly, Li Wang [23], building on the FR-type CG iterative scheme, proposed an algorithm for symmetric systems by improving the method of Zhang et al. [24]. This method satisfies (1.6) and converges globally.

Liu and Li [25], inspired by the work of Yu et al. [26], presented a multivariate spectral algorithm for solving systems of the form (1.1). Acknowledging the weak numerical performance of the classical CD method, Zhang et al. [27] proposed a three-term CG direction satisfying (1.6) and the trust-region property. The proposed method achieved convergence through a modified Weak Wolfe-Powell (MWPP) line search strategy coupled with a projection algorithm, with numerical results outperforming many existing algorithms.

Recent studies have proposed various CG-based approaches to tackle constrained and unconstrained nonlinear equations. Ibrahim et al. [28] introduced two classes of spectral three-term derivative-free methods that effectively solve nonlinear equations. However, despite their efficiency, these methods struggle with ill-conditioned problems, leading to slow convergence. Similarly, Sabi'u and Sirisubtawee [29] proposed an inertial Dai-Liao conjugate gradient method tailored for convex constrained monotone equations, ensuring better stability and robustness. Nonetheless, inertial methods often require careful parameter tuning to prevent instability.

Additionally, Zheng et al. [30] developed a conjugate gradient projection method that efficiently handles equations with convex constraints. However, projection-based methods can suffer from increased computational cost, especially in high-dimensional problems [31]. Another notable contribution is by Chankong et al. [32], who formulated a class of derivative-free three-term descent Hestenes-Stiefel conjugate gradient algorithms for constrained nonlinear problems. While these algorithms demonstrate promising performance, they often encounter difficulties in non-smooth or highly nonlinear problems where descent conditions are hard to maintain. These recent advancements accentuate the increasing focus on improving CG-based methods by incorporating new techniques, such as inertial acceleration, spectral properties, and projection methods, to enhance convergence and stability.

Originally, classical CG methods were designed for solving systems of the form:

$$\min f(x), \quad x \in \mathbb{R}^n, \quad (1.7)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a nonlinear and continuously differentiable function, and its second derivative exists. Given their desirable properties and the fact that the first-order optimality condition for (1.7), i.e., $\nabla f(x) = 0$, is equivalent to (1.1) whenever $\nabla f = \mathcal{H}_k$, all methods applied to (1.7) can be extended to tackle problems of the form (1.1).

2. Motivation and derivation

To modify an existing conjugate gradient (CG) parameter or produce a new successful one, two factors must be considered:

- The new modified CG parameter must satisfy (1.6), which will ensure global convergence.
- It must outperform existing methods computationally by carrying out numerical comparisons with the relevant methods in the literature.

In [33], Yu and Guan presented a modified version of the DY CG parameter as follows:

$$\beta_k^{DDY} = \beta_k^{DY} - \frac{b\|\mathcal{H}_k\|^2}{(d_{k-1}^T y_{k-1})^2} d_{k-1}^T \mathcal{H}_k. \quad (2.1)$$

In their effort to prove the convergence of the method, they showed that (1.6) is satisfied for $\tau = 1 - \frac{1}{4b}$, where $b \geq \frac{1}{4}$.

Following the same argument, a modified CD parameter was proposed in [33] and is presented as:

$$\beta_k^{DCD} = \beta_k^{CD} - \frac{b\|\mathcal{H}_k\|^2}{(\mathcal{H}_k^T d_{k-1})^2} \mathcal{H}_k^T d_{k-1}, \quad (2.2)$$

and the sufficient descent inequality was proved for $b > \frac{1}{4}$.

Inspired by Yu and Guan's schemes, Yu et al. presented the spectral version of Eqs (2.1) and (2.2) in [34], with CG parameters given by:

$$\beta_k^{DS DY} = \beta_k^{S DY} - \frac{b\|\mathcal{H}_k\|^2}{\delta_k(d_{k-1}^T y_{k-1})^2} d_{k-1}^T \mathcal{H}_k, \quad (2.3)$$

and

$$\beta_k^{DS\text{CD}} = \beta_k^{SCD} - \frac{b\|\mathcal{H}_k\|^2}{\delta_k(\mathcal{H}_k^T d_{k-1}^T y_{k-1})^2} \mathcal{H}_k^T d_{k-1}, \quad (2.4)$$

where $b > 0$, but δ_k was left as a topic for further research.

The limitation of these four schemes (2.1)–(2.4) is that they do not satisfy the sufficient descent inequality (1.6) for the interval $(0, \frac{1}{4}]$, and consequently, they cannot attain global convergence for $b \in (0, \frac{1}{4}]$. Furthermore, the value of $b > 0$ was fixed throughout.

Motivated by the above and the fact that, to the best of our knowledge, modified CD schemes are limited in the literature, we aim to fill this gap by proposing a scheme that satisfies (1.6) and achieves global convergence for the interval $b_k \in (0, \infty)$, where the value of $b_k > 0$ is not fixed.

Consider the following:

$$\beta_k^{MCD} = \xi \frac{\|\mathcal{H}_k\|^2}{-\mathcal{H}_{k-1}^T s_{k-1}} - \xi \frac{b_k \|\mathcal{H}_k\|^2}{(\mathcal{H}_{k-1}^T s_{k-1})^2} \mathcal{H}_k^T s_{k-1}, \quad (2.5)$$

$\xi > 0$ and $b_k > 0$ is a parameter to be determined.

By (1.4) and (2.5), we have

$$d_k = -\mathcal{H}_k - \xi \frac{\|\mathcal{H}_k\|^2}{\mathcal{H}_{k-1}^T s_{k-1}} s_{k-1} - \xi \frac{b_k \|\mathcal{H}_k\|^2 \mathcal{H}_k^T s_{k-1}}{(\mathcal{H}_{k-1}^T s_{k-1})^2} s_{k-1}. \quad (2.6)$$

By Perry's idea in [35], (2.6) can be written as

$$d_k = -Q_k \mathcal{H}_k. \quad (2.7)$$

Here, Q_k is called the search direction matrix, which is expressed as

$$Q_k = I + \xi \frac{s_{k-1} \mathcal{H}_k^T}{\mathcal{H}_{k-1}^T s_{k-1}} + \xi \frac{b_k \|\mathcal{H}_k\|^2 s_{k-1} s_{k-1}^T}{(\mathcal{H}_{k-1}^T s_{k-1})^2}. \quad (2.8)$$

Q_k is non-symmetric, it shall be symmetrized as follows:

$$A_k = I + \xi \frac{s_{k-1} \mathcal{H}_k^T + \mathcal{H}_k s_{k-1}^T}{\mathcal{H}_{k-1}^T s_{k-1}} + \xi \frac{b_k \|\mathcal{H}_k\|^2 s_{k-1} s_{k-1}^T}{(\mathcal{H}_{k-1}^T s_{k-1})^2}. \quad (2.9)$$

Then (2.7) becomes

$$d_k = -A_k \mathcal{H}_k, \quad (2.10)$$

where A_k is the symmetric version of Q_k in (2.7).

Theorem 2.1. *The matrix A_k defined in (2.9) has eigenvalues λ_k^+ , λ_k^- , and 1 with multiplicity $(n-2)$, where*

$$\lambda_k^+ = \frac{1}{2} \left[(2 + 2\xi p_k + \xi b_k q_k) + \sqrt{(\xi b_k q_k + 2\xi p_k)^2 + 4\xi^2 (q_k - p_k^2)} \right] \quad (2.11)$$

and

$$\lambda_k^- = \frac{1}{2} \left[(2 + 2\xi p_k + \xi b_k q_k) - \sqrt{(\xi b_k q_k + 2\xi p_k)^2 + 4\xi^2 (q_k - p_k^2)} \right], \quad (2.12)$$

with

$$p_k = \frac{\mathcal{H}_k^T s_{k-1}}{\mathcal{H}_{k-1}^T s_{k-1}}, \quad q_k = \frac{\|\mathcal{H}_k\|^2 \|s_{k-1}\|^2}{(\mathcal{H}_{k-1}^T s_{k-1})^2}. \quad (2.13)$$

Moreover, the eigenvalues are real and positive.

Proof. The objective function $\mathcal{H}_k$ is not zero unless the solution is reached, as is the vector s_{k-1} . Suppose the solution is not reached; then, both $\mathcal{H}_k$ and s_{k-1} are nonzero vectors, and hence, $\mathcal{H}_k^T s_{k-1} > 0$. Let V be the vector space spanned by $\{\mathcal{H}_k, s_{k-1}\}$. The dimensions of V and its orthogonal complement $V^\perp$ are $\dim(V) \leq 2$ and $\dim(V^\perp) \geq n - 2$, respectively.

Let a set of mutually orthogonal vectors $\{\iota_{k-1}^i\}_i^{n-2} \subset V^\perp$ satisfy

$$\mathcal{H}_k^T \iota_{k-1}^i = s_{k-1}^T \iota_{k-1}^i \quad \text{for } i = 1, \dots, n-2. \quad (2.14)$$

Using (2.9) and (2.14), we obtain

$$A_k \iota_{k-1}^i = \iota_{k-1}^i. \quad (2.15)$$

The above equation is an eigenvector equation, and ι_{k-1}^i , for $i = 1, 2, \dots, n-2$, is an eigenvector with a corresponding eigenvalue of 1, repeated $n-2$ times. Let the remaining eigenvalues be λ_k^+ and λ_k^- . Furthermore, (2.9) can also be written as

$$A_k = I + \xi \frac{s_{k-1} \mathcal{H}_k^T}{\mathcal{H}_{k-1}^T s_{k-1}} + \xi \left(\frac{b_k \|\mathcal{H}_k\|^2 s_{k-1} + \mathcal{H}_k^T s_{k-1} \mathcal{H}_k}{(\mathcal{H}_{k-1}^T s_{k-1})^2} \right) s_{k-1}^T. \quad (2.16)$$

It is clear that (2.16) is a rank-2 update matrix therefore; from Theorem 1.2.16 of [36], the relation

$$\det(I + a_1 a_2^T + a_3 a_4^T) = (1 + a_1^T a_2)(1 + a_3^T a_4) - (a_1^T a_4)(a_2^T a_3),$$

with $a_1 = \xi \frac{s_{k-1}}{\mathcal{H}_{k-1}^T s_{k-1}}$, $a_2 = \mathcal{H}_k$, $a_3 = \xi \frac{b_k \|\mathcal{H}_k\|^2 s_{k-1} + \mathcal{H}_k^T s_{k-1} \mathcal{H}_k}{(\mathcal{H}_{k-1}^T s_{k-1})^2}$ and $a_4 = s_{k-1}$ can be used to get the determinant matrix of A_k as

$$\det(A_k) = \left(\xi \frac{(s_{k-1}^T \mathcal{H}_k)}{(\mathcal{H}_{k-1}^T s_{k-1})} + 1 \right)^2 + \xi \frac{b_k \|\mathcal{H}_k\|^2 \|s_{k-1}\|^2}{(\mathcal{H}_{k-1}^T s_{k-1})^2} - \xi^2 \frac{\|\mathcal{H}_k\|^2 \|s_{k-1}\|^2}{(\mathcal{H}_{k-1}^T s_{k-1})^2}. \quad (2.17)$$

Furthermore, it is known that the trace of (2.9) is equivalent to the total number of its eigenvalues; hence we have

$$\text{trace}(A_k) = \underbrace{1 + 1 + \dots + 1}_{(n-2)\text{times}} + \lambda_k^+ + \lambda_k^- = n + 2\xi \frac{\mathcal{H}_k^T s_{k-1}}{\mathcal{H}_{k-1}^T s_{k-1}} + \xi \frac{b_k \|\mathcal{H}_k\|^2 \|s_{k-1}\|^2}{(\mathcal{H}_{k-1}^T s_{k-1})^2}, \quad (2.18)$$

and consequently,

$$\lambda_k^+ + \lambda_k^- = 2 + 2\xi \frac{\mathcal{H}_k^T s_{k-1}}{\mathcal{H}_{k-1}^T s_{k-1}} + \xi \frac{b_k \|\mathcal{H}_k\|^2 \|s_{k-1}\|^2}{(\mathcal{H}_{k-1}^T s_{k-1})^2}. \quad (2.19)$$

Then by (2.13), (2.17), and (2.19), we obtain

$$\lambda_k^2 - (2 + 2\xi p_k + \xi b_k q_k) \lambda_k + ((\xi p_k + 1)^2 + \xi(b_k - \xi)q_k) = 0. \quad (2.20)$$

Then, using the quadratic formula, we obtain

$$\lambda_k^\pm = \frac{1}{2} \left[(2 + 2\xi p_k + \xi b_k q_k) \pm \sqrt{(\xi b_k q_k + 2\xi p_k)^2 + 4\xi^2(q_k - p_k^2)} \right]. \quad (2.21)$$

This gives (2.11) and (2.12). Next, we show that λ_k^+ and λ_k^- are real and positive. The discriminant is positive since $q_k > p_k^2$. Therefore, we need to have

$$2 + 2\xi \frac{\mathcal{H}_k^T s_{k-1}}{\mathcal{H}_{k-1}^T s_{k-1}} + \xi \frac{b_k \|\mathcal{H}_k\|^2 \|s_{k-1}\|^2}{(\mathcal{H}_{k-1}^T s_{k-1})^2} > 0. \quad (2.22)$$

λ_k^+ is real and positive when

$$b_k > -2 \frac{(\mathcal{H}_k^T s_{k-1})(\mathcal{H}_{k-1}^T s_{k-1})}{\|\mathcal{H}_k\|^2 \|s_{k-1}\|^2} - 2 \frac{(\mathcal{H}_{k-1}^T s_{k-1})^2}{\xi \|\mathcal{H}_k\|^2 \|s_{k-1}\|^2}. \quad (2.23)$$

For λ_k^- to be real and positive, we require

$$\lambda_k^- = \frac{1}{2} \left[(2 + 2\xi p_k + \xi b_k q_k) - \sqrt{(\xi b_k q_k + 2\xi p_k)^2 + 4\xi^2 (q_k - p_k^2)} \right] > 0. \quad (2.24)$$

After algebraic simplification, we obtain

$$b_k > \xi - \left(\frac{\sqrt{\xi} (\mathcal{H}_k^T s_{k-1})}{\|\mathcal{H}_k\| \|s_{k-1}\|} + \frac{\mathcal{H}_{k-1}^T s_{k-1}}{\sqrt{\xi} \|\mathcal{H}_k\| \|s_{k-1}\|} \right)^2. \quad (2.25)$$

Now, (2.25) can be written as

$$b_k = \xi - \varphi \left(\frac{\sqrt{\xi} (\mathcal{H}_k^T s_{k-1})}{\|\mathcal{H}_k\| \|s_{k-1}\|} + \frac{\mathcal{H}_{k-1}^T s_{k-1}}{\sqrt{\xi} \|\mathcal{H}_k\| \|s_{k-1}\|} \right)^2. \quad (2.26)$$

with $\xi > 0$ and $\varphi \leq 0$.

□

Here, we present the proposed direction.

$$d_k = \begin{cases} -\mathcal{H}_k, & \text{if } k = 0, \\ -\mathcal{H}_k - \frac{\xi \|\mathcal{H}_k\|^2}{\mathcal{H}_{k-1}^T s_{k-1}} s_{k-1} - \frac{\xi b_k \|\mathcal{H}_k\|^2 (\mathcal{H}_{k-1}^T s_{k-1})}{(\mathcal{H}_{k-1}^T s_{k-1})^2} s_{k-1}, & \text{if } \mathcal{H}_k^T s_{k-1} > 0 \text{ and } \mathcal{H}_{k-1}^T s_{k-1} \geq r \|\mathcal{H}_k\| \|s_{k-1}\|, r > 1, \\ -\mathcal{H}_k + \frac{\|\mathcal{H}_k\|^2}{\max\{-\mathcal{H}_{k-1}^T s_{k-1}, \gamma \|\mathcal{H}_{k-1}\| \|s_{k-1}\|\}} s_{k-1}, & \gamma \in (0, 1), \text{ otherwise.} \end{cases} \quad (2.27)$$

To improve its convergence attributes, b_k in (2.26) can be modified to

$$b_k = \xi - \varphi \left(\frac{\sqrt{\xi} (\mathcal{H}_k^T s_{k-1})}{U_k} + \frac{\mathcal{H}_{k-1}^T s_{k-1}}{V_k} \right)^2, \quad \xi > 0, \quad \varphi \leq 0, \quad (2.28)$$

$$U_k = \max\{\|\mathcal{H}_k\| \|s_{k-1}\|, \xi \|\mathcal{H}_{k-1}\| \|s_{k-1}\|\},$$

$$V_k = \max\{\|\mathcal{H}_{k-1}\| \|s_{k-1}\|, \xi \|\mathcal{H}_k\| \|s_{k-1}\|\}.$$

Let the projection mapping be defined as follows before presenting the proposed algorithm.

Definition 2.1. Let $\psi \subseteq \mathbb{R}^n$ be a not empty, convex, and closed set. For $x \in \mathbb{R}^n$, the projection of x onto ψ is given by

$$P_\psi = \arg \min\{\|x - y\| : y \in \psi\}. \quad (2.29)$$

Equation (2.29) is called a projection mapping. The projection, P_ψ , is nonexpensive, namely,

$$\|P_\psi(x) - P_\psi(y)\| \leq \|x - y\|, \quad x, y \in \psi. \quad (2.30)$$

and consequently,

$$\|P_\psi(x) - y\| \leq \|x - y\|, \quad \forall y \in \psi. \quad (2.31)$$

Input: Input: Given $x_0 \in \psi$, $d_0 = -\mathcal{H}_0$, $\rho \in (0, 1)$, $\sigma > 0$, $\epsilon = 10^{-6}$, and $\zeta > 0$, set $k = 0$.

Step i: find $\mathcal{H}_k$. If $\|\mathcal{H}_k\| \leq \epsilon$ stop, otherwise go to Step ii.

Step ii: Let $\alpha_k = \zeta \rho^{m_k}$, where m_k is the least positive integer m such that

$$-\mathcal{H}(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \|\mathcal{H}(x_k + \alpha_k d_k)\| \|d_k\|^2. \quad (2.32)$$

Step iii: Set

$$z_k = x_k + \alpha_k d_k. \quad (2.33)$$

Step iv: If $z_k \in \psi$ and $\|\mathcal{H}(z_k)\| \leq \epsilon$, stop else, go to Step v.

Step v: Solve for x_{k+1} by

$$x_{k+1} = P_\psi[x_k - \lambda_k \mathcal{H}(z_k)], \quad (2.34)$$

where, $\lambda_k = \frac{(x_k - z_k)^T \mathcal{H}(z_k)}{\|\mathcal{H}(z_k)\|^2}$.

Step vi: Get the new d_{k+1} using (2.27) and (2.28).

Step vii: Update $k = k + 1$ and repeat Step ii.

An improved CD Iterative Scheme

3. Convergence analysis

This part presents the analysis of the proposed algorithm UMCD and some certain preassumptions.

Assumption 3.1. The map $\mathcal{H}$ is Lipschitz continuous, i.e., there is a constant $L > 0$ such that

$$\|\mathcal{H}(x) - \mathcal{H}(y)\| \leq L\|x - y\| \quad \text{holds for any } x, y \in \mathbb{R}^n. \quad (3.1)$$

Assumption 3.2. $\mathcal{H}$ is strongly monotone, i.e., $\forall x, y \in \mathbb{R}^n$, then, there exists $c > 0$ such that

$$(x - y)^T (\mathcal{H}(x) - \mathcal{H}(y)) \geq c\|x - y\|^2, \quad \text{for all } x, y \in \mathbb{R}^n. \quad (3.2)$$

Assumption 3.3. Let a nonempty set $\bar{S}$, be the solution set of the system (1.1)

Lemma 3.1. Let $\xi > 0$, $b_k > 0$ and $\varphi \leq 0$, then $\forall k \geq 0$ the sequence of search directions $\{d_{k-1}\}$ presented by (2.27) with (2.28) satisfies (1.6) where $\tau = 1$.

Proof. From (2.27), it is clear that for $k = 0$, $\mathcal{H}_0^T d_0 = -\|\mathcal{H}_0\|^2$. This satisfies (1.6) with $\tau = 1$,

Case I. For $k > 1$, If $\mathcal{H}_k^T s_{k-1} > 0$ and $\mathcal{H}_{k-1}^T s_{k-1} \geq r\|\mathcal{H}_k\|\|s_{k-1}\|$,

$$d_k^T \mathcal{H}_k = -\|\mathcal{H}_k\|^2 - \frac{\xi\|\mathcal{H}_k\|^2(\mathcal{H}_k^T s_{k-1})}{\mathcal{H}_{k-1}^T s_{k-1}} - \frac{\xi b_k\|\mathcal{H}_k\|^2(\mathcal{H}_k^T s_{k-1})^2}{(\mathcal{H}_{k-1}^T s_{k-1})^2} \leq -\|\mathcal{H}_k\|^2.$$

Case II. If $\mathcal{H}_k^T s_{k-1} = 0$, then $d_k^T \mathcal{H}_k = -\|\mathcal{H}_k\|^2 + \frac{\|\mathcal{H}_k\|^2(\mathcal{H}_k^T s_{k-1})}{\max\{-\mathcal{H}_{k-1}^T s_{k-1}, \gamma\|\mathcal{H}_k\|\|s_{k-1}\|\}} \leq -\|\mathcal{H}_k\|^2 + \frac{\|\mathcal{H}_k\|^2(\mathcal{H}_k^T s_{k-1})}{\gamma\|\mathcal{H}_k\|\|s_{k-1}\|} \leq -\|\mathcal{H}_k\|^2.$

Case III. If $\mathcal{H}_k^T s_{k-1} < 0$, then $d_k^T \mathcal{H}_k = -\|\mathcal{H}_k\|^2 + \frac{\|\mathcal{H}_k\|^2(\mathcal{H}_k^T s_{k-1})}{\max\{-\mathcal{H}_{k-1}^T s_{k-1}, \gamma\|\mathcal{H}_k\|\|s_{k-1}\|\}} \leq -\|\mathcal{H}_k\|^2 + \frac{\|\mathcal{H}_k\|^2(\mathcal{H}_k^T s_{k-1})}{\gamma\|\mathcal{H}_k\|\|s_{k-1}\|} \leq -\|\mathcal{H}_k\|^2$. where $\left(\frac{\|\mathcal{H}_k\|^2(\mathcal{H}_k^T s_{k-1})}{\gamma\|\mathcal{H}_k\|\|s_{k-1}\|} < 0\right)$, and hence the required result. $\square$

Lemma 3.2. Let the assumptions hold and $\{x_k\}, \{z_k\}$ be sequences defined by the UMCD algorithm, then $\{x_k\}, \{z_k\}$ are bounded and

$$\|d_k\| \leq M. \quad (3.3)$$

$$\lim_{k \rightarrow \infty} \|x_k - z_k\| = 0. \quad (3.4)$$

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0. \quad (3.5)$$

Proof. Let $\bar{x} \in \bar{S}$ be the solution set of the system given by (1.1); then by the monotonicity of $\mathcal{H}$, we obtain

$$(x_k - \bar{x})^T \mathcal{H}(z_k) \geq (x_k - z_k)^T \mathcal{H}(z_k). \quad (3.6)$$

Then, using Eq (2.32) (the line search condition) and the definition of z_k , it follows

$$(x_k - z_k)^T \mathcal{H}(z_k) \geq \sigma \alpha_k^2 \|d_k\|^2 > 0. \quad (3.7)$$

Also, from (2.34), we have

$$\begin{aligned} \|x_{k+1} - \bar{x}\|^2 &= \|x_k - \lambda_k \mathcal{H}(z_k) - \bar{x}\|^2 \\ &= \|x_k - \bar{x}\|^2 - 2\lambda_k (x_k - \bar{x})^T \mathcal{H}(z_k) + \lambda_k^2 \|\mathcal{H}(z_k)\|^2 \\ &\leq \|x_k - \bar{x}\|^2 - 2\lambda_k (x_k - z_k)^T \mathcal{H}(z_k) + \lambda_k^2 \|\mathcal{H}(z_k)\|^2 \\ &= \|x_k - \bar{x}\|^2 - \frac{((x_k - z_k)^T \mathcal{H}(z_k))^2}{\|\mathcal{H}^T(z_k)\|^2} \\ &\leq \|x_k - \bar{x}\|^2. \end{aligned} \quad (3.8)$$

Hence,

$$\|x_{k+1} - \bar{x}\| \leq \|x_k - \bar{x}\|. \quad (3.9)$$

Repeatedly, $\|x_k - \bar{x}\| \leq \|x_0 - \bar{x}\| \forall k$.

This means that $\{x_k - \bar{x}\}$ is decreasing and $\{x_k\}$ is bounded. Furthermore, by (3.1)–(3.3) and (3.9), we get

$$\|\mathcal{H}_k\| = \|\mathcal{H}_k - \mathcal{H}_k(\bar{x})\| \leq L\|x_k - \bar{x}\| \leq L\|x_0 - \bar{x}\|. \quad (3.10)$$

Hence

$$\|\mathcal{H}_k\| \leq N_1 \quad \text{for} \quad N_1 = L\|x_0 - \bar{x}\|. \quad (3.11)$$

Then from (2.28)

$$\begin{aligned}
 |b_k| &= \left| \xi - \varphi \left(\frac{\sqrt{\xi}(\mathcal{H}_k^T s_{k-1})}{U_k} + \frac{(\mathcal{H}_{k-1}^T s_{k-1})}{V_k} \right)^2 \right| \\
 &\leq |\xi| + |\varphi| \left| \left(\frac{\sqrt{\xi} \|\mathcal{H}_k\| \|s_{k-1}\|}{U_k} + \frac{\|\mathcal{H}_{k-1}\| \|s_{k-1}\|}{V_k} \right)^2 \right| \\
 &\leq \xi + |\varphi| \left| \left(\frac{\sqrt{\xi} \|\mathcal{H}_k\| \|s_{k-1}\|}{\|\mathcal{H}_k\| \|s_{k-1}\|} + \frac{\|\mathcal{H}_{k-1}\| \|s_{k-1}\|}{\|\mathcal{H}_{k-1}\| \|s_{k-1}\|} \right)^2 \right| \\
 &= \xi + |\varphi| (\sqrt{\xi} + 1)^2.
 \end{aligned} \tag{3.12}$$

This implies

$$|b_k| \leq \lambda \quad \text{for} \quad \lambda = \xi + |\varphi| (\sqrt{\xi} + 1)^2. \tag{3.13}$$

From (2.27) and using (2.28), (3.11), and (3.13), we have

If, $\mathcal{H}_k^T s_{k-1} > 0$ and $\mathcal{H}_{k-1}^T s_{k-1} \geq r \|\mathcal{H}_k\| \|s_{k-1}\|$,

$$\begin{aligned}
 \|d_k\| &= \left\| -\mathcal{H}_k - \left(\frac{\xi \|\mathcal{H}_k\|^2}{\mathcal{H}_{k-1}^T s_{k-1}} - \frac{b_k \|\mathcal{H}_k\|^2 (\mathcal{H}_k^T s_{k-1})}{(\mathcal{H}_{k-1}^T s_{k-1})^2} \right) s_{k-1} \right\| \\
 &\leq \|\mathcal{H}_k\| + \left\| \frac{\xi \|\mathcal{H}_k\|^2}{\mathcal{H}_{k-1}^T s_{k-1}} + \frac{b_k \|\mathcal{H}_k\|^3 \|s_{k-1}\|}{(\mathcal{H}_{k-1}^T s_{k-1})^2} \right\| \|s_{k-1}\| \\
 &\leq \|\mathcal{H}_k\| + \frac{\xi \|\mathcal{H}_k\|^2 \|s_{k-1}\|}{r \|\mathcal{H}_k\| \|s_{k-1}\|} + \frac{\|b_k\| \|\mathcal{H}_k\|^3 \|s_{k-1}\|^2}{r^2 \|\mathcal{H}_k\|^2 \|s_{k-1}\|^2} \\
 &\leq \|\mathcal{H}_k\| + \frac{\xi \|\mathcal{H}_k\|}{r} + \frac{\lambda \|\mathcal{H}_k\|}{r^2} \\
 &\leq N_1 + \frac{\xi N_1}{r} + \frac{\lambda N_1}{r^2}.
 \end{aligned} \tag{3.14}$$

Let $M_1 = N_1 + \frac{\xi N_1}{r} + \frac{\lambda N_1}{r^2}$.

Otherwise,

$$\begin{aligned}
 \|d_k\| &= \left\| -\mathcal{H}_k + \frac{\|\mathcal{H}_k\|^2}{\max\{-\mathcal{H}_{k-1}^T s_{k-1}, \gamma \|\mathcal{H}_k\| \|s_{k-1}\|\}} s_{k-1} \right\| \\
 &\leq \|\mathcal{H}_k\| + \frac{\|\mathcal{H}_k\|^2 \|s_{k-1}\|}{\gamma \|\mathcal{H}_k\| \|s_{k-1}\|} \\
 &\leq N_1 + \frac{\lambda N_1}{\gamma}.
 \end{aligned} \tag{3.15}$$

Let $M_2 = N_1 + \frac{N_1}{\gamma}$ and take $M = \max\{M_1, M_2\}$, then the Lemma (3.2) is proved.

The sequences $\{d_k\}$ and $\{x_k\}$ are bounded; Eq (2.33) shows that $\{z_k\}$ is bounded. Using the same argument as in (3.10), it implies that

$$\|\mathcal{H}(z_k)\| \leq \ell, \quad \ell \text{ is a positive constant.} \tag{3.16}$$

Equation (3.8) shows that,

$$\left((x_k - z_k)^T \mathcal{H}(z_k)\right)^2 \leq \|\mathcal{H}_k\|^2 (\|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2). \quad (3.17)$$

From (2.32), we obtain

$$\sigma^2 \alpha_k^4 \|d_k\|^4 \leq \alpha_k^2 (\mathcal{H}(z_k)^T d_k)^2. \quad (3.18)$$

By (3.8) and (3.18), we have

$$\sigma^2 \alpha_k^4 \|d_k\|^4 \leq \|\mathcal{H}(z_k)\|^4 (\|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2), \quad (3.19)$$

$\{x_k - \bar{x}\}$ converged and $\{\mathcal{H}(z_k)\}$ is bounded; we can apply the limit on (3.17) to obtain

$$\sigma^2 \lim_{k \rightarrow \infty} \alpha_k^4 \|d_k\|^4 \leq 0, \quad (3.20)$$

and hence

$$\lim_{k \rightarrow \infty} \alpha_k d_k = 0, \quad (3.21)$$

(3.19) and (2.33) indicates that (3.4) is true. However, it is deduced from the definition of λ_k that

$$\|x_{k+1} - x_k\| = \|x_k - \lambda_k \mathcal{H}(z_k) - x_k\| = \|\lambda_k \mathcal{H}(z_k)\| \leq \|x_k - z_k\|. \quad (3.22)$$

This means Eq (3.5) is proved. $\square$

Theorem 3.1. *Let the assumptions (3.1)–(3.3), be true and the sequence $\{x_k\}$ be defined by the UMCD algorithm; then*

$$\liminf_{k \rightarrow \infty} \|\mathcal{H}_k\| = 0. \quad (3.23)$$

Proof. By contradicting the above assumptions, i.e., suppose (3.23) is not true, then

$$\|\mathcal{H}_k\| \geq \epsilon_0 \quad \text{is true} \quad \forall k > 0. \quad (3.24)$$

Let $\|d_k\| \neq 0$ unless when $\|\mathcal{H}_k\| = 0$, then there exists $\delta_1 > 0$ such that

$$\|d_k\| \geq \delta_1. \quad (3.25)$$

If $\alpha_k \neq \xi$, then $\alpha_k, \rho^{-1} \alpha_k$ does not agrees (2.32), i.e.,

$$-\mathcal{H}(x_k + \rho^{-1} \alpha_k d_k)^T d_k < \sigma \rho^{-1} \alpha_k \|\mathcal{H}(x_k + \rho^{-1} \alpha_k d_k)\| \|d_k\|^2. \quad (3.26)$$

Using (3.26) and (1.6), it yields

$$\begin{aligned} \tau \|\mathcal{H}_k\|^2 &\leq -\mathcal{H}_k^T d_k \\ &= (\mathcal{H}(x_k + \rho^{-1} \alpha_k d_k) - \mathcal{H}_k)^T d_k - \mathcal{H}_k(x_k + \rho^{-1} \alpha_k d_k)^T d_k, \\ &\leq L \rho^{-1} \alpha_k \|d_k\|^2 + \sigma \rho^{-1} \alpha_k \|d_k\|^2 \\ &= \alpha_k \|d_k\| (L + \sigma) \rho^{-1} \|d_k\|. \end{aligned} \quad (3.27)$$

And from (3.27) we obtain

$$\alpha_k \|d_k\| > \frac{\rho}{L + \sigma} \frac{\tau \|\mathcal{H}_k\|^2}{\|d_k\|} \geq \frac{\rho}{L + \sigma} \frac{\tau \epsilon_0^2}{M}. \quad (3.28)$$

Equation (3.28) contradicts Eq (3.21). Hence, (3.23) is true. This completes the proof. $\square$

4. Numerical experiments

This section is divided into two parts. In this part, the numerical results of the proposed algorithm are presented. To measure the strength of the proposed algorithm (UMCD), the following algorithms were used:

- MDDYM (2021) algorithm proposed in [37].
- DTCG1 (2020) algorithm proposed in [6].
- ZANG (2023) algorithm proposed in [38].
- MCHCG (2024) algorithm proposed in [39].

In the second part of this study, we extend the UMCD algorithm to explore its effectiveness in compressive sensing experiments. To rigorously assess the performance of our proposed algorithm, we conduct a signal recovery experiment in conjunction with the CHCG algorithm [2]. The implementation is executed using MATLAB version 8.3.0 (R2014a) on a laptop equipped with a 1.80 GHz processor (CPU) and 8 GB of RAM.

For the UMCD algorithm, we selected the following key parameters: $\xi = 1$, $\sigma = \varphi = 10^{-4}$, and $\rho = \zeta = 0.9$. In comparison, the parameters for the other algorithms are provided in their respective documents. This structured approach allows us to draw meaningful comparisons and insights regarding the performance of the proposed algorithm.

The iterations of all five algorithms were configured to stop when either $|\mathcal{H}_k| \leq 10^{-6}$, $|\mathcal{H}(z_k)| \leq 10^{-6}$ are reached, or the number of iterations exceeds 2000 without meeting the stopping criterion. Moreover, the symbol '–' indicates a failure during the iterative process.

The numerical experiments for the five algorithms were carried out using eight (8) different initial points and twelve (12) test problems (T1–T12).

$x_1 = (0.01, 0.01, \dots, 0.01)^T$, $x_2 = (0.25, 0.25, \dots, 0.25)^T$, $x_3 = (0.4, 0.4, \dots, 0.4)^T$, $x_4 = (0.5, 0.5, \dots, 0.5)^T$, $x_5 = (1.25, 1.25, \dots, 1.25)^T$, $x_6 = (0.3, 0.3, \dots, 0.3)^T$, $x_7 = (1, 1, \dots, 1)^T$, and $x_8 = (0.1, 0.1, \dots, 0.1)^T$.

T1 [6]

$$\mathcal{H}_1(x) = e^{x_1} - 1,$$

$$\mathcal{H}_i(x) = e^{x_i} + x_{i-1} - 1, \quad i = 2, 3, 4, \dots, n-1,$$

where $\psi = \mathbb{R}_+^n$.

T2 [5]

$$\mathcal{H}_i(x) = \log(x_i + 1) - \frac{x_i}{n}, \quad i = 2, 3, \dots, n-1,$$

where $\psi = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq n, x_i > -1, \quad i = 1, 2, 3, \dots, n\}$.

T3 [37]

$$\mathcal{H}_i(x) = 2x_i - \sin |x_i|, \quad i = 1, 2, 3, \dots, n,$$

where $\psi = \mathbb{R}_+^n$.

T4 [5]

$$\mathcal{H}_i(x) = \cos x_i + x_i - 1, \quad i = 1, 2, 3, \dots, n,$$

where $\psi = \mathbb{R}_+^n$.

T5 [6]

$$\mathcal{H}_i(x) = e^{x_i} - 1, \quad i = 1, 2, 3, \dots, n,$$

where $\psi = \mathbb{R}_+^n$.

T6 [37]

$$\mathcal{H}_1(x) = -2x_1 - x_2 + e^{x_1} - 1,$$

$$\mathcal{H}_i(x) = -x_{i-1} + 2x_i - x_{i+1} + e^{x_i} - 1,$$

$$\mathcal{H}_n(x) = -x_{n-1} + 2x_n + e^{x_n} - 1, \quad i = 2, 3, 4, \dots, n-1,$$

where $\psi = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq n, x_i \geq 0, \quad i = 1, 2, 3, \dots, n\}$.

T7 [6]

$$\mathcal{H}_1(x) = x_1 - e^{\cos(b(x_1+x_2))},$$

$$\mathcal{H}_i(x) = x_i - e^{\cos(b(x_{i-1}+x_i+x_{i+1}))}, \quad i = 2, 3, 4, \dots, n-1,$$

$$\mathcal{H}_n(x) = x_n - e^{\cos(b(x_{n-1}+x_n))},$$

$$b = \frac{1}{n+1}, \text{ where } \psi = \mathbb{R}_+^n.$$

T8 [37]

$$\mathcal{H}_i(x) = x_i - \sin|x_i - 1|, \quad i = 1, 2, \dots, n,$$

where $\psi = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq n, x_i > -1, \quad i = 1, 2, 3, \dots, n\}$.

T9 [40]

$$\mathcal{H}_i(x) = e^{x_i^2} + \frac{3}{2} \sin(2x_i) - 1, \quad i = 1, 2, 3, \dots, n,$$

where $\psi = \mathbb{R}_+^n$.

T10

$$\mathcal{H}_1(x) = \cos x_1 - 9 + 3x_1 + 8e^{x_2},$$

$$\mathcal{H}_i(x) = \cos x_i - 9 + 3x_i + 8e^{x_{i-1}}, \quad i = 2, 3, 4, \dots, n,$$

where $\psi = \mathbb{R}_+^n$.

T11 Modification of T1 in [41]

$$\mathcal{H}_1(x) = e^{\sin x_1} - 1,$$

$$\mathcal{H}_i(x) = e^{\sin x_i} + x_{i-1} - 1, \quad i = 2, 3, 4, \dots, n-1,$$

where $\psi = \mathbb{R}_+^n$.

T12 Modification of T3 in [41]

$$\mathcal{H}_i(x) = 3x_i - \sin x_i, \quad i = 1, 2, 3, \dots, n,$$

where $\psi = \mathbb{R}_+^n$.

Tables 1–5 display the performance of the proposed algorithm (UMCD) in comparison with recent CG algorithms: MDDYM [37] and DTG1 [6]. In the tables, the terms *IP*, *Iter*, and *Fval* refer to the initial point, number of iterations, and function evaluations, respectively. Additionally, *Time(s)* and $\|\mathcal{H}_k\|$ represent CPU time and the norm of residual functions, respectively.

To ensure fairness, the five algorithms were run with the same initial starting points and dimensions. As observed from the tables, the UMCD algorithm solves all the test functions with the fewest number of iterations, CPU time, and function evaluations compared to the DTG1 and MDDYM algorithms. Based on these results, the UMCD algorithm outperformed both MDDYM and DTG1.

Table 1. Numerical of UMCD, MDDYM & DTCG1 methods for problems 1 and 2.

Problem	Dim	IP	UMCD				MDDYM				DTCG1			
			Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $
P1	100	x1	15	16	0.03303	0.00E+00	15	47	0.05952	8.76E-07	1001	1050	0.29023	6.55E+04
		x2	4	5	0.00783	0.00E+00	12	37	0.01407	5.31E-07	1001	1116	0.32494	6.55E+04
		x3	4	5	0.01144	0.00E+00	14	44	0.01315	2.00E-07	37	181	0.05921	6.96E-07
		x4	2	3	0.00478	0.00E+00	14	46	0.02308	7.27E-07	18	133	0.04298	0.00E+00
		x5	4	5	0.01512	0.00E+00	16	51	0.01617	7.30E-07	29	321	0.04569	0.00E+00
		x6	4	5	0.01100	0.00E+00	13	43	0.01019	4.74E-10	29	318	0.06099	0.00E+00
		x7	4	5	0.01278	0.00E+00	11	33	0.02672	3.35E-08	1001	1028	0.29032	6.55E+04
		x8	13	14	0.02215	0.00E+00	18	58	0.01440	6.38E-07	8	46	0.01104	0.00E+00
	10,000	x1	4	5	0.05099	0.00E+00	14	43	0.12490	8.65E-07	103	1155	1.27990	0.00E+00
		x2	18	19	0.21214	0.00E+00	13	42	0.06798	1.72E-07	32	284	0.30602	0.00E+00
		x3	4	5	0.07857	0.00E+00	6	15	0.03300	4.90E-07	16	122	0.25066	0.00E+00
		x4	4	5	0.04852	0.00E+00	6	15	0.03077	1.65E-07	92	1015	1.71821	0.00E+00
		x5	9	10	0.11444	0.00E+00	11	33	0.06174	2.99E-07	25	45	0.15900	9.34E-07
		x6	5	6	0.06893	0.00E+00	10	28	0.05002	2.13E-07	92	1030	1.19462	0.00E+00
		x7	14	15	0.17102	0.00E+00	8	23	0.06200	8.54E-07	9	60	0.13710	0.00E+00
		x8	14	15	0.16226	0.00E+00	17	61	0.11029	9.75E-07	1001	1082	2.36323	6.55E+04
	100,000	x1	6	7	0.50356	0.00E+00	16	46	0.63699	4.68E-07	30	199	1.74338	0.00E+00
		x2	19	20	1.70708	0.00E+00	10	27	0.36805	1.28E-08	28	286	2.20160	0.00E+00
		x3	6	7	0.51893	0.00E+00	7	17	0.29252	1.13E-07	101	1163	8.55412	0.00E+00
		x4	4	5	0.34368	0.00E+00	6	15	0.26364	9.96E-09	26	43	0.63220	8.31E-07
		x5	11	12	0.97077	0.00E+00	10	31	0.41796	5.01E-07	29	72	0.86051	7.39E-07
		x6	11	12	1.01489	0.00E+00	8	21	0.30025	4.92E-08	27	68	0.82920	5.14E-07
		x7	18	19	1.65894	0.00E+00	8	23	0.41963	9.48E-08	29	71	0.84210	5.86E-07
		x8	15	16	1.34973	0.00E+00	11	34	0.42575	7.74E-07	9	58	0.52940	0.00E+00
P2	100	x1	2	4	0.00584	9.83E-07	5	11	0.01489	8.95E-08	18	20	0.01423	6.20E-07
		x2	4	6	0.00556	3.31E-07	7	15	0.01311	1.74E-07	23	25	0.01436	6.10E-07
		x3	5	7	0.00835	4.29E-07	6	12	0.00810	9.99E-07	24	26	0.01319	5.57E-07
		x4	5	7	0.00640	9.76E-07	14	38	0.04578	9.56E-07	24	26	0.04415	7.58E-07
		x5	6	8	0.00936	1.29E-07	9	18	0.02229	1.10E-07	26	28	0.01594	8.44E-07
		x6	6	8	0.01050	9.05E-07	10	26	0.00969	8.72E-07	23	25	0.01364	7.66E-07
		x7	5	7	0.00931	7.25E-07	14	30	0.03262	7.75E-07	26	28	0.03901	5.62E-07
		x8	4	6	0.00662	9.39E-07	9	25	0.01071	5.40E-08	21	23	0.02760	8.49E-07
	10,000	x1	3	5	0.02577	2.11E-07	5	11	0.03185	2.71E-07	19	21	0.08529	9.65E-07
		x2	4	6	0.02683	9.71E-07	7	15	0.04860	5.14E-07	24	26	0.11387	9.45E-07
		x3	6	8	0.03915	2.67E-08	7	13	0.03646	8.67E-08	25	27	0.09513	8.62E-07
		x4	6	8	0.03951	1.19E-07	14	38	0.08905	1.68E-07	26	28	0.10267	5.86E-07
		x5	6	8	0.04087	3.76E-07	9	18	0.07042	4.65E-07	28	30	0.18724	6.52E-07
		x6	7	9	0.05484	2.58E-07	11	27	0.07024	7.88E-08	25	27	0.14486	5.93E-07
		x7	6	8	0.03782	1.59E-07	15	32	0.08824	3.92E-09	27	29	0.11938	8.68E-07
		x8	5	7	0.03146	1.62E-07	12	32	0.07698	5.56E-08	23	25	0.12050	6.59E-07
	100,000	x1	3	5	0.15491	6.43E-07	5	11	0.21010	8.52E-07	21	23	0.53615	7.62E-07
		x2	5	7	0.24336	1.95E-08	8	16	0.35311	2.72E-08	26	28	0.66623	7.46E-07
		x3	6	8	0.25928	8.34E-08	7	13	0.32370	2.70E-07	27	29	0.69745	6.80E-07
		x4	6	8	0.26963	3.73E-07	17	45	0.72980	5.08E-08	27	29	0.72102	9.25E-07
		x5	7	9	0.29765	3.60E-08	10	19	0.37575	3.93E-08	30	32	0.78366	5.14E-07
		x6	7	9	0.35906	8.12E-07	11	27	0.47475	2.47E-07	26	28	0.66078	9.36E-07
		x7	6	8	0.25959	4.99E-07	15	32	0.57791	2.94E-08	29	31	0.76046	6.85E-07
		x8	5	7	0.21840	5.11E-07	12	32	0.56030	1.73E-07	25	27	0.64154	5.20E-07

Table 2. Numerical of UMCD, MDDYM & DTCG1 methods for problems 3 and 4.

Problem	Dim	IP	UMCD				MDDYM				DTCG1			
			Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $
P3	100	x1	1	3	0.00405	5.21E-07	5	11	0.00649	7.68E-08	18	20	0.00967	6.03E-07
		x2	4	6	0.00493	6.00E-07	4	6	0.00522	5.22E-08	22	24	0.01377	9.29E-07
		x3	5	7	0.00535	8.96E-08	4	6	0.00741	1.27E-07	23	25	0.02699	7.28E-07
		x4	5	7	0.00573	2.23E-07	4	6	0.00531	2.71E-08	23	25	0.01227	8.92E-07
		x5	4	6	0.00651	2.15E-07	9	21	0.00871	3.02E-07	24	26	0.01291	8.51E-07
		x6	5	7	0.00544	1.86E-08	4	6	0.00515	9.44E-08	23	25	0.01735	5.54E-07
		x7	5	7	0.00836	4.54E-07	9	23	0.01367	4.88E-07	24	26	0.01546	7.63E-07
		x8	3	5	0.00551	8.29E-08	3	5	0.00485	3.80E-09	21	23	0.01426	7.52E-07
	10,000	x1	2	4	0.01347	1.65E-07	5	11	0.02473	2.43E-07	19	21	0.10561	9.54E-07
		x2	5	7	0.02757	1.93E-08	4	6	0.02067	1.65E-07	24	26	0.06887	7.35E-07
		x3	5	7	0.02884	2.83E-07	4	6	0.01880	4.02E-07	25	27	0.07398	5.75E-07
		x4	5	7	0.02524	7.06E-07	4	6	0.01674	8.56E-08	25	27	0.11716	7.05E-07
		x5	4	6	0.02281	6.79E-07	9	21	0.05192	9.55E-07	26	28	0.08110	6.73E-07
		x6	5	7	0.02662	5.89E-08	4	6	0.01662	2.99E-07	24	26	0.06970	8.76E-07
		x7	6	8	0.03420	8.57E-08	10	24	0.06219	5.62E-08	26	28	0.08495	6.03E-07
		x8	3	5	0.01816	2.62E-07	3	5	0.02704	1.20E-08	23	25	0.06769	5.95E-07
	100,000	x1	2	4	0.09912	5.21E-07	5	11	0.17670	7.68E-07	21	23	0.42251	7.54E-07
		x2	5	7	0.18389	6.11E-08	4	6	0.11846	5.22E-07	26	28	0.51949	5.81E-07
		x3	5	7	0.19643	8.96E-07	5	7	0.13611	3.34E-08	26	28	0.65490	9.10E-07
		x4	6	8	0.20836	1.35E-07	4	6	0.14656	2.71E-07	27	29	0.58802	5.58E-07
		x5	5	7	0.19077	5.24E-09	10	22	0.35069	9.62E-08	28	30	0.61661	5.32E-07
		x6	5	7	0.18573	1.86E-07	4	6	0.11910	9.44E-07	26	28	0.52935	6.93E-07
		x7	6	8	0.22130	2.71E-07	10	24	0.32169	1.78E-07	27	29	0.54962	9.53E-07
		x8	3	5	0.12949	8.29E-07	3	5	0.12552	3.80E-08	24	26	0.48600	9.40E-07
P4	100	x1	2	4	0.00421	6.79E-07	5	11	0.01502	8.51E-08	18	20	0.01046	6.09E-07
		x2	6	8	0.00636	7.19E-07	10	26	0.01235	3.96E-07	23	25	0.01253	6.12E-07
		x3	5	7	0.00568	5.71E-07	8	16	0.01254	1.54E-07	24	26	0.01194	5.79E-07
		x4	5	7	0.00797	2.11E-08	14	32	0.01171	9.25E-08	24	26	0.01520	8.15E-07
		x5	5	7	0.00859	8.54E-07	13	29	0.01011	3.14E-08	27	29	0.01864	7.33E-07
		x6	5	7	0.00707	2.46E-07	6	12	0.00468	1.16E-07	23	25	0.01120	7.76E-07
		x7	6	8	0.00707	2.00E-08	13	30	0.01238	5.36E-08	26	28	0.01676	7.95E-07
		x8	4	6	0.00606	9.67E-07	10	26	0.01546	6.07E-08	21	23	0.01189	8.35E-07
	10,000	x1	3	5	0.01823	2.05E-07	5	11	0.02307	2.69E-07	19	21	0.05790	9.63E-07
		x2	7	9	0.04195	2.07E-07	11	27	0.05787	9.78E-09	24	26	0.07274	9.67E-07
		x3	6	8	0.03112	1.03E-07	8	16	0.08280	4.87E-07	25	27	0.07898	9.16E-07
		x4	5	7	0.02690	6.67E-08	14	32	0.06476	2.93E-07	26	28	0.07790	6.44E-07
		x5	6	8	0.03229	2.15E-07	13	29	0.07800	9.93E-08	29	31	0.08593	5.79E-07
		x6	5	7	0.02941	7.78E-07	6	12	0.02703	3.67E-07	25	27	0.07716	6.13E-07
		x7	6	8	0.03457	6.32E-08	13	30	0.08342	1.70E-07	28	30	0.08366	6.28E-07
		x8	5	7	0.02739	1.69E-07	10	26	0.09130	1.92E-07	23	25	0.07009	6.60E-07
	100,000	x1	3	5	0.13002	6.49E-07	5	11	0.17798	8.51E-07	21	23	0.46367	7.62E-07
		x2	7	9	0.28717	6.55E-07	11	27	0.40900	3.09E-08	26	28	0.52167	7.65E-07
		x3	6	8	0.22485	3.27E-07	9	17	0.27010	2.59E-08	27	29	0.55878	7.24E-07
		x4	5	7	0.19709	2.11E-07	14	32	0.43011	9.25E-07	28	30	0.57757	5.09E-07
		x5	6	8	0.23647	6.81E-07	13	29	0.43131	3.14E-07	30	32	0.62163	9.16E-07
		x6	6	8	0.21179	4.71E-08	7	13	0.22683	5.03E-08	26	28	0.55034	9.70E-07
		x7	6	8	0.22048	2.00E-07	13	30	0.46398	5.36E-07	29	31	0.59930	9.94E-07
		x8	5	7	0.19480	5.33E-07	10	26	0.38605	6.07E-07	25	27	0.51780	5.22E-07

Table 3. Numerical of UMCD, MDDYM & DTCG1 methods for problems 5 and 6.

Problem	Dim	IP	UMCD				MDDYM				DTCG1			
			Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $
P5	100	x1	3	5	0.00521	5.77E-08	3	5	0.00726	1.13E-08	18	20	0.00915	5.97E-07
		x2	4	6	0.00448	6.42E-07	5	9	0.01017	1.50E-08	22	24	0.00982	7.30E-07
		x3	4	6	0.00692	9.05E-09	8	18	0.01116	3.65E-08	22	24	0.01166	9.94E-07
		x4	5	7	0.00742	1.53E-07	21	62	0.01553	5.63E-07	23	25	0.01444	5.54E-07
		x5	5	7	0.00746	1.75E-07	9	22	0.02207	1.03E-08	18	20	0.01136	5.80E-07
		x6	5	7	0.00686	1.15E-07	7	15	0.00694	1.46E-08	22	24	0.00949	8.31E-07
		x7	5	7	0.00876	5.88E-07	10	21	0.00613	4.14E-08	22	24	0.01248	9.21E-07
		x8	5	7	0.00641	4.86E-08	2	4	0.00849	2.99E-08	21	23	0.01139	6.82E-07
	10,000	x1	3	5	0.02278	1.82E-07	3	5	0.01208	3.57E-08	19	21	0.05288	9.44E-07
		x2	5	7	0.02484	2.31E-08	5	9	0.03458	4.76E-08	24	26	0.06366	5.77E-07
		x3	4	6	0.01991	2.86E-08	8	18	0.03641	1.16E-07	24	26	0.06583	7.86E-07
		x4	5	7	0.03262	4.84E-07	22	65	0.09893	9.08E-07	24	26	0.08210	8.76E-07
		x5	5	7	0.03566	5.55E-07	9	22	0.04446	3.27E-08	19	21	0.09265	9.17E-07
		x6	5	7	0.02447	3.64E-07	7	15	0.02887	4.62E-08	24	26	0.08414	6.57E-07
		x7	6	8	0.04114	4.21E-08	10	21	0.03511	1.31E-07	24	26	0.06352	7.28E-07
		x8	5	7	0.02399	1.54E-07	2	4	0.02380	9.45E-08	23	25	0.06055	5.39E-07
	100,000	x1	3	5	0.10503	5.77E-07	3	5	0.09026	1.13E-07	21	23	0.41393	7.46E-07
		x2	5	7	0.16283	7.31E-08	5	9	0.12978	1.50E-07	25	27	0.42275	9.13E-07
		x3	4	6	0.14198	9.05E-08	8	18	0.24298	3.65E-07	26	28	0.47482	6.21E-07
		x4	6	8	0.18711	9.98E-08	23	67	0.68963	2.67E-08	26	28	0.41348	6.93E-07
		x5	6	8	0.21443	1.30E-07	9	22	0.42998	1.03E-07	21	23	0.34320	7.25E-07
		x6	6	8	0.17874	7.04E-08	7	15	0.25166	1.46E-07	26	28	0.42702	5.20E-07
		x7	6	8	0.23317	1.33E-07	10	21	0.23967	4.14E-07	26	28	0.47965	5.76E-07
		x8	5	7	0.15034	4.86E-07	2	4	0.05786	2.99E-07	24	26	0.42801	8.52E-07
P6	100	x1	3	4	0.00839	0.00E+00	46	338	0.08394	0.00E+00	5	17	0.01222	0.00E+00
		x2	3	4	0.00855	0.00E+00	1000	1000	10.00000	NAN	4	27	0.01917	0.00E+00
		x3	3	4	0.00985	0.00E+00	1000	1000	1.00000	NAN	4	28	0.00795	0.00E+00
		x4	3	4	0.01189	0.00E+00	152	980	0.17760	9.03E-07	4	30	0.02202	0.00E+00
		x5	2	3	0.00952	0.00E+00	7	56	0.01307	0.00E+00	2	15	0.00794	0.00E+00
		x6	3	4	0.00860	0.00E+00	34	239	0.07584	0.00E+00	4	27	0.01482	0.00E+00
		x7	2	3	0.00995	0.00E+00	140	908	0.20564	9.46E-07	3	26	0.01125	0.00E+00
		x8	3	4	0.01060	0.00E+00	33	245	0.06110	0.00E+00	4	27	0.01564	0.00E+00
	10,000	x1	3	4	0.05055	0.00E+00	10	73	0.14858	0.00E+00	4	27	0.08237	0.00E+00
		x2	3	4	0.05251	0.00E+00	8	57	0.11160	0.00E+00	3	15	0.09650	0.00E+00
		x3	3	4	0.05212	0.00E+00	9	65	0.14487	0.00E+00	3	15	0.03633	0.00E+00
		x4	2	3	0.03333	0.00E+00	11	76	0.13786	0.00E+00	4	27	0.06110	0.00E+00
		x5	2	3	0.04027	0.00E+00	4	35	0.07081	0.00E+00	2	14	0.04709	0.00E+00
		x6	2	3	0.03202	0.00E+00	8	62	0.14194	0.00E+00	3	15	0.07708	0.00E+00
		x7	2	3	0.04137	0.00E+00	8	66	0.13983	0.00E+00	2	14	0.02688	0.00E+00
		x8	3	4	0.05385	0.00E+00	14	94	0.18096	0.00E+00	3	15	0.05561	0.00E+00
	100,000	x1	3	4	0.42789	0.00E+00	11	76	1.30552	0.00E+00	3	15	0.34265	0.00E+00
		x2	3	4	0.43296	0.00E+00	5	39	0.65111	0.00E+00	3	15	0.27402	0.00E+00
		x3	2	3	0.24661	0.00E+00	4	29	0.51169	0.00E+00	3	15	0.27954	0.00E+00
		x4	2	3	0.25837	0.00E+00	10	80	1.24010	0.00E+00	3	15	0.26327	0.00E+00
		x5	2	3	0.45543	0.00E+00	3	24	0.36586	0.00E+00	2	14	0.23537	0.00E+00
		x6	2	3	0.30170	0.00E+00	4	29	0.44399	0.00E+00	3	15	0.31806	0.00E+00
		x7	2	3	0.35130	0.00E+00	6	48	0.82400	0.00E+00	2	14	0.22764	0.00E+00
		x8	3	4	0.42526	0.00E+00	9	73	1.14171	0.00E+00	3	15	0.26653	0.00E+00

Table 4. Numerical of UMCD, MDDYM & DTCG1 methods for problems 7 and 8.

Problem	Dim	IP	UMCD				MDDYM				DTCG1			
			Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $
P7	100	x1	2	3	0.00983	0.00E+00	10	40	0.01654	9.18E-07	3	21	0.01528	9.21E-08
		x2	4	5	0.01344	0.00E+00	8	31	0.01130	6.62E-07	4	25	0.01546	2.26E-07
		x3	3	4	0.00996	0.00E+00	10	37	0.02569	8.05E-08	4	24	0.00975	4.24E-07
		x4	2	3	0.01092	0.00E+00	9	32	0.01675	8.46E-07	4	23	0.00892	1.88E-07
		x5	3	4	0.01234	0.00E+00	12	39	0.01611	4.17E-07	5	26	0.00956	8.28E-08
		x6	2	3	0.01022	0.00E+00	8	30	0.01357	8.53E-07	4	24	0.01195	1.77E-08
		x7	2	3	0.01010	0.00E+00	11	38	0.02002	1.56E-07	4	21	0.01527	8.27E-07
		x8	2	3	0.00768	0.00E+00	20	84	0.02336	7.22E-07	4	26	0.01213	6.94E-08
	10,000	x1	2	3	0.06824	0.00E+00	11	43	0.14888	1.47E-07	3	21	0.12112	2.72E-07
		x2	4	5	0.10523	0.00E+00	9	34	0.15687	4.27E-07	4	25	0.05985	6.66E-07
		x3	3	4	0.07267	0.00E+00	10	37	0.10337	2.21E-07	5	29	0.07142	1.98E-08
		x4	2	3	0.04905	0.00E+00	11	40	0.10614	4.79E-07	4	23	0.09871	5.47E-07
		x5	3	4	0.07045	0.00E+00	13	42	0.12128	9.73E-08	5	26	0.06463	2.39E-07
		x6	2	3	0.05264	0.00E+00	6	23	0.09211	9.73E-07	4	24	0.07148	4.95E-08
		x7	2	3	0.04642	0.00E+00	11	38	0.13869	5.50E-07	5	26	0.06866	3.71E-08
		x8	2	3	0.06676	0.00E+00	19	79	0.18702	1.47E-07	4	26	0.10791	2.04E-07
	100,000	x1	2	3	0.40501	0.00E+00	11	43	0.90189	4.68E-07	3	21	0.36597	8.55E-07
		x2	4	5	0.81463	0.00E+00	10	38	0.82053	6.24E-07	5	30	0.53294	3.31E-08
		x3	3	4	0.58979	0.00E+00	10	37	0.74230	6.87E-07	5	29	0.59631	6.21E-08
		x4	2	3	0.40452	0.00E+00	12	43	0.94642	4.38E-08	5	28	0.47195	2.72E-08
		x5	3	4	0.54303	0.00E+00	13	42	0.94394	3.09E-07	5	26	0.46243	7.49E-07
		x6	2	3	0.37559	0.00E+00	12	43	0.93140	1.86E-07	4	24	0.53569	1.55E-07
		x7	2	3	0.36851	0.00E+00	12	41	0.94784	6.84E-07	5	26	0.48445	1.16E-07
		x8	2	3	0.36777	0.00E+00	19	79	1.60547	5.84E-07	4	26	0.44721	6.40E-07
P8	100	x1	5	7	0.01136	9.18E-10	9	25	0.01476	3.16E-07	6	8	0.02509	3.03E-07
		x2	5	7	0.01256	2.65E-08	8	21	0.02121	2.11E-07	6	8	0.00575	9.77E-08
		x3	5	7	0.01537	1.47E-07	7	19	0.01729	7.34E-07	5	7	0.00825	4.28E-07
		x4	3	5	0.00977	2.96E-07	6	17	0.01466	4.67E-07	4	6	0.00492	6.75E-07
		x5	5	7	0.01473	2.63E-07	15	48	0.02332	4.57E-07	6	8	0.00507	5.95E-07
		x6	4	6	0.01137	9.69E-07	8	21	0.01054	1.83E-07	6	8	0.00733	7.04E-08
		x7	5	7	0.01011	2.87E-07	6	15	0.00987	3.03E-07	5	7	0.00516	6.75E-07
		x8	5	7	0.01265	1.20E-07	7	19	0.00618	7.72E-08	6	8	0.00804	2.09E-07
	10,000	x1	5	7	0.06854	2.90E-09	9	25	0.08002	9.99E-07	6	8	0.02902	9.58E-07
		x2	5	7	0.06297	8.37E-08	8	21	0.04202	6.66E-07	6	8	0.02848	3.09E-07
		x3	5	7	0.07992	4.64E-07	8	21	0.03989	2.11E-07	6	8	0.03310	8.64E-08
		x4	3	5	0.04594	9.38E-07	7	19	0.04275	2.07E-07	5	7	0.02787	1.36E-07
		x5	5	7	0.06013	8.32E-07	16	50	0.11420	2.29E-07	7	9	0.03078	1.20E-07
		x6	5	7	0.06760	2.76E-07	8	21	0.04725	5.79E-07	6	8	0.02689	2.22E-07
		x7	5	7	0.06589	9.06E-07	6	15	0.03220	9.58E-07	6	8	0.04191	1.36E-07
		x8	5	7	0.06528	3.80E-07	7	19	0.04509	2.44E-07	6	8	0.02953	6.62E-07
	100,000	x1	5	7	0.47364	9.18E-09	10	27	0.45492	1.76E-07	7	9	0.22843	1.93E-07
		x2	5	7	0.46010	2.65E-07	9	23	0.40843	1.38E-07	6	8	0.27425	9.77E-07
		x3	6	8	0.58311	8.89E-08	8	21	0.35741	6.68E-07	6	8	0.23392	2.73E-07
		x4	4	6	0.38535	5.46E-08	7	19	0.31560	6.55E-07	5	7	0.18196	4.31E-07
		x5	6	8	0.52697	1.25E-07	16	50	0.76155	7.23E-07	7	9	0.27878	3.80E-07
		x6	5	7	0.48573	8.73E-07	9	23	0.38114	1.55E-07	6	8	0.20102	7.04E-07
		x7	6	8	0.57955	7.26E-08	7	17	0.30548	7.91E-09	6	8	0.22101	4.31E-07
		x8	6	8	0.53726	4.92E-08	7	19	0.27165	7.72E-07	7	9	0.22957	1.34E-07

Table 5. Numerical of UMCD, MDDYM & DTCG1 methods for problems 9 and 10.

Problem	Dim	IP	UMCD				MDDYM				DTCG1			
			Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $
p9	100	x1	1	2	0.00576	0.00E+00	3	13	0.01774	2.29E-08	4	51	0.02559	4.25E-08
		x2	1	2	0.00585	0.00E+00	11	55	0.01757	9.95E-07	5	62	0.01398	4.79E-08
		x3	1	2	0.00608	0.00E+00	2	6	0.01028	0.00E+00	5	63	0.02335	9.88E-08
		x4	1	2	0.00350	0.00E+00	1	2	0.00265	0.00E+00	5	63	0.01456	9.45E-08
		x5	1	2	0.00404	0.00E+00	9	42	0.02898	6.14E-07	1	6	0.00465	0.00E+00
		x6	1	2	0.00617	0.00E+00	6	28	0.00710	1.17E-07	5	62	0.01665	3.89E-08
		x7	1	2	0.00327	0.00E+00	1	3	0.00285	0.00E+00	1	3	0.00293	0.00E+00
		x8	1	2	0.00471	0.00E+00	2	10	0.00989	1.96E-07	5	62	0.02565	4.05E-08
	10,000	x1	1	2	0.01833	0.00E+00	3	13	0.03811	7.25E-08	4	51	0.06791	1.34E-07
		x2	1	2	0.02078	0.00E+00	12	58	0.10703	4.37E-08	5	62	0.13277	1.52E-07
		x3	1	2	0.02853	0.00E+00	2	6	0.01617	0.00E+00	5	63	0.08501	3.13E-07
		x4	1	2	0.00851	0.00E+00	1	2	0.01422	0.00E+00	5	63	0.16091	2.99E-07
		x5	1	2	0.01172	0.00E+00	10	45	0.12989	3.06E-08	1	6	0.01353	0.00E+00
		x6	1	2	0.02217	0.00E+00	6	28	0.06078	3.69E-07	5	62	0.08562	1.23E-07
		x7	1	2	0.01247	0.00E+00	1	3	0.01702	0.00E+00	1	3	0.01811	0.00E+00
		x8	1	2	0.02245	0.00E+00	2	10	0.01802	6.18E-07	5	62	0.15127	1.28E-07
	100,000	x1	1	2	0.13106	0.00E+00	3	13	0.16577	2.29E-07	4	51	0.44893	4.25E-07
		x2	1	2	0.15448	0.00E+00	12	58	0.67073	1.38E-07	5	62	0.54013	4.79E-07
		x3	1	2	0.15167	0.00E+00	2	6	0.11161	0.00E+00	5	63	0.54475	9.88E-07
		x4	1	2	0.04368	0.00E+00	1	2	0.04261	0.00E+00	5	63	0.66749	9.45E-07
		x5	1	2	0.07560	0.00E+00	10	45	0.57314	9.68E-08	1	6	0.14915	0.00E+00
		x6	1	2	0.16841	0.00E+00	7	31	0.37625	3.90E-09	5	62	0.57907	3.89E-07
		x7	1	2	0.07196	0.00E+00	1	3	0.06191	0.00E+00	1	3	0.08504	0.00E+00
		x8	1	2	0.13712	0.00E+00	3	13	0.21464	7.38E-08	5	62	0.57875	4.05E-07
p8	100	x1	1	2	0.00755	0.00E+00	3	17	0.00671	3.90E-08	1	13	0.00857	0.00E+00
		x2	1	2	0.00509	0.00E+00	13	77	0.01799	3.98E-08	1	13	0.00462	0.00E+00
		x3	1	2	0.00464	0.00E+00	5	27	0.01040	4.93E-07	1	13	0.00612	0.00E+00
		x4	1	2	0.00671	0.00E+00	14	83	0.05777	1.15E-08	1	13	0.00899	0.00E+00
		x5	1	2	0.00667	0.00E+00	12	66	0.03522	2.49E-07	1	13	0.01943	0.00E+00
		x6	1	2	0.00667	0.00E+00	9	51	0.01460	3.71E-08	1	13	0.01309	0.00E+00
		x7	1	2	0.00474	0.00E+00	13	71	0.04436	2.83E-07	1	13	0.00668	0.00E+00
		x8	1	2	0.00468	0.00E+00	4	21	0.00680	1.48E-07	1	13	0.00825	0.00E+00
	10,000	x1	1	2	0.02513	0.00E+00	3	17	0.03592	1.23E-07	1	13	0.04248	0.00E+00
		x2	1	2	0.02792	0.00E+00	13	77	0.20699	1.26E-07	1	13	0.02220	0.00E+00
		x3	1	2	0.02721	0.00E+00	6	31	0.06834	1.87E-08	1	13	0.04022	0.00E+00
		x4	1	2	0.02680	0.00E+00	14	83	0.15190	3.63E-08	1	13	0.04513	0.00E+00
		x5	1	2	0.02727	0.00E+00	12	66	0.12581	7.88E-07	1	13	0.02469	0.00E+00
		x6	1	2	0.03042	0.00E+00	9	51	0.09754	1.17E-07	1	13	0.04831	0.00E+00
		x7	1	2	0.02655	0.00E+00	13	71	0.13567	8.94E-07	1	13	0.04894	0.00E+00
		x8	1	2	0.02428	0.00E+00	4	21	0.04477	4.67E-07	1	13	0.05580	0.00E+00
	100,000	x1	1	2	0.17512	0.00E+00	3	17	0.27594	3.90E-07	1	13	0.16893	0.00E+00
		x2	1	2	0.19268	0.00E+00	13	77	1.08727	3.98E-07	1	13	0.19849	0.00E+00
		x3	1	2	0.21737	0.00E+00	6	31	0.53243	5.92E-08	1	13	0.21012	0.00E+00
		x4	1	2	0.20273	0.00E+00	14	83	1.17916	1.15E-07	1	13	0.21843	0.00E+00
		x5	1	2	0.19913	0.00E+00	13	70	1.19117	2.46E-08	1	13	0.19528	0.00E+00
		x6	1	2	0.19263	0.00E+00	9	51	0.74924	3.71E-07	1	13	0.18907	0.00E+00
		x7	1	2	0.19943	0.00E+00	14	75	1.10069	1.27E-07	1	13	0.21608	0.00E+00
		x8	1	2	0.18627	0.00E+00	5	25	0.37949	4.78E-08	1	13	0.16808	0.00E+00

Table 6 presents the numerical comparison of the proposed algorithm with the existing CHCG and ZANG algorithms using problem Problems 11 and 12. In the table, significant performance of the proposed algorithm is observed over MCHCG and ZANG algorithms. In view of the above, the proposed method is considered as the optimal choice due to its excellence in the number of iterations and overall CPU time efficiency.

Table 6. Experiments of UMCD, MCHCG & ZANG methods for problems 11 & 12.

Problem	Dim	IP	UMCD				MCHCG				ZANG			
			Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $	Iter	Fval	Time(s)	$\ \mathcal{H}_k\ $
11	1000	x1	15	16	0.03816	0.00E+00	6	7	0.02527	1.50E-07	10	12	0.02170	6.42E-07
		x2	9	10	0.03703	0.00E+00	6	8	0.02020	3.29E-07	13	14	0.00902	7.17E-07
		x3	4	5	0.00993	0.00E+00	6	8	0.02159	1.43E-07	13	14	0.01132	9.79E-07
		x4	4	5	0.01202	0.00E+00	6	8	0.02249	6.13E-07	13	15	0.01018	4.40E-07
		x5	4	5	0.01421	0.00E+00	6	8	0.02213	2.25E-07	13	15	0.01131	8.72E-07
		x6	4	5	0.01246	0.00E+00	6	8	0.02175	2.78E-07	13	14	0.01177	8.16E-07
		x7	2	3	0.00902	0.00E+00	6	8	0.02330	7.48E-07	13	15	0.00834	6.47E-07
		x8	4	5	0.01425	0.00E+00	6	8	0.02475	1.15E-07	12	14	0.01199	4.69E-07
	10,000	x1	4	5	0.06147	0.00E+00	6	7	0.11607	2.81E-07	11	13	0.04875	5.18E-07
		x2	14	15	0.18633	0.00E+00	6	8	0.13784	9.77E-07	13	15	0.05500	7.83E-07
		x3	9	10	0.11854	0.00E+00	6	8	0.13576	3.88E-07	14	15	0.06108	7.60E-07
		x4	4	5	0.07767	0.00E+00	8	9	0.15722	1.02E-07	14	15	0.05532	8.69E-07
		x5	4	5	0.05527	0.00E+00	8	9	0.14186	1.25E-07	14	16	0.05873	7.08E-07
		x6	13	14	0.17925	0.00E+00	6	8	0.12612	8.07E-07	13	15	0.05752	8.97E-07
		x7	2	3	0.03364	0.00E+00	7	9	0.15516	3.37E-07	14	16	0.05924	5.33E-07
		x8	6	7	0.09767	0.00E+00	6	8	0.14821	9.00E-08	13	14	0.05091	9.12E-07
	100,000	x1	6	7	0.60125	0.00E+00	6	7	0.79950	7.90E-07	12	14	0.33064	4.44E-07
		x2	6	7	0.56274	0.00E+00	8	9	1.16167	2.30E-07	14	16	0.37496	6.69E-07
		x3	18	19	1.77732	0.00E+00	7	9	1.10606	9.02E-08	14	16	0.38147	9.34E-07
		x4	15	16	1.46303	0.00E+00	5	7	0.83329	1.90E-07	15	16	0.39235	7.43E-07
		x5	8	9	0.74268	9.80E-07	7	9	1.14606	3.13E-07	15	17	0.40514	6.08E-07
		x6	6	7	0.61239	0.00E+00	8	9	1.13741	1.89E-07	14	16	0.38782	7.66E-07
		x7	2	3	0.24444	0.00E+00	9	10	1.40988	2.37E-07	15	17	0.41061	4.59E-07
		x8	5	6	0.40187	0.00E+00	6	8	0.96455	1.73E-07	14	15	0.36014	7.78E-07
12	1000	x1	3	5	0.00895	1.51E-07	5	7	0.01557	9.09E-08	10	12	0.00909	5.96E-07
		x2	5	7	0.02239	7.73E-07	6	8	0.01930	1.37E-07	13	14	0.01035	7.64E-07
		x3	6	8	0.01628	1.05E-07	6	8	0.01651	1.46E-07	13	15	0.01111	4.72E-07
		x4	5	7	0.02212	6.27E-07	6	8	0.02404	1.15E-07	13	15	0.01170	5.77E-07
		x5	6	8	0.02581	2.08E-07	6	8	0.01971	9.05E-07	14	15	0.01159	7.14E-07
		x6	5	7	0.03400	4.33E-07	6	8	0.01731	1.48E-07	13	14	0.01135	9.11E-07
		x7	5	7	0.01919	9.13E-11	6	8	0.01682	8.10E-07	13	15	0.01143	9.57E-07
		x8	4	6	0.00910	5.58E-08	5	7	0.01524	8.43E-07	12	14	0.01086	4.46E-07
	10,000	x1	3	5	0.04222	4.77E-07	5	7	0.10146	2.87E-07	11	13	0.04298	5.16E-07
		x2	6	8	0.07704	1.33E-07	6	8	0.11393	4.33E-07	13	15	0.05166	9.55E-07
		x3	6	8	0.08050	3.31E-07	6	8	0.11742	4.61E-07	14	16	0.05176	4.09E-07
		x4	6	8	0.07692	1.53E-07	6	8	0.11293	3.62E-07	14	16	0.05321	5.00E-07
		x5	6	8	0.08210	6.59E-07	7	9	0.13174	2.17E-07	14	16	0.05245	8.92E-07
		x6	6	8	0.07382	5.77E-08	6	8	0.12074	4.66E-07	14	15	0.04957	7.89E-07
		x7	5	7	0.06370	2.89E-10	7	9	0.13564	1.93E-07	14	16	0.05402	8.30E-07
		x8	4	6	0.07024	1.76E-07	6	8	0.11668	2.09E-07	13	14	0.04749	9.79E-07
	100,000	x1	4	6	0.37268	4.97E-08	5	7	0.77854	8.72E-07	12	14	0.30186	4.47E-07
		x2	6	8	0.57496	4.20E-07	7	9	1.01415	1.06E-07	14	16	0.34227	8.27E-07
		x3	7	9	0.66245	3.98E-08	7	9	1.02851	1.11E-07	15	16	0.35078	8.97E-07
		x4	6	8	0.57558	4.83E-07	7	9	1.02459	8.73E-08	15	17	0.36453	4.33E-07
		x5	7	9	0.66371	6.60E-08	7	9	1.03310	6.82E-07	15	17	0.36946	7.73E-07
		x6	6	8	0.55160	1.83E-07	7	9	1.00286	1.13E-07	14	16	0.35139	9.86E-07
		x7	5	7	0.48683	9.13E-10	7	9	1.00712	6.10E-07	15	17	0.36532	7.19E-07
		x8	4	6	0.38062	5.58E-07	6	8	0.85922	6.62E-07	14	15	0.32884	8.48E-07

Furthermore, using Dolan and Moré performance profile [42], Figures 1–3 were plotted using the data from Tables 1–5. Each figure illustrates the percentage $P(\tau)$ of problems where a specific method works within a factor τ of the best possible execution time. Figure 1 compares the proposed algorithm's (UMCD) performance with that of the MDDYM and DTCG1 algorithms based on the number of iterations. Figures 2 and 3 compare the performances of the three algorithms based on

CPU time and function evaluations, respectively. From the graphs, it is clearly evident that the proposed algorithm (UMCD) outperforms the other two algorithms across the three performance metrics used. Moreover, Figures 4–6 were plotted using the data from Table 6, all of which demonstrate the efficiency of the proposed algorithm over the compared algorithms. While the performance of the MIDS algorithm is acceptable, the UMCD algorithm is considered the best choice when considering the overall performance.

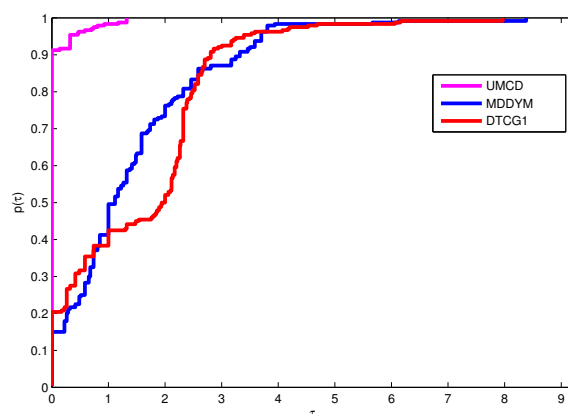


Figure 1. Performance profile for the number of iterations.

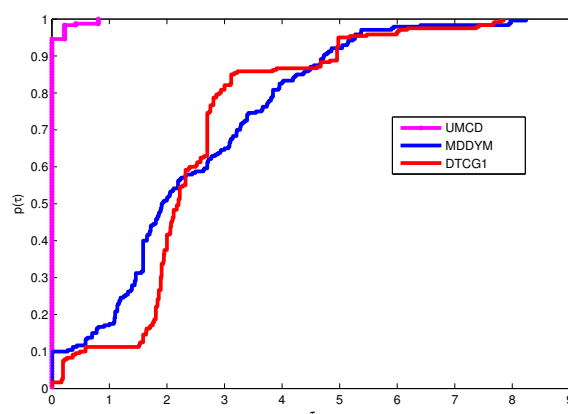


Figure 2. Performance profile for the function evaluations.

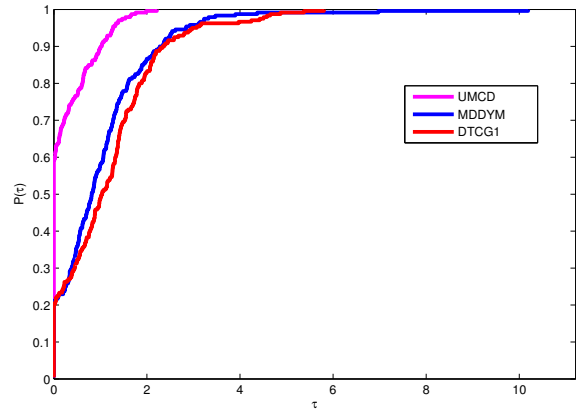


Figure 3. Performance profile for the CPU Time (in seconds).

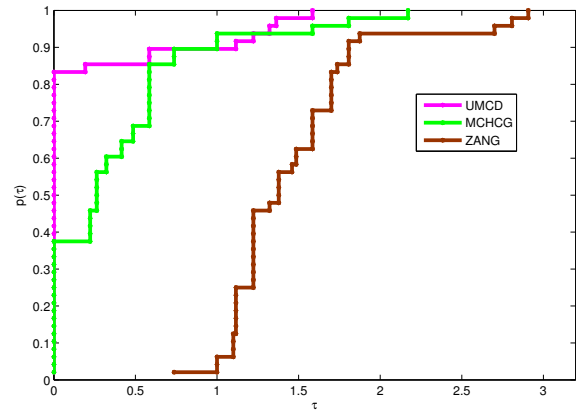


Figure 4. Performance profile of Problems 11 & 12 for the iter.

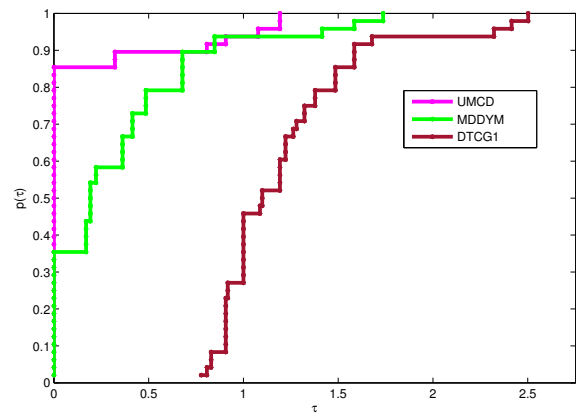


Figure 5. Performance profile of Problems 11 & 12 for the Fval.

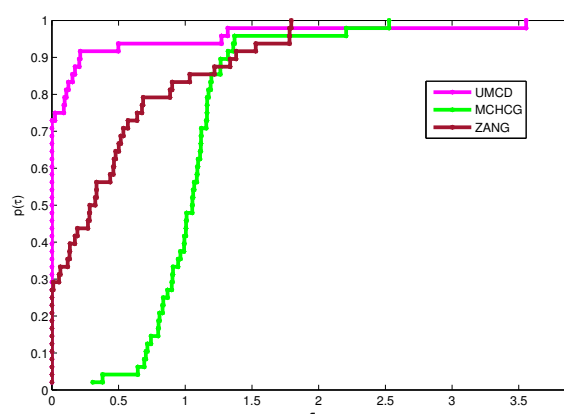


Figure 6. Performance profile of Problems 11 & 12 for the Time(s).

4.1. Application of UMCD algorithm to signal recovery

Among the applications of the proposed algorithm is signal and image recovery in compressive sensing (CS). Compressed sensing, or compressive sensing, is concerned with the reconstruction of sparse signals from incoherent measurements. It asserts that some signals in real applications have a better representation in a specific domain after transformation, while the remaining ones are insignificant or zero. It replaces the traditional sampling technique, thereby reducing the number of samples for better signal sampling and reconstruction. The technique depends mainly on mathematical algorithms, especially derivative-free-based algorithms that require less storage. Therefore, CG algorithms are a good alternative in this respect. The CS theorem is present in biological engineering, medical sciences, and various fields of science and engineering [43, 44]. It uses the following linear equation to recover the sparse signals:

$$\phi = Nx, \quad (4.1)$$

$x \in \mathbb{R}^n$, $\phi \in \mathbb{R}^m$, $N \in \mathbb{R}^{m \times n}$ ($m \ll n$) is a linear operator. The popular approach for solving the sparse solution is to minimize the relation below

$$\min_x \frac{1}{2} \|\phi - Nx\|_2^2 + \tau \|x\|_1, \quad (4.2)$$

τ is a positive parameter (regularized). Equation (4.2) is non-smooth due to the presence of the ℓ_1 -norm; therefore, derivative-free algorithms cannot be applied to solve it. As such, gradient projection for sparse reconstruction (GPSR) was proposed by Figueredo et al. [45] is among the best approaches for solving (4.2). In [45], problem (4.2) is written as

$$x = u - v, \quad u \geq 0, \quad v \geq 0, \quad u, v \in \mathbb{R}^n, \quad (4.3)$$

$u_i = (x_i)_+$, and $v_i = (-x_i)_+$ for $i = 1, 2, \dots, n$, where $(\cdot)_+$ denotes positive-part operator, which is defined as $(x)_+ = \max\{0, x\}$. Using ℓ_1 -norm definition, it implies $\|x\|_1 = e_n^T u + e_n^T v$, with $e_n = (1, 1, \dots, 1)^T \in \mathbb{R}^n$ and problem (4.2) become

$$\min_{u,v} \frac{1}{2} \|\phi - N(u - v)\|_2^2 + \tau e_n^T u + \tau e_n^T v, \quad u, v \geq 0. \quad (4.4)$$

Figueredo et al., [45], demonstrate how to express problem (4.4) in a more formal bound quadratic programming problem as

$$\min_z \frac{1}{2} z^T H z + c^T z, \quad \text{s.t. } z \geq 0, \quad (4.5)$$

where $z = \begin{pmatrix} u \\ v \end{pmatrix}$, $c = \tau e_{2n} + \begin{pmatrix} -b \\ b \end{pmatrix}$, $b = N^T \phi$, $H = \begin{pmatrix} B^T B & -B^T B \\ -B^T B & B^T B \end{pmatrix}$. The matrix H is a positive semi-definite. Equation (4.5) can be transformed to a linear variable inequality (LVT), as follows:

$$\mathcal{H}(z) = \min\{z, H z + c\} = 0. \quad (4.6)$$

The function $\mathcal{H}_k$ in (4.5) is a vector-valued function; it was shown to be Lipschitz continuous and monotone in Lemma 3 of [46] and Lemma 2.2 of [47]. Hence, our proposed algorithm, the UMCD algorithm, could be a good choice for solving it. Numerical comparison is carried out in comparison with the CHCG algorithm in signal reconstruction and recovery. The size of the signal is considered to be $n = 2^{12}$ with $k = 2^{10}$. The error, termed as means square error (MSE), is used to assess the quality of the restoration process; it is measured by

$$MSE = \frac{1}{n} \|\hat{x} - \tilde{x}\|^2, \quad (4.7)$$

where $\hat{x}$ refers to the original signal, and $\tilde{x}$ is the recovered signal. The experiment starts by initializing $x_0 = N^T \phi$, conducting ten (10) trials, and recording the data of interest for each trial. It is set to stop when $\frac{|f_k - f_{k-1}|}{|f_{k-1}|} < 10^{-5}$, where $f(x) = \frac{1}{2} \|Nx - \phi\|_2^2 + \tau \|x\|_1$ is a merit function. The following information is recorded:

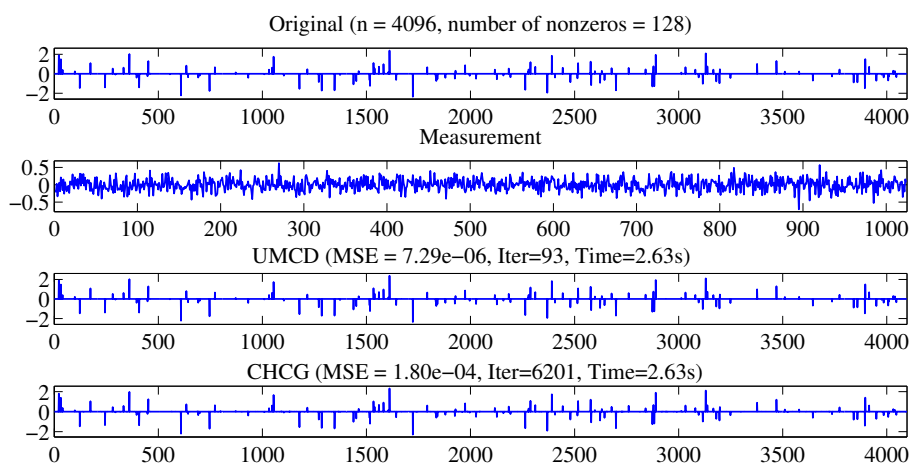


Figure 7. From top to bottom, the original signal, the measurement, the recovery signals recovered by UMCD and CHCG algorithms respectively.

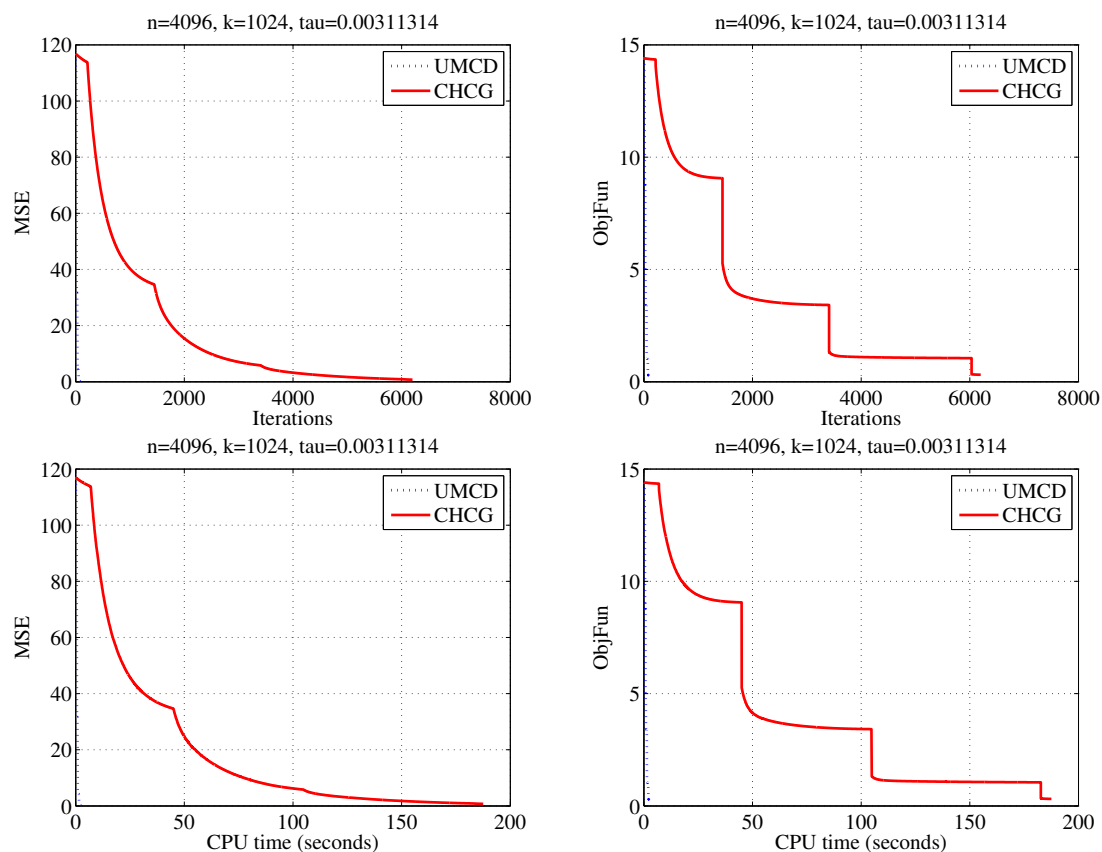


Figure 8. These figures compares UMCD and CHCG algorithms in signal recovery.

Table 7. Ten trials recorded of UMCD and CHCG signal's recovery with ℓ_1 -norm regularization.

	UMCD			CHCG		
	MSE	ITER	TIME(s)	MSE	ITER	TIME(s)
	2.13E-04	91	2.76	2.30E-05	137	2.06
	8.16E-06	92	2.52	1.96E-05	141	2.06
	8.11E-06	89	2.61	2.38E-05	121	3.55
	2.09E-04	91	2.52	2.38E-05	135	2.98
	4.62E-05	84	2.34	1.62E-05	140	3.09
	7.70E-06	96	2.88	1.85E-05	125	3.77
	2.72E-05	87	2.72	4.31E-05	135	3.16
	1.09E-05	84	2.53	1.94E-05	147	3.17
	0.74E-05	75	2.25	2.10E-04	108	0.41
	1.19E-05	72	2.53	2.71E-05	122	4.53
AVERAGE	3.62E-05	92.8	2.56	4.24E-05	132	2.87

As reported in figures 7–8 and table 7, both algorithms performed well in recovering the signals with a reasonable degree of accuracy. But considering the number of iterations (ITER) and CPU time (TIMES) obtained by two algorithms; UMCD and CHCG as reported in table 7, the UMCD algorithm outperformed the performance of the CHCG algorithm.

5. Conclusions

This paper presents “An improved convex constrained conjugate gradient descent method for nonlinear monotone equations with signal recovery applications”. The proposed scheme satisfies the descent condition by conducting an eigenvalue analysis of the search direction matrix. Unlike the methods presented by Yu and Guan [33] and Yu et al. [34], which proved global convergence for the interval $C \in (\frac{1}{4}, \infty)$ only, the proposed method extends the interval to $C \in (0, \infty)$, and global convergence is proved. Furthermore, the numerical performance of the proposed algorithm, in comparison with MDDYM (2021) [37], DTCG1 [6], ZANG [38], and MCHCG [39] algorithms, confirms the efficiency of the proposed algorithm. Being derivative-free, the proposed algorithm is applied to compressive sensing for signal recovery problems. To measure the capacity of the proposed algorithm for signal recovery problems, the experiment is run together with the CHCG algorithm, a well-known algorithm for signal recovery. The results of the experiment are recorded and indicate the efficiency of the proposed algorithm.

Author contributions

H. A. conceptualized the study, developed the methodology, conducted theoretical analysis, and drafted the initial manuscript. A. K. A. and M. Y. W. supervised the research, validated the results, refined the proofs, and critically reviewed the mathematical framework. A. S. H. and K. A. refinement of proofs, implemented the algorithm, conducted computational experiments, and analyzed the data. I. A. R. M. literature review, manuscript draft, and funding. S. M. I. revised the manuscript, result interpretation, and manuscript editing. Y. B. M. provided insights into signal recovery applications and refined the results. E. M. N. verified computational accuracy, assisted with proofreading, formatting, and ensuring compliance with journal requirements. All authors read and approved the final manuscript.

Use of Generative-AI tools declaration

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

Acknowledgments

The authors are grateful to Research Management Centre (RMC), Universiti Utara Malaysia through the geran KHAS with SO code 21777 and the Information Systems and Technology Department of the Kuwait Technical College, Kuwait, for their financial support.

This research is sponsored by the Institutional Based Research (IBR) of the Tertiary Education Trust

Fund (TETFund), Sule Lamido University Kafin Hausa, Nigeria.

Conflict of interest

The authors declare there is no conflict of interests.

References

1. N. Alfredo. Iusem, V. Michael. Solodov, Newton-type methods with generalized distances for constrained optimization, *Optimization*, **41** (1997), 257–278. <https://doi.org/10.1080/02331939708844339>
2. A. S. Halilu, A. Majumder, M. Y. Waziri, K. Ahmed, Signal recovery with convex constrained nonlinea monotone equations through conjugate gradient hybrid approach, *Math. Comput. Simulat.*, **187** (2021), 520–539. <http://doi.org/10.1016/j.matcom.2021.03.020>
3. A. M. Awwal, P. Kumam, A. B. Abubakar, A modified conjugate gradient method for monotone nonlinear equations with convex constraints, *Appl. Numer. Math.*, **145** (2019), 507–520. <https://doi.org/10.1016/j.apnum.2019.05.012>
4. H. Muhammad, A. B. Abubakar, A descent derivative-free algorithm for nonlinear monotone equations with convex constraints, *RAIRO Oper. Res.*, **54** (2020), 489–505. <https://doi.org/10.1051/ro/2020008>
5. H. Abdullahi, A. K. Awasthi, M. Y. Waziri, A. S. Halilu, A scaled three-term conjugate gradient method for convex-constrained monotone nonlinear equations with applications, *J. Phys. Conf. Ser.*, **2267** (2022), 012066 <https://doi.org/10.1088/1742-6596/2267/1/012066>
6. H. Abdullahi, A. K. Awasthi, M. Y. Waziri, A. S. Halilu, Descent three term DY-type conjugate gradient methods for constrained monotone equations with application, *Comp. Appl. Math.*, **41** (2022), 32. <https://doi.org/10.1007/s40314-021-01724-y>
7. B. Khan, P. Singh, Optimal power flow techniques under characterization of conventional and renewable energy sources: A comprehensive analysis, *J. Eng.*, **2017** (2017), 9539506. <https://doi.org/10.1155/2017/9539506>
8. S. Aji, P. Kumaam, P. Siricharoen, A. B. Abubakar, M. M. Yahaya, A modified conjugate descent projection method for monotone nonlinear equations and image restoration, *IEEE Access*, **8** (2020), 158656–158665. <https://doi.org/10.1109/ACCESS.2020.3020334>
9. M. W. Yusuf, L. W. June, M. A. Hassan, Jacobian-free diagonal Newton’s method for solving nonlinear systems with singular Jacobian, *Malasian J. Math. Sci.*, **5** (2011), 241–255.
10. D. Li, M. Fukushima, A global and superlinear convergent Gauss-Newton based BFGS method for symmetric nonlinear equations, *SIAM J. Numer. Anal.*, **37** (1999), 152–172. <https://doi.org/10.1137/S0036142998335704>
11. G. Yuan, X. Lu, A new backtracking inexact BFGS method for symmetric nonlinear equations, *Comput. Math. Appl.*, **55** (2008), 116–129. <https://doi.org/10.1016/j.camwa.2006.12.081>
12. W. Hao, A gradient descent method for solving a system of nonlineare quations, *Appl. Math. Lett.*, **112** (2021), 106739. <https://doi.org/10.1016/j.aml.2020.106739>

13. Z. Papp, S. Rapajic, FR type methods for systems of large-scale nonlinear monotone equations, *Appl. Math. Comput.*, **269** (2015), 816–823. <https://doi.org/10.1016/j.amc.2015.08.002>
14. I. S. Mohammed, M. Mamat, M. R. Abashar, Z. Salleh, A modified nonlinear conjugate gradient method for unconstrained optimization, *Appl. Math. Sci.*, **9** (2015), 54.
15. R. Fletcher, *Practical methods of optimization*, John Wiley & Sons, 2000.
16. X. Yang, Z. Luo, X. Dai, A global convergence of LS-CD hybrid conjugate gradient method, *Adv. Numer. Anal.*, **2013** (2013), 517452. <https://doi.org/10.1155/2013/517452>
17. P. Kaelo, P. Mtagulwa, M. V. Thuto, A globally convergent hybrid conjugate gradient method with strong Wolfe conditions for unconstrained optimization, *Math. Sci.*, **14** (2020), 1–9. <https://doi.org/10.1007/s40096-019-00310-y>
18. S. S. Djordjevic, New hybrid conjugate gradient method as a convex combination of LS and FR methods, *Acta Math. Sci.*, **39** (2019), 214–228. <https://doi.org/10.1007/s10473-019-0117-6>
19. J. K. Liu, J. L. Xu, L. Q. Zhang, Partially symmetrical derivative-free Liu-Storey projection method for convex constrained equations, *Int. J. Comput. Math.*, **96** (2018), 1787–1798. <https://doi.org/10.1080/00207160.2018.1533122>
20. 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>
21. B. S. He, H. Yang, S. L. Wang, Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities, *J. Optimiz. Theory App.*, **106** (2000), 337–356. <https://doi.org/10.1023/A:1004603514434>
22. X. Y. Wang, S. J. Li, X. P. Kou, A self-adaptive three-term conjugate gradient method for monotone nonlinear equations with convex constraints, *Calcolo*, **53** (2016), 133–145. <https://doi.org/10.1007/s10092-015-0140-5>
23. D. H. Li, X. L. Wang, A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations, *Numer. Algebr. Control*, **1** (2011), 71–82. <https://doi.org/10.3934/naco.2011.1.71>
24. L. Zhang, W. Zhou, D. Li, Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search, *Numer. Math.*, **104** (2006), 561–572. <https://doi.org/10.1016/j.amc.2008.03.020>
25. J. Liu, S. Li, Multivariate spectral projection method for convex constrained nonlinear monotone equations, *J. Ind. Manag. Optim.*, **13** (2017), 283–295. <https://doi.org/10.3934/jimo.2016017>
26. G. Yu, S. Niu, J. Ma, Multivariate spectral gradient projection method for non-linear monotone equations with convex constraints, *J. Ind. Manag. Optim.*, **9** (2013), 117–129. <https://doi.org/10.3934/jimo.2013.9.117>
27. Z. Wang, P. Li, X. Li, H. Pham, A modified three-term type CD conjugate gradient algorithm for unconstrained optimization problems, *Math. Probl. Eng.*, **2020** (2020), 4381515. <https://doi.org/10.1155/2020/4381515>
28. A. H. Ibrahim, M. Alshahrani, S. Al-Homidan, Two classes of spectral three-term derivative-free method for solving nonlinear equations with application, *Numer. Algor.*, **96** (2024), 1625–1645. <https://doi.org/10.1007/s11075-023-01679-7>

29. J. Sabi'u, S. Sirisubtawee, An inertial Dai-Liao conjugate method for convex constrained monotone equations that avoids the direction of maximum magnification, *J. Appl. Math. Comput.*, **70** (2024), 4319–4351. <https://doi.org/10.1007/s12190-024-02123-2>
30. L. Zheng, L. Yang, Y. Liang, A conjugate gradient projection method for solving equations with convex constraints, *J. Comput. Appl. Math.*, **375** (2020), 112781. <https://doi.org/10.1016/j.cam.2020.112781>
31. J. 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>
32. S. Chankong, P. Phaochoo, P. Charongrattanasakul, N. Thongpool, A class of derivative free three-term descent Hestenes-Stiefel conjugate gradient algorithms for constrained nonlinear problems, *Results Control Optim.*, **14** (2024), 100372. <https://doi.org/10.1016/j.rico.2024.100372>
33. G. H. Yu, L. T. Guan, New descent nonlinear conjugate gradient methods for large-scale unconstrained optimization, Department of Scientific Computation and Computer, 2005.
34. G. Yu, L. Guan, W. Chen, Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization, *Optim. Method. Softw.*, **23** (2008), 275–293. <https://doi.org/10.1080/10556780701661344>
35. A. Perry, A modified conjugate gradient algorithm, *Oper. Res.*, **26** (1978), 1073–1078. <https://doi.org/10.1287/opre.26.6.1073>
36. W. Sun, Y. Yuan, *Optimization theory and methods: Nonlinear programming*, New York: Springer, 2006. <https://doi.org/10.1007/b106451>
37. M. Y. Waziri, K. Ahmad, A. S. Halilu, A. M. Awwal, A modified Dai Yuan iterative Scheme for nonlinear systems and its application, *Numer. Algebr. Control*, **13** (2023), 53–80. <https://doi.org/10.3934/naco.2021044>
38. N. Zhang, J. K. Liu, B. Tang, A three-term projection method based on spectral secantequation for nonlinear monotone equations, *Japan J. Indust. Appl. Math.*, **41** (2024), 617–635. <https://doi.org/10.1007/s13160-023-00624-4>
39. A. I. Kiri, M. Y. Waziri, A. S. Halilu, Modification of the double direction approach for solving systems of nonlinear equations with application to Chandrasekhar's Integral equation, *Iranian J. Numer. Anal. Optim.*, **12** (2022), 426–448. <https://doi.org/10.22067/IJNAO.2022.74201.1083>
40. A. M. Awwal, P. Kumam, H. Mohammad, W. Watthayu, A. B. Abubakar, A Perry-type derivative-free algorithm for solving nonlinear system of equations and minimizing ℓ_1 regularized problem, *Optimization*, **70** (2021), 1231–1259. <https://doi.org/10.1080/02331934.2020.1808647>
41. K. Ahmed, M. Y. Waziri, A. S. Halilu, S. Murtala, H. Abdullahi, Signal and image reconstruction with a double parameter Hager-Zhang-type conjugate gradient method for system of nonlinear equations, **32** (2025), E2583. <https://doi.org/10.1002/nla.2583>
42. E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Math. program. Ser. A*, **91** (2002), 201–213. <https://doi.org/10.48550/arXiv.cs/0102001>
43. M. R. Banham, A. K. Katsaggelos, Digital image restoration, *IEEE Signal Proc. Mag.*, **14** (1997), 24–41. <https://doi.org/10.1109/79.581363>

44. C. L. Chan, A. K. Katsaggelos, A. V. Sahakian, Image sequence filtering in quantum-limited noise with applications to low-dose fluoroscopy, *IEEE T. Med. Imaging*, **12** (1993), 610–621. <https://doi.org/10.1109/42.241890>
45. 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-STSP*, **1** (2007), 586–597. <https://doi.org/10.1109/JSTSP.2007.910281>
46. J. S. Pang, Inexact Newton methods for the nonlinear complementarity problem, *Math. Program.*, **36** (1986), 54–71. <https://doi.org/10.1007/BF02591989>
47. Y. Xiao, Q. Wang, Q. Hu, Non-smooth equations based method for ℓ_1 -norm problems with applications to compressed sensing, *Nonlinear Anal Theor.*, **74** (2011), 3570–3577. <https://doi.org/10.1016/j.na.2011.02.040>



AIMS Press

© 2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)