RAIRO-Oper. Res. 57 (2023) 2561–2584
https://doi.org/10.1051/ro/2023099

RAIRO Operations Research
www.rairo-ro.org

GLOBAL CONVERGENCE *VIA* MODIFIED SELF-ADAPTIVE APPROACH FOR SOLVING CONSTRAINED MONOTONE NONLINEAR EQUATIONS WITH APPLICATION TO SIGNAL RECOVERY PROBLEMS

Muhammad Abdullahi[1,2,3,*], Auwal Bala Abubakar[3,4] and Sadiq Bashir Salihu[1,5]

Abstract. The conjugate gradient method (CG) is one of the most rapidly expanding and efficient ways of solving unconstrained minimization problems. Recently, there has been a lot of effort put into extending the CG approach to solve monotone nonlinear equations. In this paper, we describe a variation of the CG method for solving constrained monotone nonlinear equations. The approach has a sufficient descent property, and its global convergence has been demonstrated with the help of some reasonable assumptions. Two sets of numerical tests were run to demonstrate the proposed method's superior performance when compared to other methods. The initial experiment aimed to solve nonlinear equations with constraints, while in the second experiment, the method was applied to sparse signal reconstruction.

Mathematics Subject Classification. 65K05, 90C52, 90C56, 94A08.

Received August 13, 2022. Accepted July 4, 2023.

1. Introduction

Many scientists are interested in nonlinear problems in some way because of the wide range of applications that arise from disciplines of science, engineering, and other social science that are intrinsically nonlinear. Below is a nonlinear system of equations:

$$G(h) = 0, \quad h \in \Omega, \tag{1}$$

where $G : \mathbb{R}^n \to \mathbb{R}^n$ is nonlinear map. Additionally, the set $\Omega \subseteq \mathbb{R}^n$ is nonempty, closed, and convex. From beginning to the end of this paper, the space $\mathbb{R}^n$ denote the n-dimensional real space equipped with the Euclidean norm $\|\cdot\|$, $G_k = G(h_k)$.

Keywords. Non-linear equations, signal processing, projection map, convex constraint, global convergence.

[1] School of Mathematics and Statistics, HNP-LAMA, Central South University, Changsha, Hunan 410083, P.R. China.

[2] Department of Mathematics, Faculty of Natural and Applied Sciences, Sule Lamido University, Kafin-Hausa, Nigeria.

[3] Numerical Optimization Research Group, Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano, Kano, Nigeria.

[4] Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, Medunsa 0204, South Africa.

[5] Department of Mathematics, School of Science, Jigawa State College of Education, Gumel, Nigeria.

*Corresponding author: muhammad.abdullahi@slu.edu.ng

Definition 1.1. Let $h, v \in \mathbb{R}^n$, a mapping $G : \mathbb{R}^n \to \mathbb{R}^n$ is said to be monotone if for all $h, v \in \mathbb{R}^n$,

$$\langle G(h) - G(v),\ h - v\rangle \geq 0. \tag{2}$$

If G satisfies inequality (2), then (1) is called the monotone system of nonlinear equations. However, the monotone mappings, which form a class of nonlinear equations were initially investigated in [29, 36, 46], in Hilbert spaces. Many scientific areas, such as economic equilibrium problems [48], chemical equilibrium systems [35], power flow equation [42], ℓ_1-norm regularization problem in signal, image recovery [23,44] and many more, have revealed the practical application of studying such problems. For more details on these approaches, see [1,7,8,14,22,25–28,30] and the references therein. The iterative process for these methods is given by

$$h_{k+1} = h_k + \alpha_k d_k, \quad k = 0, 1, \ldots, \tag{3}$$

where, $s_k = h_{k+1} - h_k$ and the step size is denoted by α_k, and it is computed using appropriate line search. A line search is a process of computing step size α_k in line with the search direction d_k *via* (3) to generate the iterative sequence $\{h_k\}$ so as to achieve global convergence.

Whenever G is the gradient mapping of some function $f : \mathbb{R}^n \to \mathbb{R}$, problem (1) is equivalent to the first order necessary condition for the unconstrained optimization problem

$$\min f(h), \quad h \in \mathbb{R}^n, \tag{4}$$

where, $f : \mathbb{R}^n \to \mathbb{R}$ is suppose to be twice continuously differentiable [32].

Among the algorithms for solving (1), Newton's method is the most famous, due to its rapid convergence properties [15]. For instance good convergence rate from a satisfactorily initial point $h_0 \in \mathbb{R}^n$ in the proximity of the solution. However, Newton's direction d_k must solve the linear system of equations given below,

$$G_k = -G'_k d_k,$$

where, the jacobian matrix G'_k of G at $\{h_k\}$ is given. However, in Newton's method, the derivative G' at each iteration may be difficult to obtain or doesn't exist at all. This together with the huge storage requirement of the scheme, makes it not appropriate for solving (1). Quasi-Newton's methods were presented for the aforementioned reason to substitute the Jacobian matrix or its inverse with an approximation which can be modified at each iteration [14, 45], where

$$d_k = -B_k^{-1} G_k,$$

is the search direction and B_k is the $n \times n$ matrix which approximate the Jacobian of G at $\{h_k\}$. Moreover, the distinguished class of quasi-Newton update B_k needs to satisfy the secant equation

$$B_k s_{k-1} = v_{k-1},$$

where, $v_{k-1} = G_k - G(h_{k-1})$ and $s_{k-1} = h_k - h_{k-1}$. The derivative G' or its approximation is computed at each iteration, hence Newton and quasi-Newton methods are not suitable for solving large-scale problems, despite their appealing properties. To overcome this, the authors in [1] developed a matrix-free approach based on a quasi-Newton, in which the approximation to Broyden's update is achieved by generating a diagonal matrix using the acceleration parameter,

$$B_k \approx \lambda_k I,$$

where, $\lambda_k \in \mathbb{R}^n$ and I is the identity matrix. This made the method a good avenue to solve large-scale nonlinear problem.

CG methods are type of matrix-free method that uses the iterative procedure (3) to generate the next iteration, with the search direction specified as

$$d_k = \begin{cases} -G_k, & \text{if } k = 0; \\ -G_k + \beta_k d_{k-1}, & \text{otherwise,} \end{cases} \tag{5}$$

where, $G_k = \nabla f(h_k)$, and β_k is a scalar depicting the CG methods' features. It is divided into two categories based on the global convergence characteristics and numerical efficiency of CG algorithms [10]. The first category includes the numerically efficient Hestenes and Stiefel (HS) method [24], the Polak–Ribiere–Polyak (PRP) method [38], and the Liu and Storey (LS) method [34]. However, due to their poor convergence qualities, these approaches may fail to converge for general functions. The second category consists of the methods developed by Fletcher and Reeves (FR) [20], Fletcher, conjugate descent (CD) [19], and Dai and Yuan (DY) [13] have a good global convergence but are not numerically efficient.

For these reasons much effort has been put to develop efficient CG methods, that facilitate many applications in sciences and engineering, which have good numerical efficiency and strong global convergence (see [5,6,10,21]). Due to its nontrivial global convergence task, the Spectral Gradient (SG) method is one among the updated versions of the Bazilai and Bowein method that is not descent. To address this issue, Huang and Wan presented a Spectral residual approach for non-linear and non-smooth equations in [47]. Solodov and Svaiter [39], proposed a hybrid projection method for solving unconstrained monotone problems. This was accomplished by establishing an adequate separation hyperplane in the Newton direction. The method is sufficiently descending and globally convergent, but it is not suitable for solving smooth equations because of its reliance on gradient information.

Finally, Abubakar *et al.* [3] proposed modification of the Fletcher–Reeves (FR) CG method to solve nonlinear monotone equations with convex constraints. The modification of the algorithm gives automatic assurance that the search direction is descent, gives more improvement in its numerical illustrations, and maintains the rapid convergence properties. The numerical analysis displays the accuracy of the method.

In this research, motivated by Abubakar *et al.* [3] we attempt to propose a modification of the self-adaptive approach in [41] for solving problem (1). This is to enhance numerical efficiency, making it globally convergent, and broadening the field of its application to signal recovery. The method is based on the modification which ensures the search direction is automatically descent, and improves its numerical efficiency. Furthermore, the global convergence of the proposed algorithm is established under mild assumptions.

The organization of the paper is as follows. The proposed algorithm is presented in the second part, and the convergence analysis of the proposed algorithm is recorded in part 3. Part 4 lists some numerical experiments and the application of the proposed method in sparse signal reconstruction. Part 5 is the article's conclusion.

2. Preliminaries and proposed algorithm

This section gives some basic concepts and properties of projection mapping as well as some assumptions.

Definition 2.1. Let $\Omega \subseteq \mathbb{R}^n$ be a nonempty closed convex set. Then for any $h \in \mathbb{R}^n$, its orthogonal projection onto Ω, denoted by $P_\Omega(h)$, is defined by

$$P_\Omega(h) := \arg\min\{\|h - v\| : v \in \Omega\}. \tag{6}$$

It has the property that $\|P_\Omega(h) - P_\Omega(v)\| \leq \|h - v\| \ \forall\ h, v \in \mathbb{R}^n$. This property is called non-expansive.

Throughout this paper, we suppose the following:

Assumption 2.2. *The solution set* (1) *denoted by* Ω *is nonempty.*

Assumption 2.3. *The operator* G *is monotone, that is* $\forall\ h, v \in \mathbb{R}^n$,

$$(G(h) - G(v))^T (h - v) \geq 0. \tag{7}$$

Assumption 2.4. *The operator* G *is Lipschitz continuous on* $\mathbb{R}^n$. *That is* $\forall\ h, v \in \mathbb{R}^n$, *there exists a Lipschitz constant* $L > 0$, *such that*

$$\|G(h) - G(v)\| \leq L\|h - v\|. \tag{8}$$

Equation (3) is usually used in numerical methods in such a way that the direction d_k satisfy

$$G(h_k)^T d_k \leq -\lambda \|G(h_k)\|^2, \tag{9}$$

where $\lambda > 0$ is a positive constant. The condition (9) means the search direction d_k is descent if G_k is the gradient vector of some real-valued function $f : \mathbb{R}^n \to \mathbb{R}$.

Abubakar *et al.* [2]. Proposed the following direction,

$$d_k = \begin{cases} -G(h_k), & \text{if } k = 0; \\ -G(h_k) + \beta'_k d_{k-1} + \theta'_k w_{k-1}, & \text{otherwise}, \end{cases} \tag{10}$$

where,

$$\beta'_k = \frac{G(h_k)^T w_{k-1}}{d_{k-1}^T w_{k-1}}, \ \theta'_k = \frac{G(h_k)^T d_{k-1}}{d_{k-1}^T w_{k-1}}, \ v_{k-1} = G(h_k) - G(h_{k-1}) + r s_{k-1}, \ t_{k-1} = 1 + \max\left\{0, \frac{-d_{k-1}^T v_{k-1}}{d_{k-1}^T d_{k-1}}\right\},$$
$$w_{k-1} = v_{k-1} + t_{k-1} d_{k-1}, \quad s_{k-1} = h_k - h_{k-1}.$$

Using same approach in [3], we propose a modification of the (10) as follows

$$d_k = \begin{cases} -G(h_k), & \text{if } k = 0; \\ -G(h_k) + \frac{G(h_k)^T w_{k-1} d_{k-1} - G(h_k)^T d_{k-1} w_{k-1}}{\max\left\{\mu \|w_{k-1}\| \|d_{k-1}\|, \ d_{k-1}^T w_{k-1}\right\}}, & \text{otherwise}, \end{cases} \tag{11}$$

where $\mu > 0$ is a positive parameter. The difference between the two directions is the scaling term in the denominator $d_{k-1}^T w_{k-1}$ of the parameters in (10) which is adjusted to $\max\{\mu \|w_{k-1}\| \|d_{k-1}\|, \ d_{k-1}^T w_{k-1}\}$ in (11). Under Assumption 2.2–2.4, we propose an algorithm that generates approximate solutions to (1).

Algorithm 1: A global convergence *via* modified self-adaptive conjugate gradient (GMSCG).

Input. Given $h_0 \in \mathbb{R}^n$, $\rho \in (0,1)$, $\mu, r \geq 0$, $\sigma > 0$, $0 < \gamma < 2$, $tol > 0$. Set $k := 0$.
Step 1. If $\|G_k\| \leq tol$, discontinue. Else go to **Step 2**.
Step 2. Calculate d_k by (11)
Step 3. Set $z_k = h_k + \alpha_k d_k$, and Compute the step length $\alpha_k = \rho^i$, for $i = 0, 1, \ldots$, where i is the smallest positive integer satisfying

$$-G(z_k)^T d_k \geq \sigma \alpha_k \|d_k\|^2. \tag{12}$$

Step 4. Compute

$$h_{k+1} = P_\Omega[h_k - \gamma \eta_k G(z_k)], \tag{13}$$

where

$$\eta_k = \frac{G(z_k)^T (h_k - z_k)}{\|G(z_k)\|^2}, \ G(z_k) \neq 0. \tag{14}$$

Step 5. Let $k = k + 1$ and restart from **Step 1**.

Remark 2.5. From (11), if $k = 0$,

$$G(h_k)^T d_k = G(h_0)^T G(h_0) = -\|G(h_0)\|^2.$$

If $k > 0$,

$$G(h_k)^T d_k = -\|G(h_k)\|^2 + \frac{\left(G(h_k)^T w_{k-1} d_{k-1}\right)^T G(h_k) - \left(G(h_k)^T d_{k-1} w_{k-1}\right)^T G(h_k)}{\max\left\{\mu \|w_{k-1}\| \|d_{k-1}\|, \ d_{k-1}^T w_{k-1}\right\}}$$

$$
\begin{aligned}
&= -\|G(h_k)\|^2 + \frac{\left(G(h_k)^T w_{k-1}\right)\left(d_{k-1}^T G(h_k)\right) - \left(G(h_k)^T d_{k-1}\right)\left(w_{k-1}^T G(h_k)\right)}{\max\left\{\mu\|w_{k-1}\|\|d_{k-1}\|,\ d_{k-1}^T w_{k-1}\right\}} \\
&= -\|G(h_k)\|^2 + \frac{\left(G(h_k)^T w_{k-1}\right)\left(d_{k-1}^T G(h_k)\right) - \left(G(h_k)^T w_{k-1}\right)\left(d_{k-1}^T G(h_k)\right)}{\max\left\{\mu\|w_{k-1}\|\|d_{k-1}\|,\ d_{k-1}^T w_{k-1}\right\}} \\
&= -\|G(h_k)\|^2. \qquad (15)
\end{aligned}
$$

Using Cauchy–Schwartz inequality, we get

$$\|G(h_k)\| \leq \|d_k\|. \qquad (16)$$

Additionally, since $\max\{\mu\|w_{k-1}\|\|d_{k-1}\|,\ d_{k-1}^T w_{k-1}\} \geq \mu\|w_{k-1}\|\|d_{k-1}\|$, then,

$$
\begin{aligned}
\|d_k\| &= \left\| -G(h_k) + \frac{\left(G(h_k)^T w_{k-1} d_{k-1}\right)^T G(h_k) - \left(G(h_k)^T d_{k-1} w_{k-1}\right)^T G(h_k)}{\max\left\{\mu\|w_{k-1}\|\|d_{k-1}\|,\ d_{k-1}^T w_{k-1}\right\}} \right\| \\
&\leq \|G(h_k)\| + \frac{\left\|G(h_k)^T w_{k-1} d_{k-1} - G(h_k)^T d_{k-1} w_{k-1}\right\|}{\max\left\{\mu\|w_{k-1}\|\|d_{k-1}\|,\ d_{k-1}^T w_{k-1}\right\}} \\
&\leq \|G(h_k)\| + \frac{\left\|G(h_k)^T w_{k-1} d_{k-1}\right\|}{\mu\|w_{k-1}\|\|d_{k-1}\|} + \frac{\left\|G(h_k)^T d_{k-1} w_{k-1}\right\|}{\mu\|w_{k-1}\|\|d_{k-1}\|} \\
&\leq \|G(h_k)\| + \frac{\|G(h_k)\|\|w_{k-1}\|\|d_{k-1}\|}{\mu\|w_{k-1}\|\|d_{k-1}\|} + \frac{\|(G(h_k)\|\|d_{k-1}\|\|w_{k-1})\|}{\mu\|w_{k-1}\|\|d_{k-1}\|} \\
&= \|G(h_k)\| + 2\frac{\|G(h_k)\|}{\mu} \\
&= \left(1 + \frac{2}{\mu}\right)\|G(h_k)\|. \qquad (17)
\end{aligned}
$$

Combining (16) and (17), we have

$$\|G_k\| \leq \|d_k\| \leq \left(1 + \frac{2}{\mu}\right)\|G(h_k)\|. \qquad (18)$$

3. Convergence result

The following lemmas and theorem are required to prove the global convergence of Algorithm 1.

Lemma 3.1. *Let Assumptions 2.2–2.4 be satisfied, and $\{h_k\}$ be generated by GMSCG algorithm, then the line search given by (12) is well defined.*

Proof. Let's assume that there exists $k_0 \geq 0$, such that given any positive integer i, we have

$$-G\left(h_{k_0} + \rho^i d_{k_0}\right)^T d_{k_0} < \sigma\rho^i\|d_{k_0}\|. \qquad (19)$$

Since $\rho \in (0,1)$, using Assumption 2.4 and allowing $i \to \infty$, we get

$$-G(h_{k_0})^T d_{k_0} \leq 0. \qquad (20)$$

From (15), we have

$$-G(h_{k_0})^T d_{k_0} = \|G(h_k)\|^2 > 0.$$

which clearly contradicts (20). Hence, the line search is well defined. □

Lemma 3.2. *Let the Assumption 2.4 be satisfied and* $\{h_k\}$ *be generated by GMSCG algorithm, then*

$$\alpha_k \geq \rho \min\left\{\frac{1}{(L+\sigma)\left(1+\frac{2}{\mu}\right)^2}\right\}. \tag{21}$$

Proof. If $\alpha_k \neq 1$, then $\alpha_k' = \frac{\alpha_k}{\rho}$ does not guarantee (12), that is

$$-G(h_k + \alpha_k' d_k)^T d_k < \sigma \alpha_k' \|d_k\|^2. \tag{22}$$

Using (15), Cauchy–Schwartz, Assumption 2.4, we have

$$\begin{aligned}\|G(h_k)\|^2 &\leq -G(h_k)^T d_k \\ &= (G(h_k + \alpha_k' d_k) - G(h_k))^T d_k - G(h_k + \alpha_k' d_k)^T d_k < L\alpha_k' \|d_k\|^2 + \sigma \alpha_k' \|d_k\|^2 \\ &= (L+\sigma)\alpha_k \rho^{-1} \|d_k\|^2.\end{aligned}$$

Combining the above inequality with (17), we get

$$\begin{aligned}\alpha_k &> \frac{\rho \|G(h_k)\|^2}{(L+\sigma)\|d_k\|^2} \\ &\geq \rho \min\left\{\frac{1}{(L+\sigma)\left(1+\frac{2}{\mu}\right)^2}\right\}.\end{aligned} \tag{23}$$

□

Lemma 3.3. *Let Assumption 2.4 be satisfied, the sequence* $\{h_k\}$ *and* $\{z_k\}$ *are generated by GMSCG algorithm. For any* $\bar{h}$ *such that* $G(\bar{h}) = 0$*, it holds that.*

$$\left\|h_k - \bar{h}\right\|^2 - \gamma(2-\gamma)\sigma^2 \frac{\|h_k - z_k\|^4}{\|G(z_k)\|^2}. \tag{24}$$

In particular, $\{h_k\}$ *is bounded and*

$$\lim_{k\to\infty} \|h_k - z_k\| = 0. \tag{25}$$

Proof. We prove that $\{h_k\}$ and $\{z_k\}$ are bounded. Let $\bar{h} \in \Omega$ then by monotonicity of G, we get

$$\langle G(z_k), h_k - \bar{h}\rangle \geq \langle G(z_k), h_k - z_k\rangle. \tag{26}$$

By (13), (14), $\gamma \in (0,2)$ and the nonexpensive of the projection map,

$$\begin{aligned}\left\|h_{k+1} - \bar{h}\right\|^2 &= \left\|P_\Omega[h_k - \gamma\eta_k G(z_k)] - P_\Omega(\bar{h})\right\|^2 \\ &\leq \left\|h_k - \gamma\eta_k G(z_k) - \bar{h}\right\|^2 \\ &\leq \left\|h_k - \bar{h}\right\|^2 - 2\gamma\eta_k\langle G(z_k), h_k - \bar{h}\rangle + \|\gamma\eta_k G(z_k)\|^2 \\ &= \left\|h_k - \bar{h}\right\|^2 - 2\gamma\frac{\langle G(z_k), h_k - z_k\rangle}{\|G(z_k)\|^2}\langle G(z_k), h_k - \bar{h_k}\rangle + \gamma^2\left(\frac{\langle G(z_k), h_k - z_k\rangle}{\|G(z_k)\|}\right)^2 \\ &\leq \left\|h_k - \bar{h}\right\|^2 - 2\gamma\left(\frac{\langle G(z_k), h_k - z_k\rangle}{\|G(z_k)\|}\right)^2 + \gamma^2\left(\frac{\langle G(z_k), h_k - z_k\rangle}{\|G(z_k)\|}\right)^2\end{aligned}$$

$$= \|h_k - \bar{h}\|^2 - \gamma(2-\gamma)\left(\frac{\langle G(z_k), h_k - z_k\rangle}{\|G(z_k)\|}\right)^2$$
$$\leq \|h_k - \bar{h}\|^2 - \gamma(2-\gamma)\sigma^2 \frac{\|h_k - z_k\|^4}{\|G(z_k)\|^2}. \tag{27}$$

This prove that the sequence $\{\|h_k - \bar{h}\|\}$ is non increasing and convergent, thus $\{h_k\}$ is bounded and

$$\|h_{k+1} - \bar{h}\|^2 \leq \|h_k - \bar{h}\|^2,$$

which recursively implies

$$\|h_k - \bar{h}\|^2 \leq \|h_0 - \bar{h}\|^2, \qquad \forall\ k \geq 0. \tag{28}$$

Thus from Assumption 2.4, and $\bar{h} \in \Omega$ we get

$$\|G(h_k)\| = \|G(h_k) - G(\bar{h})\| \leq L\|h_k - \bar{h}\| \leq L\|h_0 - \bar{h}\|. \tag{29}$$

Letting $L\|h_0 - \bar{h}\| = \xi$, then $\{G(h_k)\}$ is bounded, that is

$$\|G(h_k)\| \leq \xi, \qquad \forall\ k \geq 0. \tag{30}$$

By the definition of $\{z_k\}$, (12), monotonicity of G and the Cauchy–Schwartz inequality we obtain

$$\sigma\|h_k - z_k\| = \frac{\sigma\|\alpha_k d_k\|^2}{\|h_k - z_k\|} \leq \frac{\langle G(z_k), h_k - z_k\rangle}{\|h_k - z_k\|} \leq \frac{\langle G(z_k), h_k - z_k\rangle}{\|h_k - z_k\|} \leq \|G(h_k)\|. \tag{31}$$

The boundedness of $\{h_k\}$ together with (30) and (31), implies the sequence $\{z_k\}$ is bounded.

Since $\{z_k\}$ is bounded, then for any $\bar{h} \in \Omega$, the sequence $\|z_k - \bar{h}\|$ is also bounded *i.e.* There exist a $h > 0$ such that

$$\|G(z_k)\| = \|G(z_k) - G(\bar{h})\| \leq L\|z_k - \bar{h}\| \leq Lh. \tag{32}$$

Using equation (27), we obtain

$$\gamma(2-\gamma)\frac{\sigma^2}{(Lh)^2}\|h_k - z_k\|^4 \leq \|h_k - \bar{h}\|^2 - \|h_{k+1} - \bar{h}\|^2,$$

which implies

$$\gamma(2-\gamma)\frac{\sigma^2}{(Lh)^2}\sum_{k=0}^{\infty}\|h_k - z_k\|^4 \leq \sum_{k=0}^{\infty}\left(\|h_k - \bar{h}\|^2 - \|h_{k+1} - \bar{h}\|^2\right) \leq \|h_0 - \bar{h}\|^2 < \infty. \tag{33}$$

Thus, we conclude from (33) that

$$\lim_{k\to\infty}\|h_k - z_k\| = \lim_{k\to\infty}\alpha_k\|d_k\| = 0. \tag{34}$$

□

Theorem 3.4. *Let Assumption 2.2–2.4 be satisfied. Then $\{h_k\}$ is generated by (13) in GMSCG algorithm, then $\{h_k\}$ converges to a solution of (1).*

Proof. Note that, from (21) and (34), we have

$$0 \leq \alpha\|d_k\| \leq \alpha_k\|d_k\| \to 0.$$

This implies that $\lim_{k\to\infty}\|d_k\| = 0$. This together with (9) yield

$$0 \leq -\alpha_k\|G(h_k)\| \leq \|d_k\| \to 0,$$

which implies $\lim_{k\to\infty}\|G(h_k)\| = 0$. Since the sequence $\{h_k\} \subset \Omega$ is bounded by (27). Thus we can find at least one limit point of $\{h_k\}$. Let $\{h^*\}$ be a limit point of the sequence $\{h_k\} \subset \Omega$ and let $\mathcal{K} \subset \{0, 1, 2, \ldots\}$ be an infinite indexing set for which

$$\lim_{k\to\infty,\ k\in\mathcal{K}} h_k = h^* \in \Omega,$$

since Ω is closed by Assumption 2.2. Thus, by Assumption 2.4, we have

$$0 = \lim_{k\to\infty}\|G(h_k)\| = \lim_{k\to\infty,\ k\in\mathcal{K}}\|G(h_k)\| = \|G(h^*)\|,$$

which implies h^* is a solution of (1). If we set $\bar{h} := h^*$ in Lemma 3.3 and the fact that $\{\|h_k - h^*\|\}$ converges, then

$$\lim_{k\to\infty}\|h_k - h^*\| = \lim_{k\to\infty,\ k\in\mathcal{K}}\|h_k - h^*\| = 0.$$

Therefore, we can conclude that the sequence $\{h_k\}$ converges to h^*. □

4. Numerical results

We assess the numerical potentiality of the proposed algorithm called GMSCG with the approach proposed by Abubakar *et al.* [2] called MSCG, and another approach by Abubakar *et al.* [4] called MCG. The number of iterations (ITER), number of function evaluations (FVAL), and time (TIME) taken to find the approximate solution are used to evaluate each algorithm. The algorithm with the least ITER, FVAL, and TIME in most of the problems is considered the most efficient or the best solver. The parameters we used for the implementation of GMSCG algorithm are: $\kappa = 1$, $\rho = 0.46$, $\sigma = 0.000001, \gamma = 1.22$, $\mu = 0.56$, $r = 0.000002$. As for MSCG and MCG, all parameters are selected as given in [2, 4], respectively. Five different dimensions of 1000, 5000, 10 000, 50 000, 100 000, and ten different initial points $h_1 = (1, \cdots, 1)^T$, $h_2 = \left(\frac{1}{2}, \cdots, \frac{1}{2^n}\right)^T$, $h_3 = \left(1 - \frac{1}{n}, 1 - \frac{2}{n}, \cdots, 0\right)^T$, $h_4 = \left(1, \frac{1}{2}, \cdots, \frac{1}{n}\right)^T$, $h_5 = (2, \cdots, 2)^T$, $h_6 = \left(\frac{-1}{4}, \frac{-1}{8}, \cdots, \frac{-1}{4n}\right)^T$, $h_7 = (0.6, \cdots, 0.6)^T$, $h_8 = (-0.5, \cdots, -0.5)^T$, $h_9 = (0.4, \cdots, 0.4)^T$, and $h_{10} = (0.1, \cdots, 0.1)^T$ were used. Matlab 8.3.0 (R2014a) on a PC with an intel Celeron (R) processor, 2 GB of RAM and CPU 1.60 GHz × 8 is used for the experiments. The stopping condition is when $\|G_k\| \leq 10^{-6}$ or number of iterations exceeds 1000. We perform the experiments on the following eight problems, where $G = (G_1, G_2, \ldots, G_n)^T$.

Example 1 ([9])**.**

$$\begin{aligned} G_1(h) &= e^{h_1} - 1, \\ G_i(h) &= e^{h_i} + h_i - 1, \qquad i = 2, 3, \ldots, n-1, \end{aligned}$$

where $\Omega = \mathbb{R}^n_+$.

Example 2 ([9])**.**

$$G_i(h) = h_i - \sin|h_i - 1|, \qquad i = 1, 2, \ldots, n,$$

where $\Omega = \mathbb{R}^n_+$.

Clearly, Problem 2 is nonsmooth at $h = 0$.

Example 3 ([9]).

$$G_i(h) = 2h_i - \sin|h_i|, \qquad i = 1, 2, \ldots, n,$$

where $\Omega = \mathbb{R}^n_+$.

Clearly, Problem 3 is nonsmooth at $h = 0$.

Example 4 ([9]).

$$G_i(h) = \min\min\left|h_i, h_i^2\right|, \ \max\left|h_i, h_i^3\right|, \qquad i = 1, 2, \ldots, n,$$

where $\Omega = \mathbb{R}^n_+$.

Example 5 ([21]).

$$G_i(h) = e^{h_i} - 1, \qquad i = 1, 2, \ldots, n,$$

where $\Omega = \mathbb{R}^n_+$.

Example 6 ([9]).

$$\begin{aligned} G_1(h) &= 2h_1 - h_2 + e^{h_1} - 1, \\ G_i(h) &= -h_{i-1} + 2h_i - h_{i+1} + e^{h_i} - 1, \qquad i = 2, 3, \ldots, n-1, \\ G_n(h) &= -h_{n-1} - 2h_n + e^{h_n} - 1, \end{aligned}$$

where $h = \frac{1}{n+1}$ and $\Omega = \mathbb{R}^n_+$.

Example 7. Tridiagonal exponential function [31]

$$\begin{aligned} G_1(h) &= h_1 - e^{\cos(h(h_1+h_2))} \\ G_i(h) &= h_i - e^{\cos(h(h_{i-1}+h_i+h_{i+1}))}, \qquad i = 2, \ldots, n-1, \\ G_n(h) &= h_n - e^{\cos(h(h=s_{n-1}+x_n))}, \\ h &= \frac{1}{n+1} \quad \text{and} \quad C = \mathbb{R}^n_+. \end{aligned}$$

Example 8 ([21]).

$$G_i(h) = e^{h_i^2} + 1.5\sin 2h_i - 1, \qquad i = 1, 2, \ldots, n,$$

where $\Omega = \mathbb{R}^n_+$.

Details of the numerical experiments are provided in Tables 1–8 with (ITER) being the number of iterations, (FVAL) the number of function evaluations, (TIME) the CPU time, and (NORM) the function at the approximate solution.

Figures 1–3 are plotted based on Dolan and Moré [16] performance profiles. The figures are used to compare performance of `GMSCG` against the `MSCG` and `MCG` with respect to ITER, FVAL, and TIME, respectively.

From Figures 1–3, one can see that `GMSCG` is the best solver as it is the most efficient and promising. The performance of `GMSCG` with respect to the number of iterations in Figure 1 is outstanding as it is successful in over 69% of the examples with lesser number of iterations compared to the `MSCG` and `MCG`. In addition, Figures 2 and 3 reveal that `GMSCG` is more efficient, since it is successful in almost 72% of the examples with lesser number of function evaluations, and over 47% of the examples with less CPU time compared to the `MSCG` and `MCG`.

TABLE 1. Numerical results for GMSCG, MSCG and MCG on Example 1.

		GMSCG				MSCG				MCG			
Dimension	IP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	h_1	5	13	0.12208	5.97E−06	11	9	0.030585	0	1	3	0.22255	0
	h_2	8	16	0.12154	5.99E−15	17	39	0.11743	2.42E−06	1001	14488	22.6787	0.000999
	h_3	2	5	0.038569	0	3	9	0.035819	0	1	3	0.052468	0
	h_4	4	9	0.053239	6.28E−07	4	12	0.033473	1.07E−06	1001	20907	40.6797	0.000169
1000	h_5	1	3	0.02366	0	4	4	0.042723	0	1	4	0.033746	0
	h_6	2	4	0.038813	0	2	6	0.023553	0	2	6	0.03638	0
	h_7	2	5	0.040511	0	3	9	0.028299	0	1	3	0.047053	0
	h_8	3	6	0.048192	0	2	6	0.042489	0	2	6	0.054136	0
	h_9	2	5	0.041995	0	5	9	0.022796	0	1	3	0.031102	0
	h_{10}	1	2	0.034278	0	1	9	0.011845	0	1	3	0.038312	0
	h_1	5	13	0.28379	5.91E−06	9	9	0.043211	0	1	3	0.10678	0
	h_2	8	16	0.22288	5.99E−15	17	39	0.52432	2.42E−06	1001	14488	22.8166	0.000999
	h_3	2	5	0.095226	0	3	9	0.090758	0	1	3	0.060115	0
	h_4	4	9	0.15849	6.27E−07	4	12	0.25691	1.06E−06	1001	20908	33.1927	0.000171
5000	h_5	1	3	0.065546	0	3	4	0.059843	0	1	4	0.066181	0
	h_6	2	4	0.075348	0	2	9	0.045275	0	2	6	0.10074	0
	h_7	2	5	0.23105	0	4	3	0.044599	0	1	3	0.063058	0
	h_8	2	4	0.24488	0	2	6	0.13705	0	2	6	0.098452	0
	h_9	2	5	0.094828	0	6	9	0.045318	0	1	3	0.069361	0
	h_{10}	1	2	0.21162	0	5	8	0.073074	0	1	3	0.0853	0
	h_1	5	13	0.30571	5.90E−06	9	9	0.084001	0	1	3	0.084172	0
	h_2	8	16	0.36924	5.99E−15	17	39	0.31921	2.42E−06	1001	14488	35.9684	0.000999
	h_3	2	5	0.098514	0	4	9	0.051662	0	1	3	0.16297	0
	h_4	4	9	0.29653	6.27E−07	4	12	0.093933	1.06E−06	1001	20908	57.7487	0.000171
10 000	h_5	1	3	0.10293	0	7	4	0.13344	0	1	4	0.13829	0
	h_6	2	4	0.21557	0	8	6	0.09044	0	2	6	0.27731	0
	h_7	2	5	0.17069	0	9	9	0.041454	0	1	3	0.08547	0
	h_8	2	4	0.14143	0	2	6	0.064003	0	2	6	0.2011	0
	h_9	2	5	0.14757	0	5	8	0.031618	0	1	3	0.13353	0
	h_{10}	1	2	0.072983	0	9	9	0.10529	0	1	3	0.14901	0
	h_1	5	13	1.1471	5.89E−06	7	9	0.34149	0	1	3	0.32008	0
	h_2	8	16	1.2383	5.99E−15	17	39	2.2883	2.42E−06	1001	14488	149.2373	0.000999
	h_3	2	5	0.71066	0	3	9	0.32432	0	1	3	0.37267	0
	h_4	4	9	0.78939	6.27E−07	4	12	0.89415	1.06E−06	1001	20908	257.8186	0.000171
50 000	h_5	1	3	0.34156	0	6	4	0.35518	0	1	4	0.40734	0
	h_6	2	4	0.5329	0	2	6	0.55713	0	2	6	0.57388	0
	h_7	2	5	0.50614	0	6	8	0.30654	0	1	3	0.38661	0
	h_8	2	4	0.56958	0	2	6	0.5928	0	2	6	0.53802	0
	h_9	2	5	0.52068	0	4	9	0.35953	0	1	3	0.37026	0
	h_{10}	1	2	0.25112	0	8	9	0.33162	0	1	3	0.35342	0
	h_1	5	13	2.0395	5.89E−06	9	9	0.78033	0	1	3	0.76679	0
	h_2	8	16	1.151	5.99E−15	17	39	2.9209	2.42E−06	1001	14488	391.8248	0.000999
	h_3	2	5	1.0054	0	4	9	0.66662	0	1	3	0.6775	0
	h_4	4	9	1.4355	6.27E−07	4	12	1.665	1.06E−06	1001	20908	1341.953	0.000171
100 000	h_5	1	3	0.78687	0	8	4	0.772	0	1	4	0.81231	0
	h_6	2	4	0.81683	0	2	6	1.09	0	2	6	0.95186	0
	h_7	2	5	0.85388	0	4	9	0.69303	0	1	3	0.60013	0
	h_8	2	4	0.77342	0	2	6	1.0219	0	2	6	0.64475	0
	h_9	2	5	1.0655	0	8	7	0.75772	0	1	3	0.76652	0
	h_{10}	1	2	0.38503	0	9	7	0.56816	0	1	3	0.62524	0

TABLE 2. Numerical results for GMSCG, MSCG and MCG on Example 2.

		GMSCG				MSCG				MCG			
Dimension	IP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	h_1	24	50	0.28243	7.41E−06	10	30	0.12992	6.68E−06	10	30	0.71508	6.68E−06
	h_2	23	48	0.25875	9.94E−06	9	30	0.11893	5.32E−06	1001	20876	35.326	0.001892
	h_3	22	46	0.25342	6.89E−06	9	30	0.14905	3.45E−06	1001	20883	36.0899	0.002147
	h_4	23	48	0.20812	9.87E−06	11	36	0.14646	7.51E−06	1001	20894	36.3485	0.01331
1000	h_5	25	51	0.26527	6.52E−06	10	31	0.13459	4.60E−06	10	31	0.12605	4.60E−06
	h_6	24	50	0.21986	7.06E−06	11	32	0.21021	6.68E−06	11	32	0.086722	6.68E−06
	h_7	21	44	0.22083	8.67E−06	8	27	0.12644	9.70E−06	8	27	0.12117	9.70E−06
	h_8	24	50	0.14728	7.09E−06	11	32	0.13809	6.68E−06	11	32	0.14515	6.68E−06
	h_9	21	44	0.23701	6.66E−06	8	27	0.06749	6.90E−06	8	27	0.098826	6.90E−06
	h_{10}	23	48	0.24117	8.44E−06	9	30	0.12777	4.57E−06	9	30	0.15357	4.57E−06
	h_1	25	52	0.72617	9.55E−06	10	33	0.44161	4.87E−06	10	33	0.4575	4.87E−06
	h_2	25	52	0.89452	7.38E−06	10	30	0.43754	7.41E−06	1001	20846	33.0781	0.000892
	h_3	23	48	0.77507	8.88E−06	9	30	0.54055	7.70E−06	1001	20883	28.5067	0.0048
	h_4	25	52	0.76627	7.37E−06	10	33	0.47916	6.05E−06	1001	20884	33.3151	0.010994
5000	h_5	26	53	0.89815	8.40E−06	11	31	0.44956	6.73E−06	11	31	0.58501	6.73E−06
	h_6	25	52	0.76278	9.09E−06	11	35	0.49757	4.87E−06	11	35	0.53815	4.87E−06
	h_7	23	48	0.7122	6.44E−06	9	30	0.39348	4.62E−06	9	30	0.37743	4.62E−06
	h_8	25	52	0.92517	9.13E−06	11	35	0.66053	4.87E−06	11	35	0.5195	4.87E−06
	h_9	22	46	0.78829	8.59E−06	9	30	0.43509	3.29E−06	9	30	0.59848	3.29E−06
	h_{10}	25	52	0.75556	6.26E−06	10	30	0.38058	6.69E−06	10	30	0.46139	6.69E−06
	h_1	26	54	1.3838	7.78E−06	10	33	0.5516	6.89E−06	10	33	0.69877	6.89E−06
	h_2	26	54	1.4492	6.01E−06	10	33	0.40179	3.34E−06	1001	20854	55.3347	0.001185
	h_3	24	50	1.158	7.23E−06	10	33	0.7495	3.76E−06	1001	20883	55.2341	0.006788
	h_4	26	54	1.467	6.01E−06	10	33	0.74607	5.79E−06	1001	20873	50.8714	0.008442
10 000	h_5	27	55	1.3115	6.85E−06	11	31	0.64505	9.51E−06	11	31	0.84295	9.51E−06
	h_6	26	54	1.3266	7.41E−06	11	35	0.44058	6.89E−06	11	35	0.79943	6.89E−06
	h_7	23	48	0.81319	9.10E−06	9	30	0.61557	6.54E−06	9	30	0.72501	6.54E−06
	h_8	26	54	0.66573	7.44E−06	11	35	0.72625	6.89E−06	11	35	0.67465	6.89E−06
	h_9	23	48	0.3942	6.99E−06	9	30	0.62624	4.66E−06	9	30	0.62545	4.66E−06
	h_{10}	25	52	1.4706	8.85E−06	10	30	0.81565	9.46E−06	10	30	0.68225	9.46E−06
	h_1	28	58	2.7717	5.77E−06	11	36	2.447	3.28E−06	11	36	2.5363	3.28E−06
	h_2	27	56	3.8205	7.75E−06	10	33	2.2812	7.27E−06	1001	20859	3634.083	0.001013
	h_3	25	52	3.7287	9.32E−06	10	33	2.1726	8.41E−06	1001	20883	245.2057	0.015178
	h_4	27	56	3.7844	7.74E−06	10	33	2.3891	8.33E−06	1001	20855	251.6284	0.006866
50 000	h_5	28	57	3.9461	8.82E−06	11	34	2.2154	6.93E−06	11	34	2.2872	6.93E−06
	h_6	27	56	2.0759	9.54E−06	12	38	2.5632	3.28E−06	12	38	2.5422	3.28E−06
	h_7	25	52	3.8887	6.76E−06	10	30	2.0058	9.57E−06	10	30	2.2757	9.57E−06
	h_8	27	56	2.4961	9.59E−06	12	38	2.4783	3.28E−06	12	38	2.5962	3.28E−06
	h_9	24	50	3.621	9.01E−06	10	30	2.0709	6.81E−06	10	30	2.055	6.81E−06
	h_{10}	27	56	4.0541	6.57E−06	10	33	2.2127	6.89E−06	10	33	1.9613	6.89E−06
	h_1	28	58	6.7463	8.17E−06	11	36	4.105	4.64E−06	11	36	4.2416	4.64E−06
	h_2	28	58	3.687	6.31E−06	11	33	3.8055	6.74E−06	1001	20828	19030.98	0.000742
	h_3	26	54	6.5555	7.59E−06	11	36	4.1782	5.87E−06	1001	20883	530.8846	0.021465
	h_4	28	58	6.8555	6.31E−06	11	33	4.0933	8.01E−06	1001	20853	588.6565	0.005267
100 000	h_5	29	59	6.8965	7.19E−06	11	34	2.3284	9.80E−06	11	34	4.077	9.80E−06
	h_6	28	58	6.7454	7.77E−06	12	38	4.3202	4.64E−06	12	38	4.2943	4.64E−06
	x_7	25	52	5.9594	9.55E−06	10	33	3.6202	4.41E−06	10	33	3.815	4.41E−06
	h_8	28	58	6.5556	7.81E−06	12	38	4.3557	4.64E−06	12	38	4.2102	4.64E−06
	h_9	25	52	6.646	7.34E−06	10	30	3.7158	9.63E−06	10	30	3.4943	9.63E−06
	h_{10}	27	56	6.5929	9.29E−06	10	33	3.8819	9.75E−06	10	33	3.815	9.75E−06

TABLE 3. Numerical results for GMSCG, MSCG and MCG on Example 3.

		GMSCG				MSCG				MCG			
Dimension	IP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	h_1	7	16	0.088048	6.62E−06	1001	2002	7.2087	3.66E+01	29	57	0.9684	0
	h_2	6	12	0.072774	7.46E−06	1001	2469	8.5645	2.76E+01	1001	20002	39.3136	35.8519
	h_3	8	16	0.091003	3.99E−06	1001	2349	8.5872	1.93E+01	1001	20000	39.6991	35.971
	h_4	6	14	0.084408	5.93E−06	1001	2403	8.3881	2.65E+01	1001	20002	38.5204	35.7761
1000	h_5	8	18	0.058255	6.62E−06	1001	2002	7.9256	36.6359	39	77	0.36892	0
	h_6	8	17	0.088515	3.09E−06	1001	2004	7.2097	36.6359	6	13	0.088407	0
	h_7	8	16	0.090735	5.09E−06	1001	2002	7.5918	36.6359	10	19	0.10979	0
	h_8	8	17	0.090957	5.69E−06	1001	2004	7.2743	36.6359	20	41	0.20187	0
	h_9	8	16	0.092232	4.25E−06	1001	2002	7.8794	36.6359	67	133	0.60016	0
	h_{10}	7	14	0.091203	7.77E−06	1001	2002	7.5001	36.6359	28	55	0.29357	0
	h_1	8	16	0.30182	4.34E−06	1001	2002	10.9194	81.9204	265	529	4.596	0
	h_2	6	12	0.25056	7.46E−06	1001	2429	11.2035	81.9578	1001	20002	33.8779	80.2698
	h_3	8	16	0.31718	8.91E−06	1001	2475	11.3076	29.8341	1001	20000	34.7956	80.4336
	h_4	6	14	0.25261	5.93E−06	1001	2425	11.2586	75.37	1001	20002	34.6661	80.0622
5000	h_5	9	18	0.34397	4.34E−06	1001	2002	10.641	81.9204	50	99	1.4253	0
	h_6	8	17	0.37324	6.90E−06	1001	2004	10.7079	81.9204	146	293	1.9323	0
	h_7	8	18	0.3617	6.15E−06	1001	2002	10.4367	81.9204	34	67	0.86123	0
	h_8	8	19	0.29008	6.87E−06	1001	2004	10.5881	81.9204	79	159	1.8865	0
	h_9	8	16	0.31664	9.51E−06	1001	2002	10.6987	81.9204	81	161	1.9428	0
	h_{10}	7	16	0.28531	9.38E−06	1001	2002	10.7524	81.9204	313	625	5.1526	0
	h_1	8	16	0.44401	6.13E−06	1001	2002	15.9111	115.8529	182	363	5.3343	0
	h_2	6	12	0.35337	7.46E−06	1001	2374	16.1566	80.7829	1001	20419	58.5714	111.6446
	h_3	8	18	0.48899	6.81E−06	1001	2465	16.4213	137.1481	1001	20000	58.8514	113.7503
	h_4	6	14	0.4061	5.94E−06	1001	2417	15.9737	115.8969	1001	20002	58.4276	113.3604
10 000	h_5	9	18	0.43283	6.13E−06	1001	2002	15.6495	115.8529	54	107	1.9033	0
	h_6	8	17	0.61199	9.76E−06	1001	2004	10.9387	115.8529	1001	2004	14.7077	115.8529
	h_7	8	18	0.47972	8.69E−06	1001	2002	15.5049	115.8529	174	347	5.1424	0
	h_8	8	19	0.47238	9.71E−06	1001	2004	15.4673	115.8529	1001	2004	14.8463	115.8529
	h_9	8	18	0.54812	7.26E−06	1001	2002	15.256	115.8529	1001	2002	15.3887	115.8529
	h_{10}	8	16	0.48738	3.89E−06	1001	2002	15.4452	115.8529	52	103	2.3337	0
	h_1	8	18	1.6153	7.41E−06	1001	2002	55.9852	259.055	1001	2002	55.1742	259.055
	h_2	6	12	1.1043	7.46E−06	1001	2474	59.9754	259.0747	1001	20502	270.4426	254.1064
	h_3	9	18	1.7253	4.46E−06	1001	2360	58.7149	235.16	1001	20000	41559.84	254.3534
	h_4	6	14	1.1935	5.94E−06	1001	2479	59.4377	259.0628	1001	20502	274.5794	252.6789
50 000	h_5	9	19	1.8164	2.96E−21	1001	2002	55.2452	259.055	566	1131	33.2884	0
	h_6	9	19	1.8081	3.45E−06	1001	2004	55.692	259.055	1001	2004	56.4793	259.055
	h_7	9	18	1.6978	5.69E−06	1001	2002	55.8241	259.055	36	71	4.4726	0
	h_8	9	19	1.7769	6.36E−06	1001	2004	55.8441	259.055	1001	2004	56.5054	259.055
	h_9	9	18	1.7454	4.76E−06	1001	2002	55.1154	259.055	1001	2002	55.5949	259.055
	h_{10}	8	16	1.6094	8.69E−06	1001	2002	55.117	259.055	1001	2002	56.1594	259.055
	h_1	9	18	2.9172	3.07E−06	1001	2002	112.5256	366.359	1001	2002	113.1628	366.359
	h_2	6	12	1.7559	7.46E−06	1001	2676	128.2371	350.6418	1001	21002	588.3889	352.7647
	h_3	9	18	2.8055	6.31E−06	1001	2414	121.7566	192.5641	1001	20000	768.7568	359.7101
	h_4	6	14	2.2271	5.94E−06	1001	2567	125.5412	341.5656	1001	21002	1171.552	351.655
100 000	h_5	9	19	3.1257	4.19E−21	1001	2002	111.9677	366.359	1001	2002	119.7642	366.359
	h_6	8	18	2.9928	8.37E−20	1001	2004	107.7143	366.359	1001	2004	154.1149	366.359
	h_7	9	18	2.9141	8.05E−06	1001	2002	111.8237	366.359	1001	2002	112.2792	366.359
	h_8	9	19	3.0479	9.00E−06	1001	2004	111.8116	366.359	1001	2004	233.2247	366.359
	h_9	9	18	2.7432	6.73E−06	1001	2002	111.8536	366.359	639	1277	68.6123	0
	h_{10}	8	17	2.6771	2.51E−20	1001	2002	10911.28	366.359	1001	2002	130.0667	366.359

TABLE 4. Numerical results for GMSCG, MSCG and MCG on Example 4.

		GMSCG				MSCG				MCG			
Dimension	IP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	h_1	0	1	0.026793	0	0	1	0.006596	0	0	1	0.68727	0
	h_2	5	5	0.094292	1.85E−06	264	265	0.71327	9.96E−06	5	5	0.07641	7.92E−07
	h_3	23	23	0.33162	1.87E−08	972	973	2.271	9.99E−06	21	21	0.30121	2.37E−06
	h_4	2	1	1.251	0	9	9	0.026313	0	8	10	0.1468	7.94E−06
1000	h_5	0	1	0.027074	0	0	1	0.006053	0	0	1	0.033599	0
	h_6	1	1	0.033812	0	1	1	0.0091	0	1	1	0.032393	0
	h_7	1	1	0.033281	0	1	1	0.007761	0	1	1	0.023594	0
	h_8	1	1	0.031902	0	1	1	0.007923	0	1	1	0.019537	0
	h_9	26	26	0.35322	1.78E−06	979	979	2.2255	9.99E−06	25	25	0.28407	1.78E−06
	h_{10}	25	25	0.34146	8.43E−07	977	978	2.2329	9.99E−06	24	24	0.29475	3.89E−07
	h_1	0	1	0.32593	0	0	1	0.033775	0	0	1	0.062563	0
	h_2	5	5	0.20406	1.85E−06	264	265	1.7533	9.96E−06	5	5	0.20069	7.92E−07
	h_3	45	45	1.8877	1.07E−08	1001	1001	8.8659	2.11E−05	42	42	1.6714	3.09E−06
	h_4	2	1	0.001212	1.78E−10	9	9	0.10779	1.75E−07	10	12	0.64352	1.23E−08
5000	h_5	0	1	0.15973	0	0	1	0.01781	0	0	1	0.12929	0
	h_6	1	1	0.085041	0	1	1	0.029313	0	1	1	0.1127	0
	h_7	1	1	0.1694	0	1	1	0.034917	0	1	1	0.13651	0
	h_8	1	1	0.070775	0	1	1	0.018645	0	1	1	0.11722	0
	h_9	3	4	1.7999	9.21E−07	1001	1001	8.0521	2.14E−05	48	48	1.6797	1.44E−06
	h_{10}	48	48	1.7239	2.85E−06	1001	1001	7.9453	2.13E−05	46	46	1.5952	7.64E−06
	h_1	0	1	0.070092	0	0	1	0.12249	0	0	1	0.087564	0
	h_2	5	5	0.34226	1.85E−06	264	265	6.7605	9.96E−06	5	5	0.3285	7.92E−07
	h_3	57	57	2.3323	8.18E−06	1001	1001	19.5313	2.99E−05	58	58	2.9832	1.96E−06
	h_4	5	3	0.10232	4.55E−08	9	9	0.83106	4.29E−07	10	12	0.76723	1.76E−08
10 000	h_5	0	1	0.08356	0	0	1	0.091999	0	0	1	0.036114	0
	h_6	1	1	0.11699	0	1	1	0.12716	0	1	1	0.12533	0
	h_7	1	1	0.12821	0	1	1	0.10414	0	1	1	0.10584	0
	h_8	1	1	0.12524	0	1	1	0.073977	0	1	1	0.097108	0
	h_9	67	67	3.4971	2.17E−08	1001	1001	19.8991	3.02E−05	65	65	2.0714	1.36E−06
	h_{10}	65	65	3.2579	1.25E−07	1001	1001	15.5006	3.01E−05	63	63	3.2311	4.40E−06
	h_1	0	1	0.20639	0	0	1	0.23355	0	0	1	0.25505	0
	h_2	5	5	0.97897	1.85E−06	264	265	16.0111	9.96E−06	5	5	0.96627	7.92E−07
	h_3	23	21	13.4778	4.45E−07	1001	1001	73.2477	6.68E−05	129	129	13.2611	5.26E−07
	h_4	4	4	12.7613	1.09E−07	9	9	0.67826	5.37E−07	10	12	2.6469	3.94E−08
50 000	h_5	0	1	0.2722	0	0	1	0.23661	0	0	1	0.41283	0
	h_6	1	1	0.43984	0	1	1	0.22637	0	1	1	0.37382	0
	h_7	1	1	0.28637	0	1	1	0.53305	0	1	1	0.37196	0
	h_8	1	1	0.41841	0	1	1	0.4213	0	1	1	0.22925	0
	h_9	3	3	14.1738	5.75E−07	1001	1001	72.9415	6.76E−05	134	134	13.0977	8.23E−06
	h_{10}	134	134	14.0343	7.39E−08	1001	1001	73.4396	6.74E−05	133	133	13.3623	3.20E−06
	h_1	0	1	0.31999	0	0	1	0.36835	0	0	1	0.38175	0
	h_2	5	5	1.6167	1.85E−06	264	265	28.3776	9.96E−06	5	5	1.4024	7.92E−07
	h_3	11	12	28.0237	8.29E−07	1001	1001	143.4662	9.44E−05	182	182	30.3476	7.33E−07
	h_4	6	5	21.2721	1.13E−07	9	9	3.8263	5.38E−07	10	12	4.1623	5.28E−08
100 000	h_5	0	1	0.45975	0	0	1	0.45819	0	0	1	0.48786	0
	h_6	1	1	0.75435	0	1	1	0.80317	0	1	1	0.78052	0
	h_7	1	1	0.73622	0	1	1	0.41751	0	1	1	0.66486	0
	h_8	1	1	0.76803	0	1	1	0.37826	0	1	1	0.7419	0
	h_9	18	12	3.12345	3.98E−07	1001	1001	140.7371	9.56E−05	187	187	31.1431	5.79E−07
	h_{10}	4	5	13.3456	5.90E−06	1001	1001	144.409	9.53E−05	185	185	30.6139	2.70E−06

TABLE 5. Numerical results for GMSCG, MSCG and MCG on Example 5.

		GMSCG				MSCG				MCG			
Dimension	IP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	h_1	1	2	0.08934	0	1	3	1.145	0	1	3	0.54528	0
	h_2	1	2	0.027116	0	1	2	0.047644	2.22E−16	1	2	0.089882	2.22E−16
	h_3	1	2	0.027708	0	1	2	0.048182	0	1	2	0.030139	0
	h_4	1	2	0.065917	5.18E−06	20	43	0.17271	7.67E−06	1001	16543	27.6943	0.65042
1000	h_5	1	3	0.019976	0	1	4	0.030092	0	1	4	0.035038	0
	h_6	1	1	0.07518	6.57E−06	2	3	0.039345	0	2	3	0.024135	0
	h_7	1	2	0.028026	0	1	2	0.019875	0	1	2	0.03929	0
	h_8	2	2	0.074987	6.11E−06	2	3	0.053515	0	2	3	0.035048	0
	h_9	1	2	0.028262	0	1	2	0.03608	0	1	2	0.029091	0
	h_{10}	1	2	0.069626	5.48E−06	1	2	0.032267	0	1	2	0.033415	0
	h_1	1	2	0.05737	0	1	3	0.080625	0	1	3	0.094438	0
	h_2	1	2	0.049451	0	1	2	0.17142	2.22E−16	1	2	0.07998	2.22E−16
	h_3	1	2	0.047873	0	1	2	0.071443	0	1	2	0.039151	0
	h_4	2	12	0.1007	5.15E−06	20	43	0.47291	7.67E−06	1001	16541	24.2061	0.64945
5000	h_5	1	3	0.056228	0	1	4	0.33601	0	1	4	0.034643	0
	h_6	1	2	0.30386	7.94E−06	2	3	0.11702	0	2	3	0.016726	0
	h_7	1	2	0.20631	0	1	2	0.048069	0	1	2	0.077514	0
	h_8	1	1	0.2868	7.38E−06	2	3	0.081786	0	2	3	0.36407	0
	h_9	1	2	0.060072	0	1	2	0.060945	0	1	2	0.055347	0
	h_{10}	1	1	0.26116	6.62E−06	1	2	0.053459	0	1	2	0.061745	0
	h_1	1	2	0.078055	0	1	3	0.27211	0	1	3	0.091629	0
	h_2	1	2	0.07968	0	1	2	0.079677	2.22E−16	1	2	0.096415	2.22E−16
	h_3	1	2	0.062512	0	1	2	0.078017	0	1	2	0.065455	0
	h_4	1	12	0.3238	5.15E−06	20	43	0.92067	7.67E−06	1001	16541	39.9377	0.64912
10 000	h_5	1	3	0.092758	0	1	4	0.095824	0	1	4	0.10172	0
	h_6	1	2	0.48266	3.29E−06	2	3	0.10414	0	2	3	0.12015	0
	h_7	1	2	0.067317	0	1	2	0.071388	0	1	2	0.074864	0
	h_8	1	1	0.51732	3.06E−06	2	3	0.1116	0	2	3	0.10044	0
	h_9	1	2	0.087176	0	1	2	0.067994	0	1	2	0.079289	0
	h_{10}	1	1	0.39559	9.36E−06	1	2	0.078734	0	1	2	0.045353	0
	h_1	1	2	0.25744	0	1	3	0.29712	0	1	3	0.15823	0
	h_2	1	2	0.19241	0	1	2	0.24579	2.22E−16	1	2	0.15543	2.22E−16
	h_3	1	2	0.20955	0	1	2	0.23464	0	1	2	0.27495	0
	h_4	1	12	0.85048	5.15E−06	20	43	1.7304	7.67E−06	1001	16542	657.5464	0.65084
50 000	h_5	1	3	0.3709	0	1	4	0.17753	0	1	4	0.096956	0
	h_6	1	1	1.4398	7.35E−06	2	3	0.5636	0	2	3	0.10113	0
	h_7	1	2	0.17806	0	1	2	0.2023	0	1	2	0.081868	0
	h_8	1	1	1.3047	6.84E−06	2	3	0.33337	0	2	3	0.10533	0
	h_9	1	2	0.23735	0	1	2	0.23786	0	1	2	0.083765	0
	h_{10}	1	1	1.3037	6.13E−06	1	2	0.19439	0	1	2	0.067443	0
	h_1	1	2	0.42723	0	1	3	0.48876	0	1	3	0.13591	0
	h_2	1	2	0.34263	0	1	2	0.28514	2.22E−16	1	2	0.10806	2.22E−16
	h_3	1	2	0.35792	0	1	2	0.57844	0	1	2	0.10745	0
	h_4	1	11	1.4598	5.15E−06	20	43	3.8916	7.67E−06	1001	16542	338.6779	0.6508
100 000	h_5	1	3	0.55762	0	1	4	0.69611	0	1	4	0.66005	0
	h_6	1	2	2.5994	5.62E−06	2	3	0.64471	0	2	3	0.59861	0
	h_7	1	2	0.45396	0	1	2	0.3787	0	1	2	0.23701	0
	h_8	1	1	2.3793	9.67E−06	2	3	0.89631	0	2	3	0.69281	0
	h_{10}	1	2	0.46899	0	1	2	0.44679	0	1	2	0.45241	0
	x_{10}	8	2	2.2401	8.67E−06	1	2	0.32435	0	1	2	0.48938	0

TABLE 6. Numerical results for GMSCG, MSCG and MCG on Example 6.

		GMSCG				MSCG				MCG			
Dimension	IP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	h_1	66	258	1.1905	6.54E−06	269	1076	2.9957	7.78E−06	1001	15132	42.1165	0.1746
	h_2	16	56	0.21307	9.52E−06	43	169	0.55886	9.94E−06	1001	15706	35.6649	0.023915
	h_3	50	163	0.66863	7.24E−06	39	173	0.59552	7.42E−06	1001	16737	46.5465	7.7766
	h_4	24	82	0.37463	5.95E−06	278	1109	3.2455	6.64E−06	1001	15940	41.9493	0.10137
1000	h_5	58	216	0.56479	9.24E−06	186	746	2.3541	9.83E−06	1001	18356	52.6861	36.4261
	h_6	58	196	0.91213	9.20E−06	261	1041	2.8548	9.06E−06	1001	18237	52.7625	35.5126
	h_7	41	132	0.6511	9.74E−06	133	549	1.6788	9.97E−06	1001	18784	54.3577	38.691
	h_8	65	243	0.88625	8.09E−06	117	469	1.5465	9.28E−06	1001	21106	59.4439	45.9243
	h_9	34	111	0.58186	7.68E−06	268	1070	3.8083	9.63E−06	1001	18574	53.5577	37.686
	h_{10}	31	101	0.49432	8.71E−06	203	810	2.9724	9.99E−06	1001	16891	45.9595	0.091127
	h_1	79	339	2.4718	7.03E−06	234	933	6.3901	8.59E−06	1001	18909	44.5691	92.7012
	h_2	16	56	0.84049	9.52E−06	43	169	1.5693	9.94E−06	1001	15706	33.2705	0.023915
	h_3	53	173	2.174	8.94E−06	36	154	1.6486	5.72E−06	1001	18194	41.55	16.6025
	h_4	24	82	0.79104	5.95E−06	278	1109	7.3743	6.67E−06	1001	15931	36.8037	0.1006
5000	h_5	144	992	5.8242	9.97E−06	177	703	5.2355	4.86E−06	1001	19102	44.7326	91.9368
	h_6	92	426	4.3597	9.23E−06	293	1164	7.4815	8.53E−06	1001	21973	45.0982	118.3279
	h_7	47	151	1.8463	7.82E−06	227	924	6.1033	6.78E−06	1001	19998	46.1698	102.1275
	h_8	79	327	3.1038	6.88E−06	140	568	4.6677	8.96E−06	1001	19222	44.6941	90.6521
	h_9	34	111	1.5865	8.62E−06	150	602	4.8932	6.92E−06	1001	18913	44.7219	94.3146
	h_{10}	38	122	1.8536	8.90E−06	224	892	6.7453	8.77E−06	1001	15173	35.7464	0.47994
	h_1	131	742	8.6255	8.53E−06	272	1084	10.0157	9.29E−06	1001	19976	77.739	141.8067
	h_2	16	56	1.3749	9.52E−06	43	169	2.6957	9.94E−06	1001	15706	80.3913	0.023915
	h_3	54	176	2.5703	8.12E−06	47	196	3.2674	9.62E−06	1001	18527	109.2501	23.5566
	h_4	24	82	2.3932	5.95E−06	278	1109	10.195	6.67E−06	1001	15939	64.7947	0.10148
10 000	h_5	1001	25601	100.6267	8.47E+09	335	1339	10.8922	9.43E−06	1001	19777	82.0466	136.6265
	h_6	189	1272	10.6804	7.00E−06	250	1005	9.7063	9.22E−06	1001	21976	130.725	166.5782
	h_7	524	14055	57.1552	9.68E−06	426	1723	12.7613	8.83E−06	1001	21054	127.4557	167.1554
	h_8	374	6730	31.1912	7.68E−06	352	1410	11.6233	7.44E−06	1001	21976	130.6229	166.6889
	h_9	37	120	2.6415	8.55E−06	169	676	3.4446	9.82E−06	1001	21989	88.255	164.141
	h_{10}	39	125	2.6337	9.88E−06	218	869	9.2019	9.32E−06	1001	15402	66.8369	0.56488
	h_1	614	4570	112.7939	9.20E−06	289	1151	31.9238	9.36E−06	1001	20023	557.4044	341.1336
	h_2	16	56	3.8446	9.52E−06	43	169	7.2007	9.94E−06	1001	15706	364.28	0.023915
	h_3	58	189	9.849	6.96E−06	360	1439	38.7734	5.40E−06	1001	18842	676.2279	52.3374
	h_4	24	82	5.7835	5.95E−06	278	1109	29.6674	6.68E−06	1001	15954	435.2881	0.10123
50 000	h_5	331	2601	65.1575	8.99E−06	165	656	20.5013	9.69E−06	1001	20750	569.4641	341.6135
	h_6	486	3695	90.4675	7.03E−06	140	581	14.1762	8.42E−06	1001	22137	3607.916	372.9863
	h_7	408	3069	76.2199	8.17E−06	95	387	12.766	9.13E−06	1001	21816	13334.5	372.7854
	h_8	612	4839	117.0753	9.38E−06	306	1271	34.5705	9.50E−06	1001	22137	431.9012	373.0574
	h_9	40	129	8.0665	9.88E−06	234	931	25.8395	9.63E−06	1001	21514	426.7617	364.9158
	h_{10}	39	125	7.9096	8.70E−06	213	849	24.5653	9.83E−06	1001	15678	367.1772	1.8942
	h_1	1001	8824	67269.82	1.32E+03	289	1151	63.7151	8.49E−06	1001	21456	16154.99	516.8035
	h_2	16	56	5.331	9.52E−06	43	169	12.2252	9.94E−06	1001	15706	1925.096	0.023915
	h_3	58	189	15.9419	9.84E−06	235	941	54.5197	6.29E−06	1001	18900	38823.23	73.9481
	h_4	24	82	5.2549	5.95E−06	278	1109	59.878	6.68E−06	1001	15913	660.4745	0.10166
100 000	h_5	1001	9341	457.633	1.52E+03	199	796	45.6506	2.43E−06	1001	21486	4106.208	489.9509
	h_6	1001	22482	1053.113	2.32E+06	239	988	54.6682	7.47E−06	1001	22238	1617.43	531.7307
	h_7	1001	9679	1244.998	3.38E+03	135	553	31.997	9.99E−06	1001	21513	3595.716	535.1727
	h_8	1001	10668	551.2411	8.65E+03	161	752	42.1788	9.28E−06	1001	22331	1085.891	531.746
	h_9	43	138	13.1659	9.31E−06	198	791	44.2434	8.39E−06	1001	21513	3595.716	535.1727
	h_{10}	36	116	7.4289	9.75E−06	213	849	47.1744	9.83E−06	1001	22331	1085.891	531.746

TABLE 7. Numerical results for GMSCG, MSCG and MCG on Example 7.

		GMSCG				MSCG				MCG			
Dimension	IP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	h_1	8	14	0.3899	4.44E−06	8	38	0.69933	5.73E−06	5	8	1.2013	2.50E−07
	h_2	6	13	0.044053	6.48E−06	9	41	0.13846	9.05E−06	1001	16968	26.5276	0.10213
	h_3	8	15	0.099506	5.16E−06	8	38	0.061283	7.82E−06	1001	17469	31.9602	1.9537
	h_4	8	14	0.086071	3.65E−06	9	41	0.13662	9.03E−06	1001	17749	32.1015	0.33015
1000	h_5	8	16	0.22268	8.35E−06	8	38	0.093299	5.99E−06	1001	12220	0.049992	6.55E−06
	h_6	7	14	0.08167	6.36E−06	9	41	0.12177	9.89E−06	1001	56544	0.08983	1.13E−06
	h_7	8	15	0.076505	7.03E−06	8	38	0.13563	7.06E−06	1001	24553	0.075803	2.61E−07
	h_8	8	15	0.09378	9.02E−06	9	41	0.12299	2.15E−06	1001	45322	0.080205	7.33E−08
	h_9	8	15	0.08705	7.05E−06	8	41	0.10194	7.73E−06	1001	12454	0.086215	3.33E−07
	h_{10}	8	15	0.091227	4.52E−06	9	41	0.12832	8.72E−06	1001	13456	0.066864	4.90E−06
	h_1	8	14	0.39052	9.92E−06	9	41	0.37678	2.55E−06	1001	8	0.3102	5.58E−07
	h_2	6	13	1.7073	6.48E−06	9	41	0.40522	4.03E−06	1001	16968	25.963	0.10213
	h_3	8	17	0.37186	6.23E−06	9	41	0.325	3.48E−06	1001	17469	29.512	4.3686
	h_4	8	14	0.33984	3.66E−06	9	41	0.5337	4.03E−06	1001	17727	24.9286	0.33012
5000	h_5	9	16	0.33521	5.47E−06	8	38	0.3611	5.33E−06	1001	43333	0.25839	6.03E−08
	h_6	7	16	0.32913	7.67E−06	9	41	0.44036	4.40E−06	1001	56785	0.14776	2.52E−06
	h_7	8	17	0.5334	8.48E−06	9	41	0.40584	3.14E−06	1001	89098	0.24782	5.84E−07
	h_8	9	17	0.30002	3.19E−06	9	41	0.34529	4.77E−06	1001	65443	0.28874	1.64E−07
	h_9	8	17	0.39196	8.51E−06	9	41	0.35312	3.44E−06	1001	35676	0.28214	7.45E−07
	h_{10}	8	17	0.33531	5.45E−06	9	41	0.37748	3.88E−06	1001	23432	0.34841	4.51E−08
	h_1	8	16	0.51296	7.58E−06	9	41	0.66604	3.60E−06	1001	12343	0.55054	7.90E−07
	h_2	6	13	0.54193	6.48E−06	9	44	0.63437	5.70E−06	1001	16968	41.8868	0.10213
	h_3	8	17	0.32912	8.82E−06	9	41	0.61119	4.92E−06	1001	17469	49.2697	6.1781
	h_4	8	14	0.41279	3.67E−06	9	44	0.68639	5.70E−06	1001	17725	49.2976	0.33007
10 000	h_5	9	16	0.52574	7.74E−06	8	38	0.77462	7.53E−06	5	8	0.091911	8.53E−08
	h_6	8	16	0.50013	3.18E−06	9	44	0.58494	6.23E−06	5	9	0.090902	3.56E−06
	h_7	9	17	0.49475	3.51E−06	9	41	0.60053	4.44E−06	6	10	0.38507	8.26E−07
	h_8	9	17	0.61516	4.51E−06	9	44	0.53054	6.75E−06	7	12	0.66287	2.32E−07
	h_9	9	17	0.57319	3.53E−06	9	41	0.61747	4.86E−06	6	10	0.42208	1.05E−06
	h_{10}	8	17	0.4904	7.71E−06	9	44	0.57374	5.49E−06	6	10	0.41254	6.37E−08
	h_1	9	16	1.771	4.96E−06	9	44	2.0602	8.06E−06	5	8	1.1854	1.77E−06
	h_2	6	13	1.227	6.48E−06	10	44	2.108	5.10E−06	1001	16968	220.795	0.10213
	h_3	9	17	1.7248	5.78E−06	9	44	1.984	4.40E−06	1001	17469	280.7601	13.8146
	h_4	8	14	1.5493	3.67E−06	10	44	2.1213	5.10E−06	1001	17723	275.3777	0.33005
50 000	h_5	9	18	1.7686	9.34E−06	9	41	1.8345	3.37E−06	1001	8	1.2004	1.91E−07
	h_6	8	16	1.5879	7.11E−06	10	44	2.0405	5.57E−06	1001	9	1.1019	7.96E−06
	h_7	9	17	1.5988	7.86E−06	9	44	1.8856	9.93E−06	6	10	1.1993	1.85E−06
	h_8	9	19	1.8092	5.45E−06	10	44	1.9793	6.04E−06	7	12	1.274	5.18E−07
	h_9	9	17	1.7031	7.88E−06	10	44	1.987	4.35E−06	6	10	1.1092	2.35E−06
	h_{10}	9	17	1.7013	5.05E−06	10	44	2.048	4.91E−06	6	10	1.0859	1.43E−07
	h_1	9	16	2.9967	7.02E−06	10	44	3.599	4.56E−06	5	8	1.7795	2.50E−06
	h_2	6	13	2.17	6.48E−06	10	47	3.6162	7.21E−06	1001	16968	475.834	0.10213
	h_3	9	17	2.8158	8.17E−06	10	44	2.3008	6.22E−06	1001	17469	592.5006	19.5368
	h_4	8	14	2.575	3.67E−06	10	47	3.6748	7.21E−06	1001	17722	594.2151	0.33009
100 000	h_5	10	18	3.1804	3.87E−06	9	41	2.6559	4.76E−06	1001	16968	1.9417	2.70E−07
	h_6	8	18	2.9588	5.43E−06	10	47	3.594	7.87E−06	1001	17469	2.044	4.62E−08
	h_7	9	19	3.2584	6.00E−06	10	44	3.4827	5.62E−06	1001	17722	1.9979	2.61E−06
	h_8	9	19	3.1218	7.70E−06	10	47	3.6202	8.54E−06	1001	16968	2.2134	7.33E−07
	h_9	9	19	3.0896	6.02E−06	10	44	2.1841	6.15E−06	1001	17469	2.006	3.33E−06
	h_{10}	9	17	2.9774	7.15E−06	10	44	3.1812	6.95E−06	1001	17722	2.125	2.02E−07

TABLE 8. Numerical results for GMSCG, MSCG and MCG on Example 8.

		GMSCG				MSCG				MCG			
Dimension	IP	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM	ITER	FVAL	TIME	NORM
	h_1	3	4	0.124554	245.1715	9	46	0.83184	6.12E−06	9	46	1.0139	6.12E−06
	h_2	6	24	0.079856	2.48E−06	15	80	0.26063	8.79E−06	1001	23444	66.2968	238.9184
	h_3	7	28	0.12544	6.19E−06	19	100	0.35738	9.66E−06	1001	19464	54.9336	104.9339
	h_4	7	28	0.12887	4.66E−06	28	142	0.48271	8.47E−06	1001	18471	44.9346	3.2529
1000	h_5	7	30	0.12785	5.63E−06	10	52	0.20312	6.12E−06	10	52	0.24735	6.12E−06
	h_6	8	31	0.12585	2.28E−06	9	49	0.10658	7.86E−06	9	49	0.1974	7.86E−06
	h_7	8	32	0.13615	2.29E−06	10	54	0.27556	8.27E−06	10	54	0.32257	8.27E−06
	h_8	7	31	0.12075	8.89E−06	10	49	0.19759	4.96E−06	10	49	0.26899	4.96E−06
	h_9	7	28	0.12585	2.80E−06	9	45	0.17769	7.83E−06	9	45	0.22769	7.83E−06
	h_{10}	6	28	0.121	9.25E−06	8	45	0.1771	3.94E−06	8	45	0.22647	3.94E−06
	h_1	12	3	0.23234	548.22	9	51	0.62644	4.82E−06	9	51	0.73568	4.82E−06
	h_2	6	24	0.093204	2.48E−06	15	80	0.68634	8.79E−06	1001	23395	46.6025	528.4274
	h_3	7	32	0.11746	7.10E−06	21	110	1.2678	8.05E−06	1001	19518	41.375	233.7801
	h_4	7	28	0.6332	4.65E−06	28	142	1.4966	8.54E−06	1001	18472	34.1099	3.2604
5000	h_5	7	34	0.50564	6.46E−06	10	57	0.7229	4.82E−06	10	57	0.78803	4.82E−06
	h_6	8	31	0.60266	5.09E−06	10	49	0.64745	8.31E−06	10	49	0.64721	8.31E−06
	h_7	8	32	0.50993	5.13E−06	11	54	0.65865	8.74E−06	11	54	0.63653	8.74E−06
	h_8	8	31	0.44368	4.24E−06	10	54	0.74926	3.90E−06	10	54	0.83428	3.90E−06
	h_9	7	28	0.43572	6.26E−06	9	50	0.72075	6.17E−06	9	50	0.71989	6.17E−06
	h_{10}	7	28	0.41065	4.41E−06	8	45	0.53306	8.82E−06	8	45	0.67486	8.82E−06
	h_1	4	3	1.93764	775.3002	9	51	1.1436	6.81E−06	9	51	1.0169	6.81E−06
	h_2	6	24	0.47314	2.48E−06	15	80	1.296	8.79E−06	1001	22479	80.278	728.9908
	h_3	8	32	0.81025	2.14E−06	22	115	1.9222	6.95E−06	1001	19542	73.0158	330.3085
	h_4	7	28	0.75278	4.65E−06	28	142	2.289	8.55E−06	1001	18472	59.1009	3.2609
10 000	h_5	7	34	0.51411	9.13E−06	10	57	1.1598	6.81E−06	10	57	1.0821	6.81E−06
	h_6	8	31	0.75807	7.20E−06	10	54	1.0352	4.14E−06	10	54	0.71777	4.14E−06
	h_7	8	32	0.76016	7.26E−06	11	59	1.1361	4.35E−06	11	59	1.0368	4.35E−06
	h_8	8	31	0.8975	5.99E−06	10	54	0.89671	5.52E−06	10	54	1.0313	5.52E−06
	h_9	7	28	0.66053	8.85E−06	9	50	1.0396	8.72E−06	9	50	1.1897	8.72E−06
	h_{10}	7	28	0.66156	6.23E−06	9	45	0.66637	5.90E−06	9	45	0.8956	5.90E−06
	h_1	9	7	2.49765	1733.624	10	51	3.1329	7.20E−06	10	51	3.021	7.20E−06
	h_2	6	24	1.5793	2.48E−06	15	80	3.4483	8.79E−06	1001	21775	482.4942	1069.367
	h_3	8	32	2.4904	4.79E−06	23	120	5.933	9.49E−06	1001	19607	460.3633	739.3295
	h_4	7	28	2.3686	4.65E−06	28	142	6.5926	8.55E−06	1001	18472	350.0449	3.2613
50 000	h_5	8	34	2.5775	4.35E−06	11	57	3.5797	7.20E−06	11	57	3.5302	7.20E−06
	h_6	8	35	2.4457	8.26E−06	10	54	3.1085	9.25E−06	10	54	3.0178	9.25E−06
	h_7	8	36	1.581	8.33E−06	11	59	3.6021	9.73E−06	11	59	3.4975	9.73E−06
	h_8	8	35	2.457	6.88E−06	11	54	3.2212	5.83E−06	11	54	3.1628	5.83E−06
	h_9	8	32	2.4617	2.16E−06	10	50	3.3661	9.22E−06	10	50	3.0737	9.22E−06
	h_{10}	7	32	1.5045	7.15E−06	9	50	3.203	4.64E−06	9	50	2.9927	4.64E−06
	h_1	8	7	2.3443	2451.715	10	56	5.7241	3.58E−06	10	56	5.4757	3.58E−06
	h_2	6	24	0.95924	2.48E−06	15	80	5.8199	8.79E−06	1001	21456	1140.191	1249.055
	h_3	8	32	1.6032	6.77E−06	24	125	9.3314	8.19E−06	1001	19640	948.2525	1040.631
	h_4	7	28	1.3355	4.65E−06	28	142	10.0163	8.55E−06	1001	18472	749.6972	3.2614
100 000	h_5	8	34	1.6316	6.15E−06	11	62	6.0092	3.58E−06	11	62	5.9397	3.58E−06
	h_6	9	35	1.6831	2.49E−06	11	54	5.6345	6.18E−06	11	54	5.5116	6.18E−06
	h_7	9	36	1.7476	2.51E−06	12	59	6.0253	6.50E−06	12	59	5.7762	6.50E−06
	h_8	8	35	1.6568	9.73E−06	11	54	5.5991	8.25E−06	11	54	5.7588	8.25E−06
	h_9	8	32	1.5257	3.06E−06	10	55	5.4147	4.59E−06	10	55	5.424	4.59E−06
	h_{10}	8	32	1.5219	2.15E−06	9	50	5.1885	6.56E−06	9	50	4.9879	6.56E−06

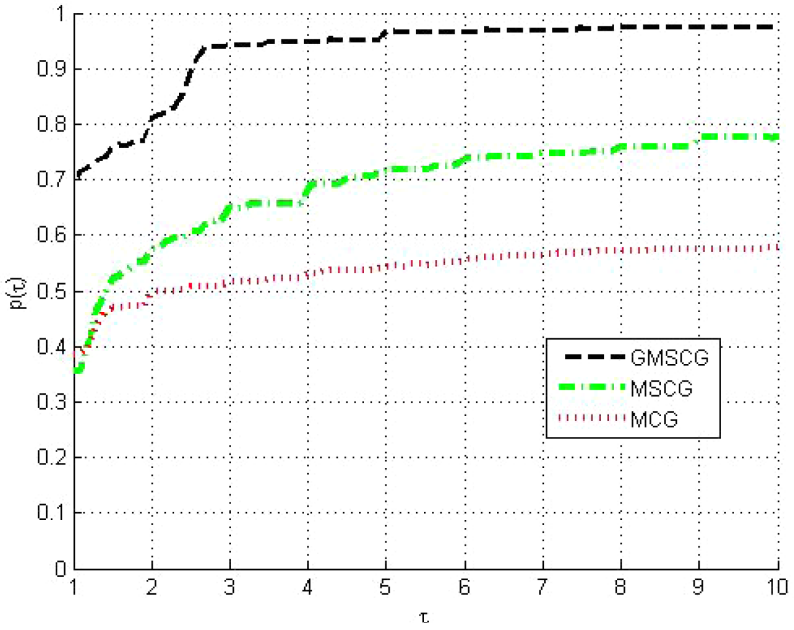

FIGURE 1. Dolan and Moré performance profile with respect to number of iterations.

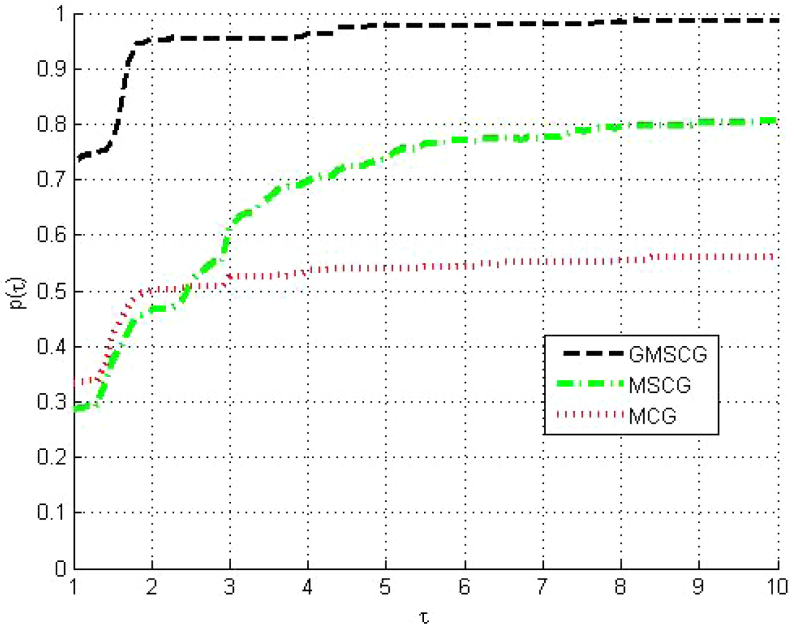

FIGURE 2. Dolan and Moré performance profile with respect to number of function evaluations.

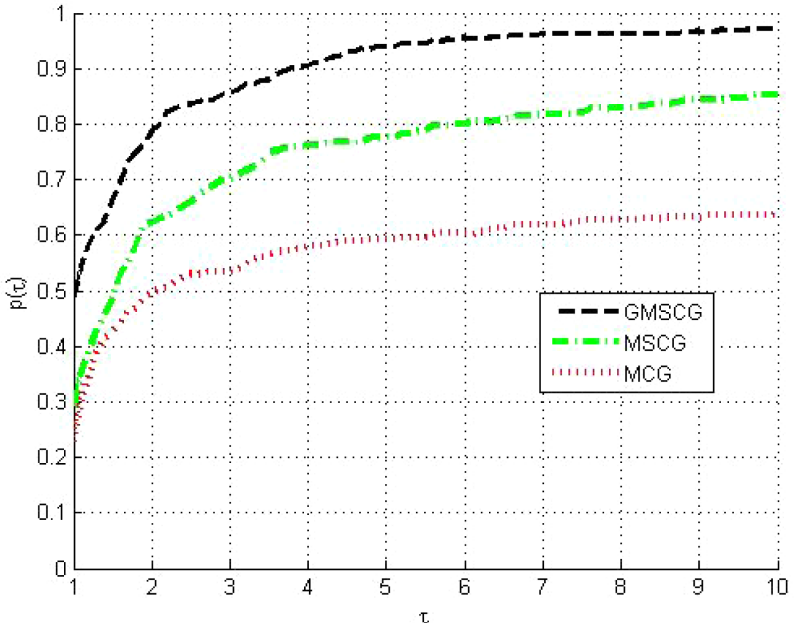

FIGURE 3. Dolan and Moré performance profile with respect to time.

5. Application on signal recovery problems in compressive sensing

Obtaining sparse solutions to ill-conditioned linear systems of equations is a common problem in signal processing and statistical inference. Some of them involve minimizing a quadratic objective function with (ℓ_2) error term and a sparse ℓ_1-regularization term,

$$\min_h \frac{1}{2}\|v - Ah\|_2^2 + \omega\|h\|_1, \tag{35}$$

where $h \in \mathbb{R}^n$, $v \in \mathbb{R}^k$ is an observation, $A \in \mathbb{R}^{k\times n}$ $(k \ll n)$ is linear and $\omega > 0$ is a parameter. Note that (35) is an unconstrained convex optimization problem. The problem (35) arises in compressive sensing due to the sparsity of the original signal, and its solution allows for a precise recovery.

There are solutions to the problem (35), (see [11,12,17,18,40]) and references therein. The gradient projection-based method (GPRS), which was proposed by Figueiredo *et al.* [18], is the most well-known among them. The procedure for the method is as follows. If $h \in \mathbb{R}^n$, then h can be expressed as

$$h = u - r,\ u \geq 0,\ r \geq 0$$

where $u_i = (h_i)_+$, $r_i = (-h_i)_+$ for all $i = 1, 2, \cdots, n$, and $(\cdot)_+ = \max\{0, \cdot\}$. By using $\|h\|_1 = e_n^T u + e_n^T r$, where $e_n = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$, equation (35) can be expressed as

$$\min_{u,r} \frac{1}{2}\|v - A(u - r)\|_2^2 + \omega e_n^T u + \omega e_n^T r, \qquad u \geq 0, \quad r \geq 0, \tag{36}$$

It is a quadratic program with bounds. The equation (36) can, however, be rewritten as

$$\min_z \frac{1}{2} z^T B z + c^T z, \qquad \text{such that} \quad z \geq 0, \tag{37}$$

where $z = \begin{pmatrix} u \\ r \end{pmatrix}$, $c = \omega e_{2n} + \begin{pmatrix} -b \\ b \end{pmatrix}$, $b = A^T y$, $B = \begin{pmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{pmatrix}$. Due to the fact that B is a positive semidefinite matrix, equation (37) is a convex quadratic program.

Xiao *et al.* [43] showed that (37) can be transformed into the problem:

$$G(z) = \min\{z, Bz + c\} = 0. \tag{38}$$

It was also demonstrated that G is continuous and monotone [44]. As a result, solving (35) is identical to solving (38), which has the form (1), which the suggested GMSCG algorithm can solve (35).

All numerical tests were run on a PC with an Intel Celeron (R) processor, 2 GB of RAM, and a CPU speed of 1.60 GHz running Matlab 8.3.0 (R2014a). The proposed GMSCG algorithm is compared with several other methods, including (CGD) by Liu *et al.* [33] method for convex constrained monotone nonlinear equations with applications and the adaptive family of projection methods for constrained monotone nonlinear equations with applications (Algorithms 4.1a and 4.1b) [37].

The experiment's purpose is to recover a length-n sparse signal from k observations, with the mean squared error (MSE) being used to evaluate the recovery. The MSE is defined as:

$$\text{MSE} := \frac{1}{n}\|h_a - h_*\|^2. \tag{39}$$

The original signal is h_a, while the recovered signal is h_*. The signal has a size of $n = 2^{11}$ and a size of $k = 2^9$, with 2^7 randomly non-zero elements. The $randn(k, n)$ program in MATLAB is used to generate the matrix A in (35), and the noise (Gaussian noise) is normally distributed with a mean of 0 and a variance 10^{-3}.

The following parameters are used in the GMSCG implementation: $\mu = 0.02$, $\rho = 0.8$, $\sigma = 0.0.87$, $r = 0.8$ with a merit function defined as follows:

$$f(h) = \frac{1}{2}\|Ah - v\|_2^2 + \omega\|h\|_1. \tag{40}$$

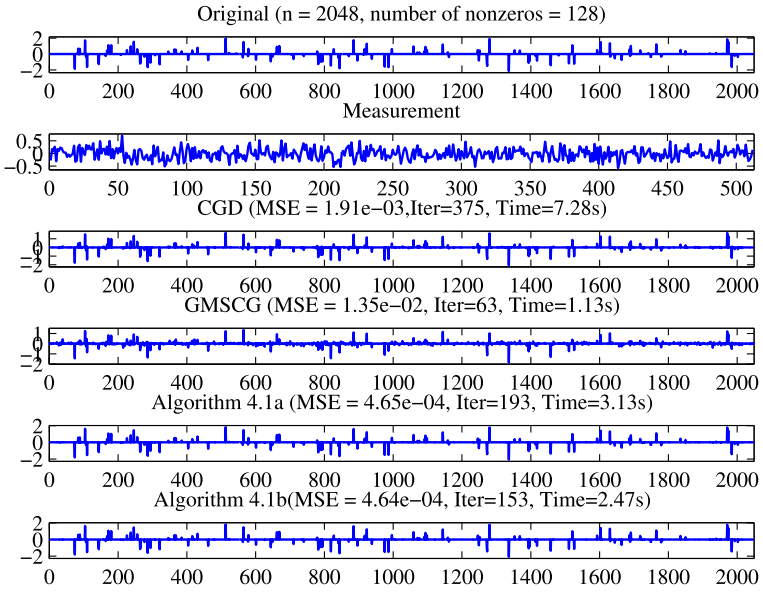

FIGURE 4. A sparse signal is reconstructed. The original signal (First plot), measurement (Second plot), CGD (Third plot), reconstructed signals by GMSCG (Fourth plot), Algorithm 4.1a (Fifth plot), and Algorithm 4.1b (Sixth plot) are shown in order from the top to bottom.

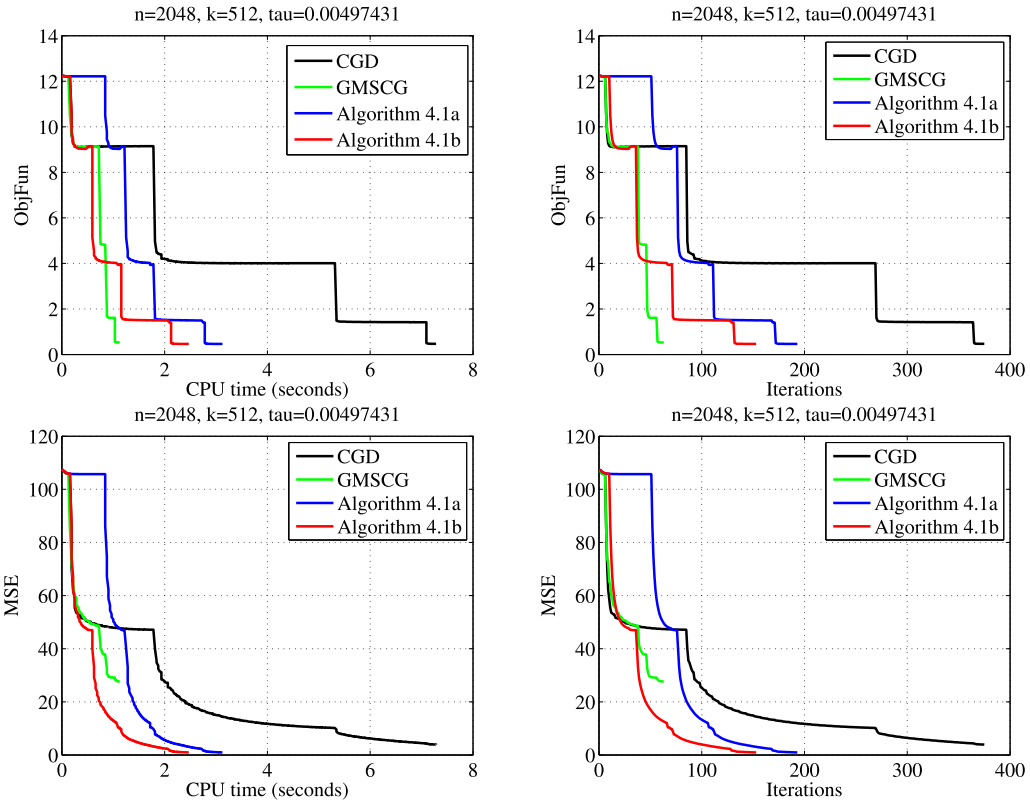

FIGURE 5. CGD, GMSCG, Algorithms 4.1a and 4.1b were compared. The changing trend of MSE corresponds to the number of iterations or CPU time in seconds from left to right, and the changing trend of objective function relates to the number of iterations or CPU time in seconds from left to right.

To achieve fairness in comparison, each code was run from the same initial point, same continuation technique on the parameter ω. That is to say,

$$\omega = 0.005\left\|A^T b\right\|_\infty.$$

$h_0 = N^T b$ is used to start the experiment, and

$$\frac{\|f_k - f_{k-1}\|}{\|f_{k-1}\|} < 10^{-5},$$

f_k is the function value at h_k.

The experiment was done several times to compare the performance of the algorithms. Figure 4 depicts the sparse signal recovery achieved by each approach. In addition, Figure 5 depicts the convergence behavior of CGD, GMSCG, Algorithms 4.1a and 4.1b based on the pattern of MSE and objective function values, as well as the number of iterations and CPU processing time.

GMSCG has better performance as well as plots, according to Table 9. In general, GMSCG recovered the sparse signal with the lowest mean square error and the smallest amount of iterations and CPU time. We may conclude that the GMSCG is a superior alternative to CGD, Algorithms 4.1a and 4.1b.

TABLE 9. Results for the experimental signal recovery problem.

	CGD			GMSCG			Algorithm 4.1a			Algorithm 4.1b		
S/N	TIME	IT	MSE	TIME	IT	MSE	TIME	IT	MSE	TIME	IT	MSE
1	7.28	375	1.914E−03	1.13	63	1.351E−02	3.13	193	4.648E−04	2.47	153	4.643E−04
2	8.17	418	5.251E−03	0.94	57	1.765E−02	4.23	262	2.270E−03	3.64	223	2.270E−03
3	4.47	223	4.614E−03	1.16	59	1.216E−02	3.47	197	8.946E−04	2.63	158	8.946E−04
4	9.44	496	1.079E−03	1.34	81	6.920E−03	2.78	172	8.331E−04	2.16	132	6.751E−04
5	11.47	619	1.513E−03	1.00	61	1.201E−02	3.00	186	1.142E−03	2.31	144	1.145E−03
6	9.03	478	1.005E−03	1.34	74	1.246E−02	2.80	169	6.539E−04	2.14	130	6.539E−04
7	3.17	153	3.697E−03	1.61	84	8.344E−03	3.16	177	7.989E−04	2.28	138	7.997E−04
8	10.56	553	1.263E−03	1.19	72	8.925E−03	3.28	195	8.530E−04	2.59	155	8.585E−04
9	12.31	504	3.180E−03	4.04	226	2.437E−02	2.44	225	2.890E−03	3.77	179	2.890E−03
10	4.75	275	1.986E−03	3.84	177	2.368E−02	2.44	218	6.363E−04	3.52	164	7.289E−04
Average	7.948	402.80	2.912E−03	2.874	161.20	1.04E−02	3.391	205.70	1.153E−03	2.983	173.80	1.112E−03

6. Conclusions

A variation of the self-adaptable conjugate gradient approach with a descent search direction was proposed in this paper, the method is a combination of the projection method and a simple modification of the scheme in [3]. The variation ensures the search direction is descent, increases numerical performance, and preserves the method's attractive convergence qualities. We establish the suggested method's global convergence under appropriate assumptions. The numerical experiments demonstrated that the proposed method has good performance and competes well with some methods in the literature. Also, to further demonstrate its effectiveness, the proposed algorithm was applied to solve a problem in signal processing.

Acknowledgements. The first author would like to thank `Prof Chen Hebai` for his guidance, advice, and support throughout the period of this research. The second author would like to thank the Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University.

Conflict of interest. The authors declare That there is no conflict of interest.

References

[1] M. Abdullahi, A.S. Halilu, A.M. Awwal and N. Pakkaranang, On efficient matrix-free method via quasi-newton approach for solving system of nonlinear equations. *Adv. Theory Nonlinear Anal. App.* **5** (2021) 568–579.

[2] A.B. Abubakar, P. Kumam, A.M. Awwal and P. Thounthong, A modified self-adaptive conjugate gradient method for solving convex constrained monotone nonlinear equations for signal reovery problems. *Mathematics* **7** (2019) 693.

[3] A.B. Abubakar, P. Kumam, H. Mohammad, A.M. Awwal and S. Kanokwan, A modified Fletcher–Reeves conjugate gradient method for monotone nonlinear equations with some applications. *Mathematics* **7** (2019) 745.

[4] A.B. Abubakar, P. Kumam and A.M. Awwal, Global convergence via descent modified three-term conjugate gradient projection algorithm with applications to signal recovery. *Results Appl. Math.* **4** (2019) 100069.

[5] A.B. Abubakar, P. Kumam, A.H. Ibrahim and J. Rilwan, Derivative-free HS-DY-type method for solving nonlinear equations and image restoration. *Heliyon* **6** (2020) e05400.

[6] A.B. Abubakar, P. Kumam, H. Mohammad and A.M. Awwal, A Barzilai–Borwein gradient projection method for sparse signal and blurred image restoration. *J. Franklin Inst.* **357** (2020) 7266–7285.

[7] A.B. Abubakar, Y. Feng and A.H. Ibrahim, Inertial projection method for solving monotone operator equations, in 2022 12th International Conference on Information Science and Technology (ICIST). IEEE (2022) 1–7.

[8] A.B. Abubakar, P. Kumam, H. Mohammad, A.H. Ibrahim and A.I. Kiri, A hybrid approach for finding approximate solutions to constrained nonlinear monotone operator equations with applications. *Appl. Numer. Math.* **177** (2022) 79–92.

[9] A.M. Awwal, P. Kumam and A.B. Abubakar, A modified conjugate gradient method for monotone nonlinear equations with convex constraints. *Appl. Numer. Math.* **145** (2019) 507–520.

[10] S. Babaie-Kafaki and N. Mahdavi-Amiri, Two modified hybrid conjugate gradient methods based on a hybrid secant equation. *Math. Modell. Anal.* **18** (2013) 32–52.

[11] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.* **2** (2009) 183–202.

[12] E.G. Birgin, J. Martínez and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.* **10** (2000) 1196–1211.

[13] Y. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **10** (1999) 177–182.
[14] J.E. Dennis and J.J. Moré, A characterization of superlinear convergence and its application to Quasi-Newton methods. *Math. Comput.* **28** (1974) 549–560.
[15] J. Dennis, E. John and J.J. Moré, Quasi-Newton methods, motivation and theory. *SIAM Rev.* **19** (1977) 46–89.
[16] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. *Math. Program.* **91** (2002) 201–213.
[17] M.A.T. Figueiredo and R.D. Nowak, An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.* **12** (2003) 906–916.
[18] M.A.T. Figueiredo, R.D. Nowak and 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.
[19] R. Fletcher, An overview of unconstrained optimization, in Algorithms for Continuous Optimization. *NATO ASI Series*. Vol. 434. Springer, Dordrecht (1994) 109–143.
[20] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients. *Comput. J.* **7** (1964) 149–154.
[21] A.S. Halilu, A. Majumder, M.Y. Waziri and H. Abdullahi, Double direction and step length method for solving system of nonlinear equations. *Eur. J. Mol. Clin. Med.* **7** (2020) 3899–3913.
[22] A.S. Halilu, M.Y. Waziri and Y.B. Musa, Inexact double step length method for solving systems of nonlinear equations. *Stat. Optim. Inf. Comput.* **8** (2020) 165–174.
[23] A.S. Halilu, A. Majumder, M.Y. Waziri and K. Ahmed, Signal recovery with convex constrained nonlinear monotone equations through conjugate gradient hybrid approach. *Math. Comput. Simul.* **187** (2021) 520–539.
[24] M.R. Hestsenes and E. Stiefel, Methods of conjugate gradients for solving linear equations. *J. Res. Nat. Bur. Stand.* **49** (1952) 409.
[25] A.H. Ibrahim, J. Deepho, A.B. Abubakar and K.O. Aremu, A modified liu-storey-conjugate descent hybrid projection method for convex constrained nonlinear equations and image restoration. *Numer. Alg. Control Optim.* (2021). DOI: 10.3934/naco.2021022.
[26] A.H. Ibrahim, J. Deepho, A.B. Abubakar and A. Kamandi, A globally convergent derivative-free projection algorithm for signal processing. *J. Interdisciplinary Math.* **25** (2022) 2301–2320.
[27] A.H. Ibrahim, P. Kumam, A.B. Abubakar, M.S. Abdullahi and H. Mohammad, A dai-liao-type projection method for monotone nonlinear equations and signal processing. *Demonstratio Math.* **55** (2022) 978–1013.
[28] A.H. Ibrahim, P. Kumam, A.B. Abubakar and J. Abubakar, A derivative-free projection method for nonlinear equations with non-lipschitz operator: application to lasso problem. *Math. Methods Appl. Sci.* **46** (2023) 9006–9027.
[29] R.I. Kachurovskii, Monotone operators and convex functionals. *Uspekhi Matematicheskikh Nauk* **15** (1960) 213–215.
[30] P. Kumam, A.B. Abubakar, M. Malik, A.H. Ibrahim, N. Pakkaranang and B. Panyanak, A hybrid HS-LS conjugate gradient algorithm for unconstrained optimization with applications in motion control and image recovery. *J. Comput. Appl. Math.* **433** (2023) 115304.
[31] W. La Cruz, J. Martínez and M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Math. Comput.* **75** (2006) 1429–1448.
[32] D. Li and M. Fukushima, A globally and superlinearly convergent Gauss–Newton-based BFGS method for symmetric nonlinear equations. *SIAM J. Numer. Anal.* **37** (1999) 152–172.
[33] J.K. Liu and S.J. Li, A projection method for convex constrained monotone nonlinear equations with applications. *Comput. Math. App.* **70** (2015) 2442–2453.
[34] Y. Liu and C. Storey, Efficient generalized conjugate gradient algorithms, part 1: theory. *J. Optim. Theory App.* **69** (1991) 129–137.
[35] K. Meintjes and A.P. Morgan, A methodology for solving chemical equilibrium systems. *Appl. Math. Comput.* **22** (1987) 333–361.
[36] G.J. Minty, Monotone (nonlinear) operators in hilbert space. *Duke Math. J.* **29** (1962) 341–346.
[37] G. Peiting, H. Chuanjiang and L. Yang, An adaptive family of projection methods for constrained monotone nonlinear equations with applications. *Appl. Math. Comput.* **359** (2019) 1–16.
[38] E. Polak and G. Ribiere, Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Math. Model. Numer. Anal. – Modél. Math. Anal. Numér.* **3** (1969) 35–43.
[39] M.V. Solodov and B.F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations, in Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods. Springer (1998) 355–369.
[40] E. Van Den Berg and M.P. Friedlander, Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* **31** (2008) 890–912.
[41] X.Y. Wang, S.J. Li and X.P. Kou, A self-adaptive three-term conjugate gradient method for monotone nonlinear equations with convex constraints. *Calcolo* **53** (2016) 133–145.
[42] A.J. Wood and B.F. Wollenberg, Power Generation, Operation and Control. John Wiley & Sons, New York (1996) 592.
[43] Y. Xiao, Q. Wang and Q. Hu, Non-smooth equations based method for ℓ_1-norm problems with applications to compressed sensing. *Nonlinear Anal. Theory Methods App.* **74** (2011) 3570–3577.
[44] Y. Xiao and H. Zhu, A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing. *J. Math. Anal. Appl.* **405** (2013) 310–319.
[45] G. Yuan and X. Lu, A new backtracking inexact BFGS method for symmetric nonlinear equations. *Comput. Math. App.* **55** (2008) 116–129.

[46] E.H. Zarantonello, Solving Functional Equations by Contractive Averaging. Mathematics Research Center, United States Army, University of Wisconsin (1960).
[47] L. Zheng, L. Yang and Y. Liang, A modified spectral gradient projection method for solving non-linear monotone equations with convex constraints and its application. *IEEE Access* **8** (2020) 92677–92686.
[48] D. Zhifeng, Z. Huan and K. Jie, New technical indicators and stock returns predictability. *Int. Rev. Econ. Finance* **71** (2021) 127–142.