Filomat 38:12 (2024), 4117–4146
https://doi.org/10.2298/FIL2412117H

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Image deblurring problems through accelerated Hager-Zhang projection method for convex constrained monotone nonlinear equations

Abubakar Sani Halilu[a,b,e], Arunava Majumder[b], Mohammed Yusuf Waziri[c,e], Kabiru Ahmed[c,e], Aliyu Muhammed Awwal[d,e]

[a]*Department of Mathematics, Sule Lamido University, Kafin Hausa, Nigeria*
[b]*Department of Mathematics, Lovely Professional University, Phagwara-144411, India*
[c]*Department of Mathematical Sciences, Bayero University, Kano, Nigeria*
[d]*Department of Mathematics, Faculty of Science, Gombe State University, Gombe, Nigeria*
[e]*Numerical Optimization Research Group, Bayero University, Kano, Nigeria*

Abstract. Many researchers have developed interests in finding new choices for the Hager-Zhang nonnegative parameter and developed schemes that generate descent search directions. In this paper, a conjugate gradient (CG) method with the projection technique of Solodov and Svaiter [Kluwer Academic Publishers, (1998), pp. 355-369] to solve constrained monotone nonlinear equations is presented. The proposed method is based on presenting a new value of the Hager-Zhang parameter θ. This is achieved by combining the CG method with the Newton method approach. Moreover, the Jacobian matrix is approximated via acceleration parameter to solve large-scale problems. Under some mild conditions, the proposed method is proven to be globally convergent, and numerical experiments conducted show the efficacy of the proposed method. In addition, the proposed method is successfully applied to handle the ℓ_1-norm regularization problem in image recovery, which exhibits a better result than the existing method in the literature.

1. Introduction

Many researchers in the fields of sciences, engineering, and other relevant areas try to achieve results with models in the form of the system of nonlinear equations

$$F(x) = 0, \quad x \in \Phi, \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is nonlinear map. In addition, the feasible set $\Phi \subset \mathbb{R}^n$ is nonempty, closed and convex. Throughout this paper, the space $\mathbb{R}^n$ denote the $n-$dimensional real space equipped with the Euclidean norm $\|\cdot\|$, $F_k = F(x_k)$.

Furthermore, the problem of nonlinear equations (1) is analogous to the following problem of unconstrained optimization

$$\min f(x), \quad x \in \mathbb{R}^n, \tag{2}$$

2020 *Mathematics Subject Classification*. Primary 65K05; Secondary 90C30, 90C53
Keywords. Hager-Zhang parameters, CG method, Global Convergence, Image Restoration
Received: 04 July 2021; Revised: 06 March 2022; Accepted: 12 February 2024
Communicated by Predrag Stanimirović
Email address: abubakars.halilu@slu.edu.ng (Abubakar Sani Halilu)

$f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be twice continuously differentiable [7]. If a point $x^* \in \mathbb{R}^n$ is the local minimizer of the unconstrained optimization problem in (2) then problem in (1) holds, and it is the first order necessary condition for the global optimization problem in (2) with function F as its gradient [38].

Definition 1.1. *Let $x, y \in \mathbb{R}^n$, a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be monotone if*

$$\langle F(x) - F(y),\ x - y\rangle \geq 0. \tag{3}$$

Problem (1) is called monotone system of nonlinear equations if F satisfies (3).

The monotone mappings, which form a class of nonlinear equations were initially researched by Zarantonello [42], Minty [45], and Kačurovskii [46], in Hilbert spaces. The studies of such mappings are practically applied in many scientific fields, like in the system of economic equilibrium [49], and chemical equilibrium [50]. It also has practical application in ℓ_1-norm regularization problem in signal and image recovery [10, 51]. Some iterative approaches for solving these problems include derivative-free methods [25–27, 41, 52, 54], double step length methods [8, 34, 44, 53], double direction methods [35, 55], Newton methods and their improved version, i.e., the quasi-Newton methods [7, 43, 47, 48]. Typically, iterative procedure of the methods mentioned above are established by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots, \tag{4}$$

where x_{k+1} is the current iterate, x_k is the previous iterate, $\alpha_k > 0$ is a step length that can be computed using any suitable line search, and the search directions d_k can be computed as

$$d_k = -F_k + R_k. \tag{5}$$

It is easy to observe that if $R_k = 0$, then (5) reduces to the steepest descent direction. However, the Newton direction can be obtained if $R_k = (I - (F'_k)^{-1})F_k$, where I is an identity matrix and F'_k is the Jacobian matrix. Consequently, we can find the quasi-Newton directions whenever $R_k = (I - B_k^{-1})F_k$, where B_k is $n \times n$ matrix that approximates the Jacobian matrix. Furthermore, B_k needs to satisfy the secant equation

$$B_k s_{k-1} = y_{k-1}, \tag{6}$$

where, $y_{k-1} = F_k - F_{k-1}$ and $s_{k-1} = x_k - x_{k-1}$.

Despite the attractive characteristics of the Newton and quasi-Newton's methods, the derivative F' or its approximation is computed at each iteration. This renders the schemes not ideal for solving large-scale problems because huge matrix storage is required at each iteration, which is costly in numerical experiments. Matrix-free methods are proposed to overcome these shortfalls. Barzilai and Borwein [6] proposed one of the successful matrix-free methods that generates a sequence of iterates as $x_{k+1} = x_k - \tau_k F_k$ and τ_k is the step length, which can be written as $x_{k+1} = x_k - D_k F_k$, where $D_k = \tau_k I$ which is supposed to satisfy the secant equation (6). The step length τ_k can be obtained by minimizing the least square problems $\min_{\tau} \|\tau s_{k-1} - y_{k-1}\|^2$ and $\min_{\tau} \|s_{k-1} - \tau y_{k-1}\|^2$. These yield

$$\tau_k^{BB1} = \frac{\|s_{k-1}\|^2}{y_{k-1}^T s_{k-1}} \quad \text{and} \quad \tau_k^{BB2} = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}, \tag{7}$$

respectively. Consequently, the author in [36] used Barzilai-Borwein (BB) like method that generates the search direction with diagonal matrix via (5) as

$$d_k = -\tau_k^{BB1} F_k,$$

with $R_k = (I - \tau_k I)F_k$, where $B_k^{-1} \approx \tau_k I$. This has given the method the advantage of solving large-scale problems.

Another variant of matrix-free approaches is the conjugate gradient (CG) method. One can easily observe that if $d_0 = -F_0$, and $R_k = \beta_k d_{k-1}$, then (5) reduces to classical conjugate gradient direction given by

$$d_k = \begin{cases} -F_k, & \text{if} \quad k = 0, \\ -F_k + \beta_k d_{k-1}, & \text{if} \quad k \geq 1, \end{cases} \tag{8}$$

where $F_k = \nabla f(x_k)$ and β_k is a scalar describing the characteristics of the CG methods. However, presenting an efficient conjugate gradient parameter β_k is the rationale behind any CG technique. That is why each CG algorithm corresponds to the CG parameter choice. In CG methods, the search direction d_k is needed to satisfy the conjugacy condition given as

$$d_k^T y_{k-1} = 0. \tag{9}$$

Based on this requirement, by exploiting the quasi-Newton search direction and secant equations, Perry in [30] extended condition in (9) as

$$d_k^T y_{k-1} = -F_k^T s_{k-1}. \tag{10}$$

In order to improve (10), Dai and Liao [24] proposed its variant with nonnegative parameter t given as

$$d_k^T y_{k-1} = -tF_k^T s_{k-1}. \tag{11}$$

By substituting (11) into (8), the Dai and Liao CG parameter is proposed as

$$\beta_k^{DL} = \frac{(y_{k-1} - ts_{k-1})^T F_k}{d_{k-1}^T y_{k-1}}, \qquad t \geq 0. \tag{12}$$

Numerical results have shown that the DL method is efficient; however, it depends significantly on the parameter $t \geq 0$, which its optimal value has yet to be achieved [5]. Furthermore, despite the numerical efficiency of the DL method, its search direction does not satisfy the sufficient descent condition

$$F_k^T d_k \leq -c\|F_k\|^2, \quad c > 0. \tag{13}$$

Interestingly, the CG parameter in (12) reduces to some essential CG methods for specific values of the parameter t. For example, for $t = 0$, (12) reduces to the method by Hestenes and Stiefel [56], i.e,

$$\beta_k^{HS} = \frac{F_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}.$$

Also, the method designed by Hager and Zhang [32], which is defined as

$$\beta_k^{HZ} = \frac{F_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - 2\frac{\|y_{k-1}\|^2 F_k^T d_{k-1}}{(d_{k-1}^T y_{k-1})^2}, \tag{14}$$

represents a special case of β_k^{DL} with the parameter $t = 2\frac{\|y_{k-1}\|^2}{d_{k-1}^T y_{k-1}}$. It has been shown that the search direction developed with (14) satisfies the sufficient descent condition with $c = \frac{7}{8}$. By exploiting Perry's approach [30], Liu and Shang [11] developed a CG method with descent search directions that yields prototypes for which other versions of the Perry scheme such as Hestenes-Stiefel [56] method and Dai and Liao method [24] have been developed. A new Perry CG method that generates descent search directions irrespective of the line search has been developed by Liu, and Xu [12]. Andrei [20] proposed an adaptive class of Perry-type CG algorithms by considering the self-scaling memoryless BFGS update with descent search directions that have been obtained as a result of symmetrization of the CG direction in [30]. For further research and study on other approximations of the DL parameter, the reader may refer to [15–19, 21–23].

Inspired by the friendly properties of CG methods for unconstrained optimization problems, some researchers incorporated the idea to solve the problem (1). Among them, Abubakar and Kumam [13] proposed a Dai-Liao conjugate gradient method for nonlinear equations that satisfied the sufficient descent property. However, Waziri et al. [2] incorporated the idea in [24] and presented a DL method via a modified secant equation. The presented method converged globally using the non-monotone line search proposed in [7]. As a result, Waziri et al. [4] enhanced the convergent property of the approach in [2], to solve monotone nonlinear equations. The two search directions presented in [4] have been shown to be descent using Eigenvalue analysis. Subsequently, Waziri et al. [1] proposed a family of Hager-Zhang CG methods for monotone nonlinear equations, inspired by the unconstrained optimization problems in [33] and the Solodov and Svaiter hyperplane technique [31]. By carrying out eigenvalue analysis, the authors showed that the search directions generated by the schemes were indeed descent directions for a certain value of the parameter θ. The Hager-Zhang CG parameter obtained is defined as

$$\beta_k^{HZ}(\theta) = \frac{F_k^T y_{k-1}}{s_{k-1}^T y_{k-1}} - \theta_k \frac{\|y_{k-1}\|^2 F_k^T s_{k-1}}{(s_{k-1}^T y_{k-1})^2}, \tag{15}$$

where, the Hager-Zhang nonnegative parameter θ_k is given as

$$\theta_k = P - Q\frac{(s_{k-1}^T y_{k-1})^2}{\|s_{k-1}\|^2\|y_{k-1}\|^2}, \quad P \geq \frac{1}{4}, \quad Q \leq 0.$$

Motivated by the projection method [31], many methods have been developed by some researchers to solve constrained monotone nonlinear equations. For instance, Wang et al. [57] incorporated the approach in [31] and solved the monotone nonlinear equations with convex constraint. The linear system of equations is approximately solved after the initialization to obtain a point of trial and then used the projection method to achieve the next iteration. The method in [57] is proven to be globally convergent with a linear rate of convergence. Consequently, to increase the rate of convergence of the approach in [57], its modification was developed by Wang and Wang [58] with a superlinear rate of convergence. Moreover, Abubakar et al. [14] extended the work in [13] and solved the convex constrained monotone nonlinear equations with application in compressing sensing. Awwal et al. [40] incorporated the idea in [30] and proposed a Perry-type projection method that minimized ℓ_1 regularized problem. Sabi'u et al. [28] improved on the work in [1] by obtaining other choices for the Hager-Zhang parameter in [33], which are employed to develop other versions of the scheme in [1]. Recently, Sabi'u et al. [29] extended the Hager-Zhang scheme to solve a system of monotone nonlinear equations with convex constraint by presenting two other choices for the Hager-Zhang parameter using singular value analysis. For further research and study on the projection-based methods for solving monotone nonlinear equations, the interested reader may refer to the following references [3, 36, 38–40].

Therefore, motivated by the Hager-Zhang methods in [1] and the projection technique in [31], the purpose of this article is to develop a method with a new choice of Hager-Zhang nonnegative parameter θ_k in (15) with descent direction that is globally convergent. The paper is organized as follows. In the next section, we will present the proposed method's algorithm. The convergence analysis of the proposed algorithm is shown in section 3. Section 4 lists some numerical experiments and the applications of the proposed approach to signal recovery. The article ends in section 5.

2. New choice of Hager-Zhang nonnegative parameter

Here, the projection operator is introduced to make some assumptions. Let $\Phi \subseteq \mathbb{R}^n$ be a nonempty, closed, and convex set. Then for any $x \in \mathbb{R}^n$, its projection onto Φ is given by

$$P_\Phi[x] = \arg\min\{\|x - y\| : y \in \Phi\}. \tag{16}$$

The mapping $P_\Phi : \mathbb{R}^n \to \Phi$ is known as a projection operator which has nonexpansive property namely, for any $x, y \in \mathbb{R}^n$ it holds that

$$\|P_\Phi[x] - P_\Phi[y]\| \leq \|x - y\|, \tag{17}$$

consequently, we have

$$\|P_{\Phi}[x] - y\| \leq \|x - y\|, \quad \forall y \in \Phi. \tag{18}$$

Now, consider the Hager-Zhang CG parameter defined in (15). For general function F, the CG parameter $\beta_k^{HZ}(\theta)$ at some iteration may be undefined when its denominator becomes zero. For this reason, we propose the following modified Hager-Zhang CG parameter

$$\beta_k^{AHZP}(\theta) = \frac{F_k^T w_{k-1}}{s_{k-1}^T \psi_{k-1}} - \theta_k \frac{\|w_{k-1}\|^2 F_k^T s_{k-1}}{(s_{k-1}^T \psi_{k-1})^2}. \tag{19}$$

Here,

$$\psi_{k-1} = w_{k-1} + \left(1 + \max\left\{0, -\frac{s_{k-1}^T w_{k-1}}{\|s_{k-1}\|^2}\right\}\right) s_{k-1}, \quad w_{k-1} = F_k - F_{k-1} + r s_{k-1},\ r > 0,$$

and $s_{k-1} = z_k - x_k = \alpha_k d_k$. Our search direction is therefore defined as follows.

$$d_k = \begin{cases} -F_k, & \text{if} \quad k = 0, \\ -\eta_k F_k + \beta_k^{AHZP}(\theta) s_{k-1}, & \text{if} \quad k \geq 1, \end{cases} \tag{20}$$

where η_k is a positive parameter that can be determined in such a way that the search direction satisfies the descent condition (13).

Remark 2.1. *We give the following remark.*
If $\max\left\{0, -\frac{s_{k-1}^T w_{k-1}}{\|s_{k-1}\|^2}\right\} \neq 0$ *in the definition of* ψ_{k-1}, *then*

$$\begin{aligned} s_{k-1}^T \psi_{k-1} &= s_{k-1}^T \left(w_{k-1} + s_{k-1} - \frac{s_{k-1}^T w_{k-1}}{\|s_{k-1}\|^2} s_{k-1} \right) \\ &= s_{k-1}^T w_{k-1} + \|s_{k-1}\|^2 - \frac{s_{k-1}^T w_{k-1}}{\|s_{k-1}\|^2} \|s_{k-1}\|^2 \\ &= \|s_{k-1}\|^2 > 0. \end{aligned} \tag{21}$$

Otherwise,

$$\begin{aligned} s_{k-1}^T \psi_{k-1} &= s_{k-1}^T w_{k-1} + \|s_{k-1}\|^2 \\ &= s_{k-1}^T (F_k - F_{k-1} + r s_{k-1}) + \|s_{k-1}\|^2 \\ &= s_{k-1}^T (F_k - F_{k-1}) + r\|s_{k-1}\|^2 + \|s_{k-1}\|^2 \\ &\geq (r+1)\|s_{k-1}\|^2 > 0. \end{aligned} \tag{22}$$

Since the function F is monotone, the last inequality follows. This shows that $s_{k-1}^T \psi_{k-1} > 0$ *provided that the solution is not reached. Therefore,* $\beta_k^{AHZP}(\theta)$ *is well-defined.*

On the other hand, if a point x_k is sufficiently close to the solution say x^*, then Newton's direction is the one to follow. Therefore, from the CG direction in (20) and the Hager-Zhang CG parameter in (19), the value of parameter θ_k can be computed as follows:

$$-(F_k')^{-1} F_k = -F_k + \frac{F_k^T w_{k-1}}{s_{k-1}^T \psi_{k-1}} s_{k-1} - \theta_k \frac{\|w_{k-1}\|^2 F_k^T s_{k-1}}{(s_{k-1}^T \psi_{k-1})^2} s_{k-1}. \tag{23}$$

With a view to avoiding the computation of the Jacobian matrix F'_k and its inverse, we are interested in approximating the Jacobian matrix via

$$F'_k \approx \gamma_k I, \tag{24}$$

where, γ_k is an acceleration parameter presented as

$$\gamma_k = \tau_k^{BB1}(\tau_k^{BB2})^{-1} = \frac{\|y_{k-1}\|^2\|s_{k-1}\|^2}{(y_{k-1}^T s_{k-1})^2}, \tag{25}$$

where, τ_k^{BB1} and τ_k^{BB2} are in (7). However, (24) can be modified as

$$F'_k \approx \hat{\gamma_k} I, \tag{26}$$

with γ_k in (25) modified as

$$\hat{\gamma_k} = \frac{\|w_{k-1}\|^2\|s_{k-1}\|^2}{(\psi_{k-1}^T s_{k-1})^2}. \tag{27}$$

By substituting (26) and (27) in (23), and applying some algebraic calculation, the proposed value of the parameter θ_k can be obtained as

$$\theta_k = \frac{F_k^T s_{k-1} - \frac{F_k^T s_{k-1}\|w_{k-1}\|^2\|s_{k-1}\|^2}{(\psi_{k-1}^T s_{k-1})^2} + \frac{F_k^T w_{k-1}\|w_{k-1}\|^2\|s_{k-1}\|^4}{(\psi_{k-1}^T s_{k-1})^3}}{\frac{F_k^T s_{k-1}\|w_{k-1}\|^4\|s_{k-1}\|^4}{(\psi_{k-1}^T s_{k-1})^4}}. \tag{28}$$

To ensure that our proposed parameter θ_k is nonnegative, we incorporate the idea presented in [37]. Therefore the value of the proposed nonnegative parameter denoted as $\hat{\theta}_k$ can be presented as

$$\hat{\theta}_k = \max\left\{\theta_k, \tau\frac{\|w_{k-1}\|^2}{\psi_{k-1}^T s_{k-1}}\right\}. \tag{29}$$

where, $\tau \geq \frac{1}{4}$.

Remark 2.2. *Now, we give the following remark.*
For $k = 0$, we have $F_0^T d_0 = -\|F_0\|^2$. This satisfies (13) *with $c = 1$.*
For $k \geq 0$, we have

$$\begin{aligned}
F_k^T d_k &= F_k^T\left(-\eta_k F_k + \frac{F_k^T w_{k-1}}{s_{k-1}^T\psi_{k-1}}s_{k-1} - \hat{\theta}_k\frac{\|w_{k-1}\|^2 F_k^T s_{k-1}}{(s_{k-1}^T\psi_{k-1})^2}s_{k-1}\right)\\
&= -\eta_k\|F_k\|^2 + \frac{(F_k^T w_{k-1})(F_k^T s_{k-1})}{s_{k-1}^T\psi_{k-1}} - \hat{\theta}_k\frac{\|w_{k-1}\|^2(F_k^T s_{k-1})^2}{(s_{k-1}^T\psi_{k-1})^2}\\
&\leq -\eta_k\|F_k\|^2 + \frac{\|F_k\|^2\|w_{k-1}\|\|s_{k-1}\|}{s_{k-1}^T\psi_{k-1}}\\
&\leq -\eta_k\|F_k\|^2 + \frac{\|F_k\|^2\|w_{k-1}\|\|s_{k-1}\|}{\|s_{k-1}\|^2}\\
&\leq -\eta_k\|F_k\|^2 + \frac{\|F_k\|^2\|w_{k-1}\|}{\|s_{k-1}\|}\\
&\leq -\left(\eta_k - \frac{\|w_{k-1}\|}{\|s_{k-1}\|}\right)\|F_k\|^2.
\end{aligned} \tag{30}$$

For the search direction (20) *to satisfy* (13), *we need*

$$\eta_k \geq c + \frac{\|w_{k-1}\|}{\|s_{k-1}\|}.$$

Without the loss of generality, we select

$$\eta_k = c + \frac{\|w_{k-1}\|}{\|s_{k-1}\|}. \tag{31}$$

Algorithm 1: Accelerated Hager-Zhang projection Method(AHZP)

Input: Given $x_0 \in \Phi$, $d_0 = -F_0$, $\rho \in (0,1)$, $\sigma > 0$, $\epsilon = 10^{-6}$, $0 < \zeta < 2$, and $\xi > 0$, and set $k = 0$.
Step 1: Compute F_k.
Step 2: If $\|F_k\| \leq \epsilon$ then stop, else go to Step 3.
Step 3: Let $\alpha_k = \xi\rho^{m_k}$, with m_k being the smallest nonnegative integer m such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma\alpha_k \|F(x_k + \alpha_k d_k)\| \|d_k\|^2. \tag{32}$$

Step 4: Set $z_k = x_k + \alpha_k d_k$.
Step 5: If $z_k \in \Phi$ and $\|F(z_k)\| \leq \epsilon$, stop, otherwise go to Step 6.
Step 6: Compute the next iterate by

$$x_{k+1} = P_\Phi[x_k - \zeta\lambda_k F(z_k)], \tag{33}$$

where, $\lambda_k = \dfrac{(x_k - z_k)^T F(z_k)}{\|F(z_k)\|^2}$.
Step 7: Compute the search direction

$$d_{k+1} = -\eta_{k+1} F_{k+1} + \left(\frac{F_{k+1}^T w_k}{s_k^T \psi_k} - \hat{\theta}_{k+1} \frac{\|w_k\|^2 F_{k+1}^T s_k}{(s_k^T \psi_k)^2} \right) s_k.$$

where $\hat{\theta}_{k+1}$ is defined in (29).
Step 8: Consider $k = k + 1$ and go to Step 2.

3. Convergence Analysis

In this section, the analysis of the global convergence of AHZP Algorithm is presented.

Assumption 3.1. *We state the following assumption.*

(A1) The mapping F is Lipschitz continuous; namely, there exists a positive constant L such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{34}$$

(A2) The mapping F is uniformly monotone, namely, there exists a positive constant c such that

$$(x - y)^T (F(x) - F(y)) \geq c\|x - y\|^2, \quad \forall x, y \in \mathbb{R}^n. \tag{35}$$

(A3) For any $x \in S^x$ there exist a constant $\delta > 0$ such that

$$\delta dist(x, S^x) \leq \|F(x)\|^2, \quad \forall x \in N(x^*, S^x), \tag{36}$$

where, $dist(x, S^x)$ denotes the distance from x to the solution set S^x,
$N(x^*, \Omega) := \{x \in \mathbb{R}^n \| \|x - x^*\| \leq \delta\}$. This is the local bound condition.

Remark 3.2. *We give the following remark.*
From assumption (A1) we have

$$\|w_{k-1}\| = \|F_k - F_{k-1} + rs_{k-1}\| \leq \|F_k - F_{k-1}\| + r\|s_{k-1}\| \leq (L + r)\|s_{k-1}\|. \tag{37}$$

Remark 3.3. *To establish the global convergence property of AHZP Algorithm, the proposed $\hat{\theta}_k$ in* (29) *needs to be bounded. Therefore, we set*

$$\hat{\theta}_k \leftarrow \min\{\hat{\theta}_k, H\}, \tag{38}$$

where H is a suitable positive number.

Lemma 3.4. *Suppose Assumptions (A1)-(A3) hold. Then there exists a step length α_k satisfying* (32) *for all $k \geq 0$.*

Proof. We assume that there exists a constant $k_0 \geq 0$, such that given any nonnegative integer m, we have

$$-(F(x_{k_0} + \xi\rho^m d_{k_0}))^T d_{k_0} < \sigma\xi\rho^m \|F(x_{k_0} + \xi\rho^m d_{k_0})\|\|d_{k_0}\|^2. \tag{39}$$

Since $\rho \in (0, 1)$, by using assumption $(A2)$, (32) and letting $m \to \infty$, we get

$$-F(x_{k_0})^T d_{k_0} < 0. \tag{40}$$

Also from (30) and (31), it follows that

$$-F(x_{k_0})^T d_{k_0} \geq 0, \tag{41}$$

which clearly contradicts (40). Hence, the line search is well-defined. □

Lemma 3.5. *Suppose Assumptions (A1)-(A3) hold and let $\{x_k\}$ and $\{z_k\}$ be generated by AHZP Algorithm, then $\{x_k\}$ and $\{z_k\}$ are bounded. In addition, we have*

$$\|d_k\| \leq M, \tag{42}$$

$$\lim_{k\to\infty} \|x_k - z_k\| = 0, \tag{43}$$

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{44}$$

Proof. To show the boundedness of the sequences $\{x_k\}$ and $\{z_k\}$, let $\bar{x} \in S^x$ be any solution of (1). Then by monotonicity of F we can write

$$(x_k - \bar{x})^T F(z_k) \geq (x_k - z_k)^T F(z_k). \tag{45}$$

Using the line search condition (32) and definition of z_k, we have

$$(x_k - z_k)^T F(z_k) \geq \sigma\alpha_k^2 \|F(z_k)\|\|d_k\|^2 > 0. \tag{46}$$

Also, using (18) and fact that $x_{k+1} = P_\Phi[x_k - \zeta\lambda_k F(z_k)]$, we have

$$\begin{aligned}
\|x_{k+1} - \bar{x}\|^2 &= \|P_\Phi(x_k - \zeta\lambda_k F(z_k)) - \bar{x}\|^2 \\
&\leq \|x_k - \zeta\lambda_k F(z_k) - \bar{x}\|^2 \\
&= \|x_k - \bar{x}\|^2 - 2\zeta\lambda_k (x_k - \bar{x})^T F(z_k) + \zeta^2\lambda_k^2 \|F(z_k)\|^2 \\
&\leq \|x_k - \bar{x}\|^2 - 2\zeta\lambda_k (x_k - z_k)^T F(z_k) + \zeta^2\lambda_k^2 \|F(z_k)\|^2 \\
&= \|x_k - \bar{x}\|^2 - \zeta(2 - \zeta)\frac{((x_k - z_k)^T F(z_k))^2}{\|F(z_k)\|^2} \qquad (47) \\
&\leq \|x_k - \bar{x}\|^2 - \widehat{\sigma}\|x_k - z_k\|^4, \qquad (48)
\end{aligned}$$

where $\widehat{\sigma} = \zeta(2-\zeta)\sigma^2 > 0$.
Since $0 < \zeta < 2$, from (47) we obtain

$$\|x_{k+1} - \bar{x}\| \leq \|x_k - \bar{x}\|. \tag{49}$$

This recursively implies that $\|x_k - \bar{x}\| \leq \|x_0 - \bar{x}\|$, for all k. So, $\{\|x_k - \bar{x}\|\}$ is clearly a decreasing sequence, which implies that $\{x_k\}$ is bounded. Furthermore, utilizing assumption (A1), (A3) and (49) we have

$$\|F(x_k)\| = \|F(x_k) - F(\bar{x})\| \leq L\|x_k - \bar{x}\| \leq L\|x_0 - \bar{x}\|. \tag{50}$$

Let $m_1 = L\|x_0 - \bar{x}\|$, then we have

$$\|F(x_k)\| \leq m_1. \tag{51}$$

From (20), (21), (30),(31), (37), and(38) we have

$$\begin{aligned}
\|d_k\| &= \| - \eta_k F_k + \beta_k^{AHZP}(\theta)s_{k-1}\| \\
&\leq \left|c + \frac{\|w_{k-1}\|}{\|s_{k-1}\|}\right|\|F_k\| + \left|\frac{F_k^T w_{k-1}}{s_{k-1}^T\psi_{k-1}} - \hat{\theta}_k\frac{\|w_{k-1}\|^2 F_k^T s_{k-1}}{(s_{k-1}^T\psi_{k-1})^2}\right|\|s_{k-1}\| \\
&\leq \left[c + \frac{\|w_{k-1}\|}{\|s_{k-1}\|}\right]\|F_k\| + \left|\frac{F_k^T w_{k-1}}{s_{k-1}^T\psi_{k-1}}\right|\|s_{k-1}\| + \left|\hat{\theta}_k\frac{\|w_{k-1}\|^2 F_k^T s_{k-1}}{(s_{k-1}^T\psi_{k-1})^2}\right|\|s_{k-1}\| \\
&\leq c\|F_k\| + \frac{\|w_{k-1}\|}{\|s_{k-1}\|}\|F_k\| + \frac{\|w_{k-1}\|\|s_{k-1}\|}{s_{k-1}^T\psi_{k-1}}\|F_k\| + \hat{\theta}_k\frac{\|w_{k-1}\|^2\|s_{k-1}\|^2}{(s_{k-1}^T\psi_{k-1})^2}\|F_k\| \\
&\leq c\|F_k\| + \frac{(L+r)\|s_{k-1}\|}{\|s_{k-1}\|}\|F_k\| + \frac{(L+r)\|s_{k-1}\|^2}{\|s_{k-1}\|^2}\|F_k\| + H\frac{(L+r)^2\|s_{k-1}\|^4}{\|s_{k-1}\|^4}\|F_k\| \\
&= c\|F_k\| + 2(L+r)\|F_k\| + H(L+r)^2\|F_k\| \\
&= [c + 2(L+r) + H(L+r)^2]\|F_k\| \\
&\leq [c + 2(L+r) + H(L+r)^2]m_1.
\end{aligned} \tag{52}$$

Where Cauchy-Schwarz inequality is applied in the third inequality. The fourth inequality follow from Remark (2.1), Remark (3.2), and Remark (3.3) respectively.
Taking $M = [c + 2(L+r) + H(L+r)^2]m_1$ we have (42).

Next, since the sequences $\{x_k\}$ and $\{d_k\}$ are bounded, then the definition z_k in Step 3 of Algorithm 1, it holds that $\{z_k\}$ is also bounded. Therefore, similar argument as in (50), there exists some constants, say $\kappa > 0$, such that

$$\|F(z_k)\| \leq \kappa. \tag{53}$$

Now, from (47), we have

$$((x_k - z_k)^T F(z_{k+1}))^2 \leq \frac{\|F(z_k)\|^2}{\zeta(2-\zeta)}(\|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2). \tag{54}$$

From the line search (32), we have

$$\alpha_k^4\|d_k\|^4 \leq \frac{\alpha_k^2}{\sigma^2}(F(z_k)^T d_k)^2. \tag{55}$$

Combining (47) and (55), it holds

$$\alpha_k^4\|d_k\|^4 \leq \frac{\|F(z_k)\|^2}{\sigma^2\zeta(2-\zeta)}(\|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2). \tag{56}$$

Since the sequence $\{\|x_k - \bar{x}\|\}$ is convergent, using the fact $\sigma > 0$, $0 < \zeta < 2$ and $\{F(z_k)\}$ is bounded, taking limit on both sides of (56) yields

$$\lim_{k\to\infty} \alpha_k^4 \|d_k\|^4 \leq 0,$$

and hence it holds that

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0. \tag{57}$$

Combining (56) with the definition of z_k implies (43) holds. On the other hand, from (18) and the definition of λ_k we have

$$\begin{aligned} \|x_{k+1} - x_k\| &= \|P_\Phi(x_k - \zeta\lambda_k F(z_k)) - x_k\| \\ &\leq \|x_k - \zeta\lambda_k F(z_k) - x_k\| \\ &= \zeta\|\lambda_k F(z_k)\| \\ &\leq \zeta\|x_k - z_k\|. \end{aligned} \tag{58}$$

This implies (44). □

Theorem 3.6. *Suppose Assumption (A1)-(A3) hold and $\{x_k\}$ be generated by AHZP Algorithm. Then*

$$\liminf_{k\to\infty} \|F(x_k)\| = 0. \tag{59}$$

Proof. For the sake of contradiction, suppose that (59) is not true, then there exists $\delta_0 > 0$ such that

$$\|F(x_k)\| \geq \delta_0 \quad \text{holds}, \quad \forall k > 0. \tag{60}$$

Suppose that the search direction $\|d_k\| \neq 0$ unless at the solution, then there exists some constants, say δ_1

$$\|d_k\| \geq \delta_1. \tag{61}$$

If $\alpha_k \neq \xi$, then by the definition of α_k, $\rho^{-1}\alpha_k$ does not satisfies (32) i.e.,

$$-F(x_k + \rho^{-1}\alpha_k d_k)^T d_k < \sigma\rho^{-1}\alpha_k \|F(z_k)\|\|d_k\|^2.$$

Now, combining with (30) and (31) gives

$$\begin{aligned} c\|F(x_k)\|^2 &\leq -F(x_k)^T d_k \\ &= (F(x_k + \rho^{-1}\alpha_k d_k) - F(x_k))^T d_k - F(x_k + \rho^{-1}\alpha_k d_k)^T d_k \\ &\leq L\rho^{-1}\alpha_k \|d_k\|^2 + \sigma\rho^{-1}\alpha_k \|F(z_k)\|\|d_k\|^2 \\ &= \alpha_k \|d_k\|(L + \sigma\kappa)\rho^{-1}\|d_k\|. \end{aligned} \tag{62}$$

Therefore, from (62), we have

$$\alpha_k \|d_k\| > \frac{\rho}{(L + \sigma\kappa)} \frac{c\|F_k\|^2}{\|d_k\|} \geq \frac{\rho}{(L + \sigma\kappa)} \frac{c\delta_0^2}{M}. \tag{63}$$

The inequality (63) contradicts (57). Therefore (59) holds. Hence, the proof is completed. □

The following theorem analyze the linearly convergence rate of the AHZP method.

Theorem 3.7. *Suppose Assumption (A1)-(A3) hold and $\{x_k\}$ be generated by AHZP Algorithm. Then the sequence dist$\{x_k, S^x\}$ is Q-linearly converges to 0.*

Proof. Let $u_k = \arg\min\{\|x_k - u\| | u \in S^x\}$. This means u_k is the nearest solution from x_k i.e., $\|x_k - u_k\| = \text{dist}(x_k, S^x)$. By (30), (31) and Cauchy-Schwarz inequality, we have

$$c\|F_k\|^2 \leq -F_k^T d_k \leq \|F_k\|\|d_k\|, \quad \forall k \in \mathbf{N}. \tag{64}$$

Therefore, from the inequality of (48), for $u_k \in S^x$ we have

$$\begin{aligned} \text{dist}(x_{k+1}, S^x)^2 &\leq \|x_{k+1} - u_k\|^2 \\ &\leq \text{dist}(x_k, S^x)^2 - \widehat{\sigma}\|\alpha_k d_k\|^4 \\ &\leq \text{dist}(x_k, S^x)^2 - \widehat{\sigma}\alpha_k^4 c^4 \|F_k\|^4 \\ &\leq \text{dist}(x_k, S^x)^2 - \widehat{\sigma}\eta^2 c^4 \alpha_k^4 \text{dis}(x_k, S^x)^2 \\ &= (1 - \widehat{\sigma}\eta^2 c^4 \alpha_k^4)\text{dis}(x_k, S^x)^2, \end{aligned} \tag{65}$$

where the third inequality follows from (64) and the fourth inequality follows from (36). By taking $\eta = \frac{1}{c^2\sqrt{\widehat{\sigma}}}$, $(1 - \widehat{\sigma}\eta^2 c^4 \alpha_k^4) \in (0, 1)$ holds. This implies that the sequence $\{\text{dist}(x_k, S^x)\}$ Q-linearly converges to 0. □

4. Numerical Experiments

This section is divided into two parts. In the first part, some numerical results are provided to show the effectiveness of the proposed method by comparing it with the following existing CG methods in the literature.

- DLPM: Algorithm 1 proposed in [14] with $\theta_k = p\frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}} - q\frac{y_{k-1}^T s_{k-1}}{\|s_{k-1}\|^2}$, $p \geq \frac{1}{4}$, $q \leq 0$.
- MHZ2: Algorithm 2 proposed in [29] with

$$\theta_k = \frac{\|s_{k-1}\|^2\|\hat{y}_{k-1}\|^2}{(s_{k-1}^T\hat{y}_{k-1})^2} + \sqrt{\frac{(s_{k-1}^T\hat{y}_{k-1})^2}{\|s_{k-1}\|^2\|\hat{y}_{k-1}\|^2} + \frac{\|s_{k-1}\|^4\|\hat{y}_{k-1}\|^4}{(s_{k-1}^T\hat{y}_{k-1})^4}},$$

$$\hat{y}_{k-1} = y_{k-1} + \rho\frac{\max\{v_{k-1}, 0\}s_{k-1}}{\|s_{k-1}\|}, \quad \rho = 0.1,$$

$$v_{k-1} = 6(\|F_{k-1}\| - \|F_k\|) + 3(F_{k-1} + F_k)^T s_{k-1}.$$

4.1. Numerical Results

The codes used are written in Matlab 8.3.0 (R2014a) and run on a personal computer equipped with a 1.80 GHz CPU processor and 8 GB RAM. When implementing our algorithm (AHZP) in this experiments, the following parameters are set; $\xi = 1$, $\sigma = 10^{-4}$, $\rho = 0.9$, $\tau = 0.4$, and $\zeta = 1.3$. However, the parameters of DLPM and MHZ2 algorithms, are taken as in [14] and [29] respectively. The iteration is set to stop for all the three methods if the following conditions occur: (i) when $\|F_k\| \leq 10^{-7}$, (ii) when $\|F(z_k)\| \leq 10^{-7}$, or (iii) when the iterations exceed 1000 but no x_k satisfying the stopping criterion is obtained. We have carried out the numerical experiments of the three methods on the previous seven test problems with different initial points with dimensions (n values) 1000, 10,000, and 100,000. The initial points used in the experiment are as follows: $x1 = (1, 1, ..., 1)^T$, $x2 = \left(\frac{3}{5}, \frac{3}{5}, ..., \frac{3}{5}\right)^T$, $x3 = \left(\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2}\right)^T$, $x4 = \left(\frac{2}{5}, \frac{2}{5}, ..., \frac{2}{5}\right)^T$, $x5 = \left(\frac{1}{10}, \frac{1}{10}, ..., \frac{1}{10}\right)^T$, $x6 = \left(1, \frac{1}{2}, \frac{1}{3}, ..., \frac{1}{n}\right)^T$, $x7 = \left(\frac{1}{4}, \frac{-1}{4}, ..., \frac{(-1)^n}{4}\right)^T$, $x8 = \left(-\frac{1}{2}, -\frac{1}{2}, ..., -\frac{1}{2}\right)^T$, $x9 = \left(\frac{1}{2}, \frac{1}{2^2}, ..., \frac{1}{2^n}\right)^T$, and $x10 = rand(0, 1)$ which is randomly selected numbers between 0 and 1.

The following test problems were used in the experiments:

Problem 1 [29]

$F_1(x) = e^{x_1} - 1$,

$F_i(x) = e^{x_i} + x_{i-1} - 1, \quad i = 2, 3, \ldots, n-1,$
where $\Phi = \mathbb{R}^n_+$.

Problem 2 [14]
$F_i(x) = 2x_i - \sin|x_i|, \quad i = 1, 2, \ldots, n,$
where $\Phi = \mathbb{R}^n_+$.

Problem 3
$F_i(x) = \cos(x_i) + x_i - 1, \quad i = 1, 2, \ldots, n,$
where $\Phi = \mathbb{R}^n_+$.

Problem 4 [14]
$F_i(x) = e^{x_i} - 1, \quad i = 1, 2, \ldots, n,$
where $\Phi = \mathbb{R}^n_+$.

Problem 5 [40]
$\frac{i}{n}F_i(x) = e^{x_i} - 1, \quad i = 1, 2, \ldots, n,$
where $\Phi = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i \leq n, x_i > -1, \quad i = 1, 2, \ldots, n\}$.

Problem 6 [38]
$F_i(x) = 2x_i - \sin|x_i - 1|, \quad i = 1, 2, \ldots, n,$
where $\Phi = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i \leq n, x_i > -1, \quad i = 1, 2, \ldots, n\}$.

Problem 7 [40]
$F_i(x) = e^{x_i^2} + \frac{3}{2}\sin(2x_i) - 1 \quad i = 1, 2, \ldots, n,$
where $\Phi = \mathbb{R}^n_+$.

Table 1: Numerical results of AHZP, DLPM and MHZ2 methods for problem 1

DIMENSION	IP	AHZP				DLPM				MHZ2			
		Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$
1000	x1	2	3	0.021529	0	18	20	0.024722	4.69E-08	38	78	0.101556	5.89E-08
	x2	2	3	0.002731	0	18	20	0.01409	3.65E-08	44	90	0.111216	7.19E-08
	x3	2	3	0.006608	0	18	20	0.016233	3.56E-08	7	16	0.029903	2.32E-08
	x4	2	3	0.006045	0	17	19	0.011218	9.27E-08	42	86	0.102123	8.81E-08
	x5	2	3	0.004337	0	16	18	0.019832	9.07E-08	13	28	0.038283	4.00E-09
	x6	10	11	0.018082	9.44E-08	21	23	0.020321	5.21E-08	19	40	0.061437	6.38E-09
	x7	9	10	0.017397	8.86E-09	1	2	0.002652	0	8	18	0.013331	1.30E-08
	x8	9	10	0.014801	5.45E-08	1	2	0.002754	0	8	18	0.020729	5.65E-08
	x9	9	10	0.011645	8.60E-08	21	22	0.018334	9.24E-08	18	38	0.068956	1.31E-08
	x10	9	10	0.014222	1.61E-08	18	20	0.015947	9.03E-08	23	47	0.075136	4.71E-09
10,000	x1	2	3	0.017454	0	18	20	0.121666	4.31E-08	11	24	0.213339	4.36E-09
	x2	2	3	0.018829	0	19	21	0.107768	4.01E-08	9	20	0.159113	2.02E-09
	x3	2	3	0.019676	0	19	21	0.104575	3.94E-08	9	20	0.187519	1.31E-09
	x4	2	3	0.026379	0	19	21	0.100954	3.61E-08	11	24	0.191643	1.16E-08
	x5	2	3	0.019183	0	18	20	0.095392	3.53E-08	28	58	0.641475	9.68E-08
	x6	10	11	0.086672	5.17E-08	23	25	0.129757	3.98E-08	21	44	0.526154	5.21E-08
	x7	2	3	0.016616	0	1	2	0.007503	0	6	14	0.053974	6.88E-08
	x8	2	3	0.026333	0	1	2	0.014482	0	8	18	0.087394	8.85E-08
	x9	9	10	0.064302	8.60E-08	21	22	0.110167	9.24E-08	18	38	0.401052	1.31E-08
	x10	9	10	0.083431	4.92E-08	20	21	0.110579	9.63E-08	25	52	0.593106	6.38E-08
100,000	x1	2	3	0.195079	0	23	25	0.898602	4.57E-08	56	114	6.564013	9.81E-08
	x2	2	3	0.488524	0	21	23	0.838763	4.44E-08	31	64	4.241261	9.05E-09
	x3	2	3	0.159751	0	21	23	0.795528	3.95E-08	59	120	9.130102	8.01E-08
	x4	2	3	0.194665	0	20	22	0.764689	9.58E-08	14	30	2.122003	1.37E-08
	x5	2	3	0.153238	0	19	21	0.733677	3.93E-08	16	34	3.601104	2.13E-09
	x6	10	11	0.956477	4.73E-08	24	26	1.004896	6.23E-08	12	26	1.890105	4.72E-08
	x7	2	3	0.157743	0	1	2	0.050328	0	6	14	0.421217	2.55E-08
	x8	2	3	0.114538	0	1	2	0.059275	0	7	16	0.485018	4.16E-08
	x9	9	10	0.492416	8.60E-08	21	22	0.780008	9.24E-08	18	38	3.253116	1.31E-08
	x10	10	11	0.618087	1.48E-08	21	23	0.815919	3.69E-08	31	63	5.474016	7.47E-09

Table 2: Numerical results of AHZP, DLPM and MHZ2 methods for problem 2

DIMENSION	IP	AHZP				DLPM				MHZ2			
		Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$
1000	x1	2	3	0.003886	0	8	10	0.006623	8.29E-08	7	16	0.012408	3.38E-08
	x2	2	3	0.003534	0	8	10	0.005375	6.42E-08	7	16	0.008799	1.56E-08
	x3	2	3	0.005533	0	8	10	0.008875	5.62E-08	7	16	0.008466	1.74E-08
	x4	2	3	0.004285	0	8	10	0.005965	4.69E-08	7	16	0.012986	1.68E-08
	x5	2	3	0.006192	0	8	10	0.008789	1.26E-08	6	14	0.009646	8.43E-08
	x6	17	19	0.021996	3.43E-08	13	14	0.010878	5.98E-08	8	18	0.012509	3.03E-08
	x7	1	2	0.003083	0	1	2	0.004217	0	1	3	0.003718	0
	x8	1	2	0.004019	0	1	2	0.004046	0	1	3	0.003747	0
	x9	2	3	0.004093	0	9	11	0.008247	9.77E-08	7	16	0.009015	4.24E-08
	x10	15	17	0.016228	4.32E-08	15	17	0.023813	4.84E-08	11	24	0.014605	2.12E-08
10,000	x1	2	3	0.016626	0	9	11	0.036462	2.62E-08	8	18	0.067841	0
	x2	2	3	0.014115	0	9	11	0.03275	2.03E-08	7	16	0.064425	4.92E-08
	x3	2	3	0.015007	0	9	11	0.047928	1.78E-08	7	16	0.058328	5.50E-08
	x4	2	3	0.01523	0	9	11	0.036385	1.48E-08	7	16	0.058226	5.32E-08
	x5	2	3	0.014412	0	8	10	0.034329	3.98E-08	7	16	0.057968	1.72E-08
	x6	17	19	0.119694	3.42E-08	15	17	0.072436	3.56E-08	8	18	0.065255	2.95E-08
	x7	1	2	0.009407	0	1	2	0.011902	0	1	3	0.016767	0
	x8	1	2	0.007121	0	1	2	0.009699	0	1	3	0.016486	0
	x9	2	3	0.015029	0	9	11	0.034332	9.77E-08	7	16	0.068562	4.24E-08
	x10	17	19	0.112661	7.01E-08	16	17	0.072877	3.23E-08	11	24	0.106304	6.47E-08
100,000	x1	2	3	0.097043	0	11	13	0.356388	6.27E-08	8	18	0.459701	0
	x2	2	3	0.103969	0	10	12	0.270832	5.04E-08	8	18	0.458115	0
	x3	2	3	0.094429	0	9	11	0.252986	5.62E-08	8	18	0.455731	0
	x4	2	3	0.102661	0	9	11	0.236187	4.69E-08	8	18	0.460901	0
	x5	2	3	0.102626	0	9	11	0.241462	1.26E-08	7	16	0.447212	5.45E-08
	x6	17	19	0.901923	3.42E-08	13	15	0.434628	8.61E-08	8	18	0.489511	2.95E-08
	x7	1	2	0.060183	0	1	2	0.057159	0	1	3	0.122991	0
	x8	1	2	0.056239	0	1	2	0.069409	0	1	3	0.126692	0
	x9	2	3	0.094707	0	9	11	0.270585	9.77E-08	7	16	0.450833	4.24E-08
	x10	19	21	0.975142	4.74E-08	16	17	0.446619	7.62E-08	12	26	0.694912	0

Table 3: Numerical results of AHZP, DLPM and MHZ2 methods for problem 3

DIMENSION	IP	AHZP				DLPM				MHZ2			
		Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$
1000	x1	3	4	0.006845	0	7	9	0.010683	4.21E-08	4	10	0.007088	1.52E-12
	x2	2	3	0.003896	0	6	8	0.010063	4.6E-08	4	10	0.008786	1.07E-10
	x3	2	3	0.003372	0	1	2	0.004354	0	4	10	0.007072	1.29E-12
	x4	1	2	0.002406	0	6	8	0.014063	4.88E-08	3	8	0.007188	1.97E-08
	x5	1	2	0.002909	0	5	7	0.011653	4.86E-08	4	10	0.007445	0
	x6	17	18	0.022451	6.73E-08	15	17	0.025455	8.31E-08	12	26	0.014566	6.29E-08
	x7	1	2	0.004844	0	1	2	0.003449	0	1	3	0.003646	0
	x8	1	2	0.002882	0	1	2	0.005137	0	1	3	0.004788	0
	x9	15	16	0.020974	9.74E-08	34	35	0.057051	7.09E-08	21	43	0.020695	6.80E-08
	x10	19	20	0.021913	7.22E-08	16	17	0.021173	7.41E-09	14	30	0.022804	7.29E-09
10,000	x1	3	4	0.025311	0	8	10	0.068817	3.73E-09	4	10	0.038389	4.80E-12
	x2	2	3	0.014411	0	7	9	0.058538	4.07E-09	4	10	0.035195	3.37E-10
	x3	2	3	0.016878	0	7	9	0.05786	4.16E-09	4	10	0.032745	4.09E-12
	x4	1	2	0.005907	0	7	9	0.060805	4.32E-09	3	8	0.040586	6.22E-08
	x5	1	2	0.007844	0	6	8	0.048647	4.31E-09	4	10	0.037576	0
	x6	17	18	0.115479	6.73E-08	12	14	0.100358	3.71E-08	14	30	0.119411	7.57E-09
	x7	1	2	0.006684	0	1	2	0.011162	0	1	3	0.008491	0
	x8	1	2	0.006599	0	1	2	0.010896	0	1	3	0.008491	0
	x9	15	16	0.100351	9.74E-08	35	36	0.315166	9.31E-09	21	43	0.008115	6.63E-08
	x10	20	21	0.153714	8.26E-08	18	20	0.147069	5.32E-08	14	30	0.008149	2.07E-08
100,000	x1	3	4	0.156368	0	10	12	0.647913	1.33E-08	4	10	0.008491	1.52E-11
	x2	2	3	0.098148	0	8	10	0.487149	1.34E-08	4	10	0.008491	1.07E-09
	x3	2	3	0.090468	0	8	10	0.476052	1.38E-08	4	10	0.008499	1.29E-11
	x4	1	2	0.038694	0	7	9	0.434821	1.37E-08	4	10	0.008492	0
	x5	1	2	0.036204	0	6	8	0.361978	1.36E-08	–	–	–	–
	x6	17	18	0.896099	6.73E-08	12	14	0.715902	2.86E-08	14	30	0.008493	2.76E-08
	x7	1	2	0.045366	0	1	2	0.067343	0	1	3	0.008491	0
	x8	1	2	0.053842	0	1	2	0.081775	0	1	3	0.008439	0
	x9	15	16	0.706414	9.74E-08	35	36	2.415865	9.31E-09	21	43	0.008449	6.63E-08
	x10	21	22	1.113584	9.08E-08	17	18	0.888265	5.74E-09	14	30	0.008492	7.04E-08

Table 4: Numerical results of AHZP, DLPM and MHZ2 methods for problem 4

DIMENSION	IP	AHZP				DLPM				MHZ2			
		Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$
1000	x1	1	2	0.002586	0	9	11	0.008055	1.55E-08	7	16	0.009909	1.68E-08
	x2	2	3	0.003794	0	8	10	0.010698	1.65E-08	5	12	0.008569	5.75E-08
	x3	2	3	0.004594	0	8	10	0.007551	1.65E-08	7	16	0.010662	1.19E-08
	x4	2	3	0.005619	0	8	10	0.009486	1.65E-08	6	14	0.011261	1.37E-08
	x5	2	3	0.003633	0	7	9	0.004957	9.56E-08	–	–	–	–
	x6	3	4	0.005493	0	13	15	0.012161	6.18E-08	13	28	0.019133	1.01E-08
	x7	2	3	0.007998	0	1	2	0.004058	0	1	3	0.002263	0
	x8	2	3	0.006265	0	1	2	0.003805	0	1	3	0.002271	0
	x9	14	15	0.013931	4.25E-08	11	12	0.011529	4.69E-08	11	23	0.015554	1.32E-08
	x10	19	20	0.019262	5.60E-08	14	16	0.012076	1.47E-08	22	46	0.059155	8.57E-08
10,000	x1	1	2	0.009678	0	9	11	0.049923	4.91E-08	7	16	0.062623	5.32E-08
	x2	2	3	0.016263	0	8	10	0.029134	5.22E-08	6	14	0.042619	0
	x3	2	3	0.012927	0	8	10	0.030904	5.2E-08	7	16	0.057385	3.78E-08
	x4	2	3	0.011865	0	8	10	0.027744	5.22E-08	6	14	0.041896	0
	x5	2	3	0.013777	0	8	10	0.027979	3.02E-08	7	16	0.053392	8.98E-09
	x6	3	4	0.018759	0	14	16	0.059262	2.11E-08	15	32	0.136765	2.71E-08
	x7	2	3	0.01321	0	1	2	0.005562	0	1	3	0.006551	0
	x8	2	3	0.016312	0	1	2	0.006919	0	1	3	0.007165	0
	x9	14	15	0.078734	4.25E-08	11	12	0.035089	4.69E-08	11	23	0.081224	1.32E-08
	x10	20	21	0.111599	6.68E-08	16	17	0.063001	9.6E-08	36	73	0.454409	4.16E-08
100,000	x1	1	2	0.043127	0	12	14	0.310319	1.64E-08	8	18	0.413459	0
	x2	2	3	0.081193	0	10	12	0.239006	1.64E-08	6	14	0.311962	0
	x3	2	3	0.107093	0	10	12	0.228038	1.65E-08	8	18	0.387133	0
	x4	2	3	0.087478	0	9	11	0.200339	1.65E-08	7	16	0.363858	8.87E-09
	x5	2	3	0.082591	0	8	10	0.180292	9.56E-08	–	–	–	–
	x6	3	4	0.122075	0	13	15	0.375222	1.62E-08	15	32	0.913306	1.07E-08
	x7	2	3	0.094085	0	1	2	0.027312	0	1	3	0.026445	0
	x8	2	3	0.074505	0	1	2	0.036711	0	1	3	0.023519	0
	x9	14	15	0.502872	4.25E-08	11	12	0.346618	4.69E-08	11	23	0.458557	1.32E-08
	x10	21	22	0.834635	6.60E-08	21	22	0.539635	2.86E-08	34	69	4.317829	7.32E-09

Table 5: Numerical results of AHZP, DLPM and MHZ2 methods for problem 5

DIMENSION	IP	AHZP				DLPM				MHZ2			
		Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$
1000	x1	21	23	0.027733	7.36E-08	20	22	0.018046	1.9E-08	72	146	0.287652	3.76E-08
	x2	21	23	0.029225	8.73E-08	19	21	0.014911	6.5E-08	66	134	0.232369	1.33E-08
	x3	20	22	0.030005	3.64E-08	22	24	0.025712	5.2E-08	58	118	0.238636	1.19E-08
	x4	20	22	0.028747	6.40E-08	17	19	0.013772	2.86E-08	26	54	0.083404	5.38E-08
	x5	19	21	0.029924	8.40E-08	19	21	0.018033	7.86E-08	22	46	0.033511	3.43E-11
	x6	19	21	0.027277	6.23E-08	20	22	0.022314	5.24E-08	23	48	0.063364	6.77E-10
	x7	20	22	0.025127	4.11E-08	21	23	0.018427	1.33E-08	23	48	0.029344	1.12E-08
	x8	20	22	0.030343	3.51E-08	18	20	0.013309	7E-08	24	50	0.041549	1.54E-12
	x9	19	21	0.024382	8.14E-08	23	25	0.02337	3.55E-08	24	50	0.032437	2.76E-08
	x10	22	24	0.033354	9.37E-08	26	28	0.023005	5.67E-08	97	196	0.36152	7.74E-08
10,000	x1	23	25	0.170924	7.05E-08	19	21	0.117426	6.55E-08	116	234	4.616119	3.02E-14
	x2	23	25	0.205809	6.66E-08	21	23	0.129085	9.13E-09	101	204	3.876723	2.81E-08
	x3	23	25	0.176271	6.75E-08	23	25	0.113115	1.16E-08	77	156	2.932597	2.24E-08
	x4	21	23	0.016398	9.88E-08	21	23	0.123108	8.05E-08	31	64	0.507235	2.16E-08
	x5	20	22	0.159046	9.10E-08	18	20	0.100911	2.34E-08	34	70	0.584631	1.79E-08
	x6	20	22	0.150213	9.97E-08	21	23	0.132707	9.05E-08	34	70	0.739965	2.47E-08
	x7	21	23	0.156448	3.72E-08	21	23	0.119906	2.46E-08	37	76	0.760182	4.08E-08
	x8	21	23	0.179718	7.22E-08	19	21	0.103678	3.3E-08	44	90	1.113109	6.57E-08
	x9	20	22	0.156408	9.54E-08	23	25	0.124324	7.27E-08	42	86	1.006519	9.35E-08
	x10	33	35	0.300055	3.98E-08	25	27	0.168255	6.9E-08	103	208	3.985787	4.52E-11
100,000	x1	29	31	1.871459	8.12E-08	30	32	1.322559	9.57E-09	136	274	44.98661	1.98E-11
	x2	33	35	2.222619	8.99E-08	24	26	0.966724	4.65E-08	90	182	24.87766	1.04E-13
	x3	22	24	1.259493	8.90E-08	27	29	1.290821	2.81E-09	156	314	49.74681	5.94E-11
	x4	27	29	1.757906	5.22E-08	23	25	0.853701	3.56E-09	43	88	6.515603	8.74E-08
	x5	21	23	1.529182	8.68E-08	24	26	0.878537	1.41E-08	141	284	43.60521	1.23E-12
	x6	22	24	1.282186	3.67E-08	25	27	0.979922	8.2E-08	98	198	24.36554	8.36E-08
	x7	23	25	1.533765	7.42E-08	22	24	0.874073	9.47E-08	110	222	30.01365	6.84E-08
	x8	27	29	1.618124	5.25E-08	24	26	0.899008	6.51E-08	66	134	16.01713	5.67E-08
	x9	22	24	1.599614	3.81E-08	21	23	0.766356	6.1E-08	108	218	31.08076	4.57E-08
	x10	48	50	3.614278	4.80E-08	26	28	1.173124	4.63E-08	149	300	49.26632	8.94E-09

Table 6: Numerical results of AHZP, DLPM and MHZ2 methods for problem 6

DIMENSION	IP	AHZP				DLPM				MHZ2			
		Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$
1000	x1	5	6	0.008197	1.99E-08	20	22	0.020082	6.72E-08	5	12	0.015091	1.44E-08
	x2	4	6	0.009882	3.38E-08	19	21	0.016673	4.12E-08	4	10	0.012829	7.04E-08
	x3	4	5	0.009503	2.85E-08	16	18	0.014398	6.94E-08	3	8	0.012309	5.60E-08
	x4	4	6	0.010028	6.94E-08	18	20	0.017263	9.06E-08	4	10	0.011619	1.09E-08
	x5	5	6	0.011102	8.70E-09	16	18	0.019424	6.68E-08	5	12	0.013669	1.40E-08
	x6	9	11	0.018667	2.66E-08	24	26	0.021211	4.88E-08	14	30	0.037924	1.62E-08
	x7	5	6	0.008951	6.53E-09	18	20	0.028041	5.96E-08	5	12	0.011778	1.71E-08
	x8	4	6	0.010416	8.38E-08	14	16	0.014701	4.53E-08	4	10	0.018409	4.75E-08
	x9	9	11	0.014754	3.89E-08	18	20	0.016558	4.08E-08	10	22	0.020116	8.53E-08
	x10	10	12	0.016587	8.20E-08	20	22	0.016348	4.37E-08	13	28	0.038621	1.14E-08
10,000	x1	5	6	0.050829	6.29E-08	21	23	0.132844	8.36E-08	5	12	0.075892	4.55E-08
	x2	5	6	0.054451	2.81E-09	20	22	0.101194	5.13E-08	5	12	0.075933	1.54E-08
	x3	4	5	0.037826	9.01E-08	17	19	0.084469	8.63E-08	4	10	0.061303	8.85E-10
	x4	5	6	0.049487	5.77E-09	20	22	0.101155	4.43E-08	4	10	0.057372	3.46E-08
	x5	5	6	0.049131	2.75E-08	17	19	0.088458	8.31E-08	5	12	0.076715	4.43E-08
	x6	10	11	0.084716	6.96E-08	20	22	0.101944	6.42E-08	16	34	0.215221	3.14E-08
	x7	5	6	0.050138	2.06E-08	19	21	0.094151	7.41E-08	5	12	0.074563	5.42E-08
	x8	5	6	0.046593	6.97E-09	15	17	0.078243	5.63E-08	5	12	0.072422	1.04E-08
	x9	9	11	0.088093	6.40E-08	18	20	0.090554	7.96E-08	13	28	0.170664	3.47E-08
	x10	11	13	0.115064	4.25E-08	21	23	0.107088	5.45E-08	13	28	0.207531	5.05E-08
100,000	x1	5	7	0.368131	4.50E-08	23	25	0.832082	7.69E-08	6	14	0.629553	7.19E-10
	x2	5	6	0.306706	8.89E-09	21	23	0.762478	6.38E-08	5	12	0.577905	4.88E-08
	x3	4	6	0.305309	6.44E-08	19	21	0.708245	4.22E-08	4	10	0.449126	2.80E-09
	x4	5	6	0.321479	1.83E-08	21	23	0.771601	5.51E-08	5	12	0.561066	7.59E-09
	x5	5	6	0.342185	8.70E-08	19	21	0.708118	4.07E-08	6	14	0.633854	7.01E-10
	x6	10	12	0.772599	4.73E-08	21	23	0.774138	4.62E-08	25	52	2.971307	2.27E-08
	x7	5	6	0.329965	6.53E-08	21	23	1.088519	5.12E-08	6	14	0.600771	8.57E-10
	x8	5	6	0.315683	2.20E-08	21	23	0.953348	5.42E-08	5	12	0.538802	3.30E-08
	x9	8	9	0.668158	4.41E-08	19	21	0.828752	9.17E-08	13	28	1.278113	1.96E-08
	x10	12	13	0.941143	7.22E-08	22	24	0.935143	6.81E-08	14	30	1.602602	3.05E-08

Table 7: Numerical results of AHZP, DLPM and MHZ2 methods for problem 7

DIMENSION	IP	AHZP				DLPM				MHZ2			
		Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$	Iter	Funeval	Time (s)	$\|F_k\|$
1000	x1	1	2	0.005026	0	9	11	0.007589	8E-08	1	3	0.003413	0
	x2	1	2	0.006373	0	9	11	0.007307	1.14E-08	1	3	0.004025	0
	x3	1	2	0.003898	0	8	10	0.009289	6.58E-08	1	3	0.004507	0
	x4	2	3	0.001587	0	8	10	0.005504	3.57E-08	6	14	0.034503	4.32E-08
	x5	12	14	0.026804	4.22E-08	7	9	0.008721	1.62E-08	6	14	0.030029	1.16E-08
	x6	3	4	0.009235	0	12	14	0.013601	2.29E-08	15	32	0.076977	9.71E-08
	x7	1	2	0.002775	0	1	2	0.005216	0	1	3	0.007172	0
	x8	2	3	0.005357	0	1	2	0.005377	0	1	3	0.006341	0
	x9	13	14	0.038009	7.55E-08	12	13	0.008639	3.43E-08	68	137	4.254011	0
	x10	13	15	0.036556	9.78E-08	15	17	0.013004	2.99E-08	14	30	0.067168	2.23E-08
10,000	x1	1	2	0.013503	0	10	12	0.044684	2.53E-08	1	3	0.009628	0
	x2	1	2	0.012553	0	9	11	0.041937	3.62E-08	1	3	0.008406	0
	x3	1	2	0.008625	0	9	11	0.033604	2.08E-08	1	3	0.007386	0
	x4	2	3	0.014651	0	9	11	0.039714	1.13E-08	7	16	0.199211	5.07E-09
	x5	13	14	0.156798	7.26E-08	7	9	0.026131	5.12E-08	6	14	0.167163	3.66E-08
	x6	3	4	0.047113	0	14	15	0.050996	2.79E-08	19	40	0.514222	5.87E-09
	x7	1	2	0.011784	0	1	2	0.006773	0	1	3	0.026011	0
	x8	2	3	0.016919	0	1	2	0.006426	0	1	3	0.025715	0
	x9	13	14	0.155533	7.55E-08	12	13	0.042429	3.43E-08	52	105	33.66117	0
	x10	14	16	0.170662	6.40E-08	16	18	0.068509	2.53E-08	–	–	–	–
100,000	x1	1	2	0.087812	0	12	14	0.328656	3.55E-08	1	3	0.064636	0
	x2	1	2	0.084601	0	10	12	0.289284	4.09E-08	1	3	0.049102	0
	x3	1	2	0.051268	0	10	12	0.264591	2.54E-08	1	3	0.039347	0
	x4	2	3	0.173898	0	10	12	0.267096	1.53E-08	7	16	1.393001	1.60E-08
	x5	13	15	1.314268	9.11E-08	8	10	0.281714	1.62E-08	7	16	1.353115	4.30E-09
	x6	3	4	0.272453	0	14	15	0.382704	2.68E-08	15	32	3.503018	3.80E-09
	x7	1	2	0.078522	0	1	2	0.041848	0	1	3	0.191312	0
	x8	2	3	0.106994	0	1	2	0.048667	0	1	3	0.180528	0
	x9	13	14	1.066867	7.55E-08	12	13	0.305448	3.43E-08	52	105	278.5842	0
	x10	15	17	1.597813	4.39E-08	21	23	0.679073	1.87E-08	21	43	4.950861	2.38E-08

Table 8: Summary of test results reported in Table 1-7

Methods	Iter	Percentage	Funeval	Percentage	Time (s)	Percentage
AHZP	109	51.91%	138	65.71%	115	54.76%
DLPM	29	13.81%	48	22.86%	66	31.43%
MHZ2	15	7.14%	3	1.43%	29	13.81%
Undecided	57	27.14%	21	10%	0	0%

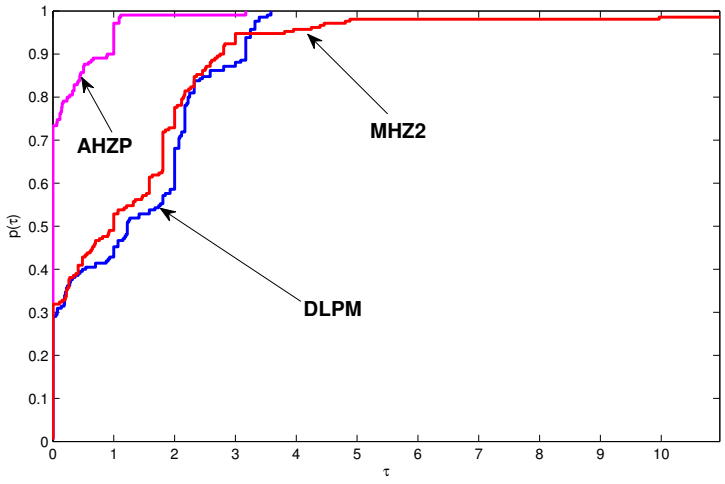

Figure 1: Performance profile for the number of iterations

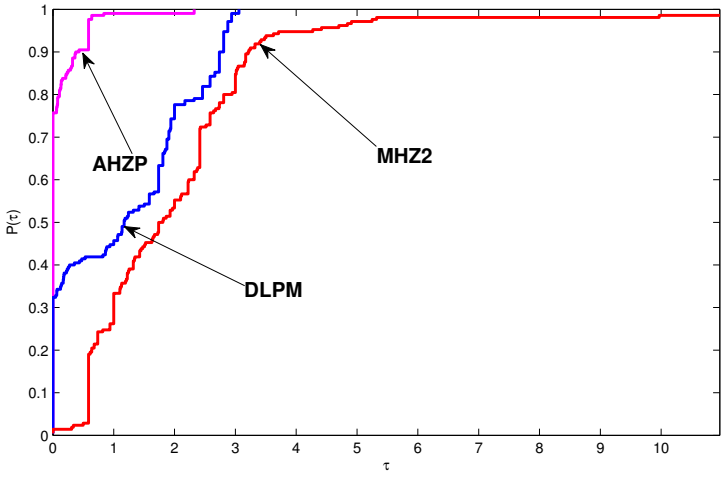

Figure 2: Performance profile for the functions evaluations

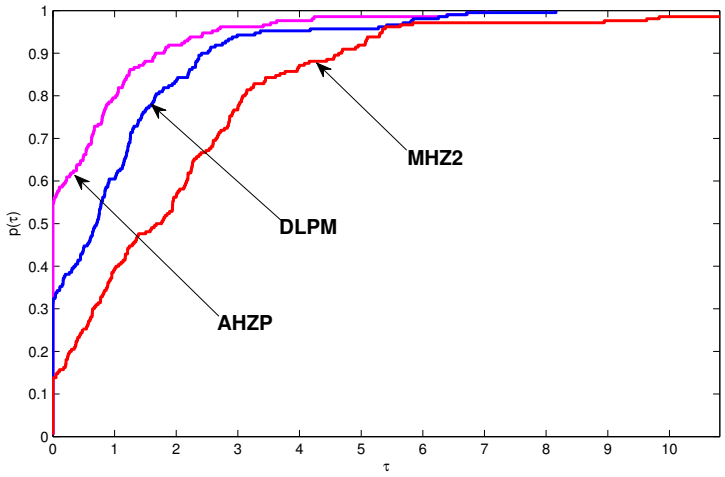

Figure 3: Performance profile for the CPU time (in second)

The numerical results of the three (3) methods are reported in Tables 1-7. From the Tables,

- IP represents the starting points,
- Iter represents the total number of iterations,
- Time (s) represents the CPU time (in seconds),
- Funeval represents the function evaluations,
- $\|F_k\|$ is the norm of the residual at the termination point.

Even though AHZP, DLPM, and MHZ2 methods have successfully solved the seven test problems reported in the numerical experiment, the AHZP method converges faster to the solution of (1) than the other methods because it has the least number of iterations. Moreover, the MHZ2 method failed during the iteration process by clear indication from Tables 3 and 4. From Table 3, MHZ2 failed in 100,000 dimension with initial point $x5$. Also, from 4, MHZ2 failed in 1000 and 100,000 dimensions with the same initial guess $x5$. For the function evaluations presented in Tables 1-7, when compared to the DLPM and MHZ2 methods, it is obvious that the AHZP method has the fewest number of function evaluations, as expected. Furthermore, the numerical results presented in Tables 1-7 demonstrate that the AHZP method solves all test problems in less CPU time than the DLPM and MHZ2 methods. The proposed method outperforms the DLPM and MHZ2 methods for nearly all problems because it has the fewest number of iterations and processor time and the fewest number of function evaluations.

Table 8 summarizes the results from Tables 1-7 to show which approach is best in terms of the number of iterations, function evaluations, and Processor time. From Table 8, the AHZP method wins with the least iterations, about 51.91% of the problems (i.e., 109 out of 210). However, DLPM and MHZ2 methods solve 13.81% (29 out of 210) and 7.14% (15 out of 210) of the problems, respectively. Meanwhile, the results summary shows that at least two of the three methods have the same number of iterations on 57 out of 210 problems, i.e., 27.14% of the problems and reported as undecided in Table 8. The AHZP method solves 65.71% (138 out of 210) of the problems for the function evaluation. At the same time, the DLPM and MHZ2 methods solve 22.86% (48 out of 210) and 1.43% (3 out of 210) of the problems, respectively. However, 10% (21 out of 210) reported being undecided. In terms of CPU time, DLPM and MHZ2 methods solve 31.43% (66 out of 210) and 13.81% (29 out of 210) of the problems, respectively. The AHZP method, on the other hand, solves 54.76% (115 out of 210) of the problems.

Dolan and Moré [59] evaluation tool was used to present a graphical view of each of the three methods used in the experiments to interpret the results presented in Tables 1-7. Figures $1-3$ depict the performance profiles of the AHZP, DLPM, and MHZ2 methods in terms of iteration numbers, number of function evaluations, and CPU time. A fraction $p(\tau)$ of the problems considered for each of the three figures are plotted. A method is within a factor τ of the best time. Each of the three figures has a top curve corresponding to the AHZP scheme. We, therefore, conclude that our proposed algorithm is more effective than the compared ones for solving large-scale nonlinear monotone equations with convex constraint, based on the results from Tables 1-7 and Figures $1-3$.

4.2. Applications in compressive sensing

This part applies the AHZP algorithm to solve image deblurring problems in compressive sensing. The procedure for efficiently acquiring and reconstructing a signal. It compresses the signal received during sensing. Compressive sensing is helpful in a variety of fields, including statistics and signals processing [63, 64]. The convex unconstrained optimization problem below expresses the most common approach in sparse recovery.

$$\min_x \frac{1}{2}\|\bar{w} - \bar{Q}x\|_2^2 + \phi\|x\|_1, \tag{66}$$

where $x \in \mathbb{R}^n, \bar{w} \in \mathbb{R}^m, \bar{Q} \in \mathbb{R}^{m\times n}(m << n)$ denotes a linear operator, the parameter $\phi \geq 0$, and $\|x\|_1 = \sum_{i=1}^{n} |x_i|$. This is usually called ℓ_1-regularized least square problem. Several methods for solving (66) iteratively can be found in [60, 62, 63]. However, gradient-based methods are the most widely used [63], where problem (66) is presented as follows:
Any vector $x \in \mathbb{R}^n$ is divided into positive and negative parts as:

$$x = \bar{p} - \bar{q}, \quad \bar{p} \geq 0, \quad \bar{q} \geq 0, \quad \bar{p}, \bar{q} \in \mathbb{R}^n. \tag{67}$$

Let $\bar{p}_i = (x_i)_+$, and $\bar{q}_i = (-x_i)_+$ for $i = 1, 2, ..., n$, where $(.)_+$ is the positive operator, which is defined as $(x)_+ = \max\{0, x\}$. Applying the definition of the ℓ_1-norm, we have $\|x\|_1 = e_n^T\bar{p} + e_n^T\bar{q}$, with $e_n = (1, 1, 1, ..., 1)^T \in \mathbb{R}^n$. So, problem (66) can be reformulated as the following

$$\min_{\bar{p},\bar{q}} \frac{1}{2}\|\bar{w} - \bar{Q}(\bar{p} - \bar{q})\|_2^2 + \phi e_n^T\bar{p} + \phi e_n^T\bar{q}, \quad \bar{p}, \bar{q} \geq 0. \tag{68}$$

Problem (68) can be expressed as a bound constrained quadratic program, as demonstrated in [63] as follows

$$\min_z \frac{1}{2}z^T\bar{H}z + c^Tz, \quad \text{s.t} \quad z \geq 0, \tag{69}$$

$z = \begin{bmatrix} \bar{p} \\ \bar{q} \end{bmatrix}$, $c = \phi e_{2n} + \begin{bmatrix} -\bar{h} \\ \bar{h} \end{bmatrix}$, $\bar{h} = \bar{Q}^T\bar{w}$, $\bar{H} = \begin{bmatrix} \bar{Q}^T\bar{Q} & -\bar{Q}^T\bar{Q} \\ -\bar{Q}^T\bar{Q} & \bar{Q}^T\bar{Q} \end{bmatrix}$, where $\bar{H}$ is positive semi-definite matrix. Therefore, (69) is a convex quadratic programming problem, which is translated into the following problem of linear variable inequality (LVI) [65]. Find the value of $z \in \mathbb{R}^n$, such that

$$(z' - z)^T(\bar{H}z + c) \geq 0 \quad \forall z' \geq 0. \tag{70}$$

Furthermore, problem in(69) is equivalent to the following linear complementary problem [65]. Find $z \in \mathbb{R}^n$,

$$z \geq 0, \quad \bar{H}z + c \geq 0, \quad \text{and} \quad z^T(\bar{H}z + c) = 0. \tag{71}$$

where, $z \in \mathbb{R}^n$ is the solution of (71) if and only if it satisfies the following nonlinear equations

$$F(z) = \min\{z, \bar{H}z + c\} = 0, \tag{72}$$

where, F is a vector-valued function that is Lipschitz continuous, and monotone, as proved in citePang, SGCS, and the " min " interpreted as a component-wise minimum. Therefore, problem (66) can be translated into (1). Therefore, the AHZP algorithm can be applied to solve it.

Numerical tests were performed to demonstrate the effectiveness of the AHZP method in restoring certain blurred images by comparing it with the SGCS method [65]. When implementing AHZP algorithm in this experiments, and the following parameters are set $\xi = 1$, $\sigma = 10^{-4}$, $\rho = 0.5$, $\tau = 0.4$, and $\zeta = 1.3$. The parameters of the SGCS Algorithm are taken as in [65]. The iteration is set to stop for both methods if the following conditions occur:

$$\frac{|f(x_k) - f(x_{k-1})|}{|f(x_{k-1})|} < 10^{-5},$$

with a merit function $f(x)$ define as $f(x) = \frac{1}{2}\|\bar{w} - \bar{Q}x\|_2^2 + \phi\|x\|_1$. In addition, during the image de-blurring experiment, the codes were ran with $x_0 = \bar{Q}^T\bar{w}$, as initial point. Signal-to-Noise Ratio (SNR) defined by

$$\text{SNR} = 20 \times \log_{10}\left(\frac{\|\hat{x}\|}{\|x - \hat{x}\|}\right),$$

where, $\hat{x}$ and x are the original image and the restored image, respectively. Furthermore, Structural Similarity (SSIM) index is used in this paper in order to measure the quality of the restored images [66]. The MATLAB implementation of the SSIM index can be obtained at `http://www.cns.nyu.edu/~lcv/ssim/`.

Table 9: Numerical results for AHZP and SGCS in image restoration

		AHZP				SGCS			
IMAGE	SIZE	ITER	TIME (s)	SNR	SSIM	ITER	TIME (s)	SNR	SSIM
Figure 4 (Halilu)	256×256	31	4.44	30.31	0.92	200	20.78	30.43	0.92
Figure 5 (Tajmahal)	256×256	30	4.48	23.26	0.85	399	42.58	23.41	0.87
Figure 6 (LPU Mall)	256×256	40	5.91	20.03	0.81	701	72.86	20.02	0.82
Figure 7 (Duck)	256×256	32	4.88	20.84	0.89	345	35.53	20.77	0.90

Figure 4: The original image (first row first column), the blurred image (first row second column), the restored image by AHZP (second row first column) and SGCS (second row second column).

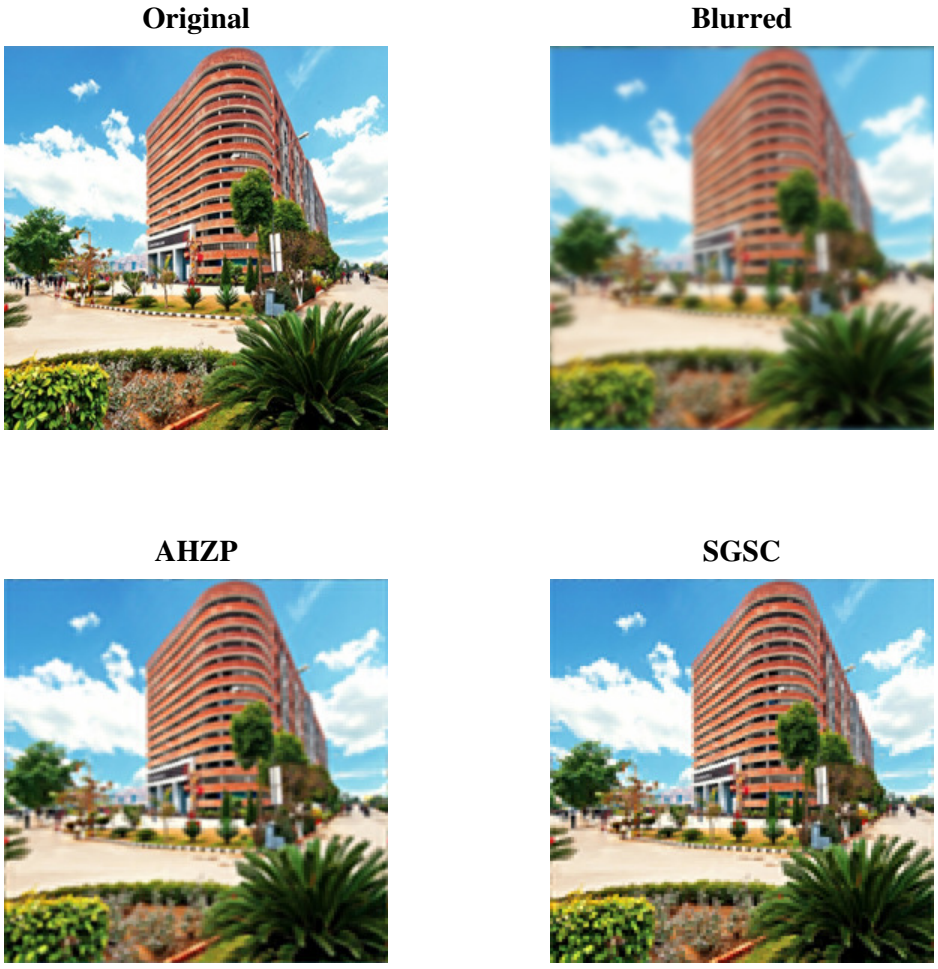

Figure 5: The original image (first row first column), the blurred image (first row second column), the restored image by AHZP (second row first column) and SGCS (second row second column).

Figure 6: The original image (first row first column), the blurred image (first row second column), the restored image by AHZP (second row first column) and SGCS (second row second column).

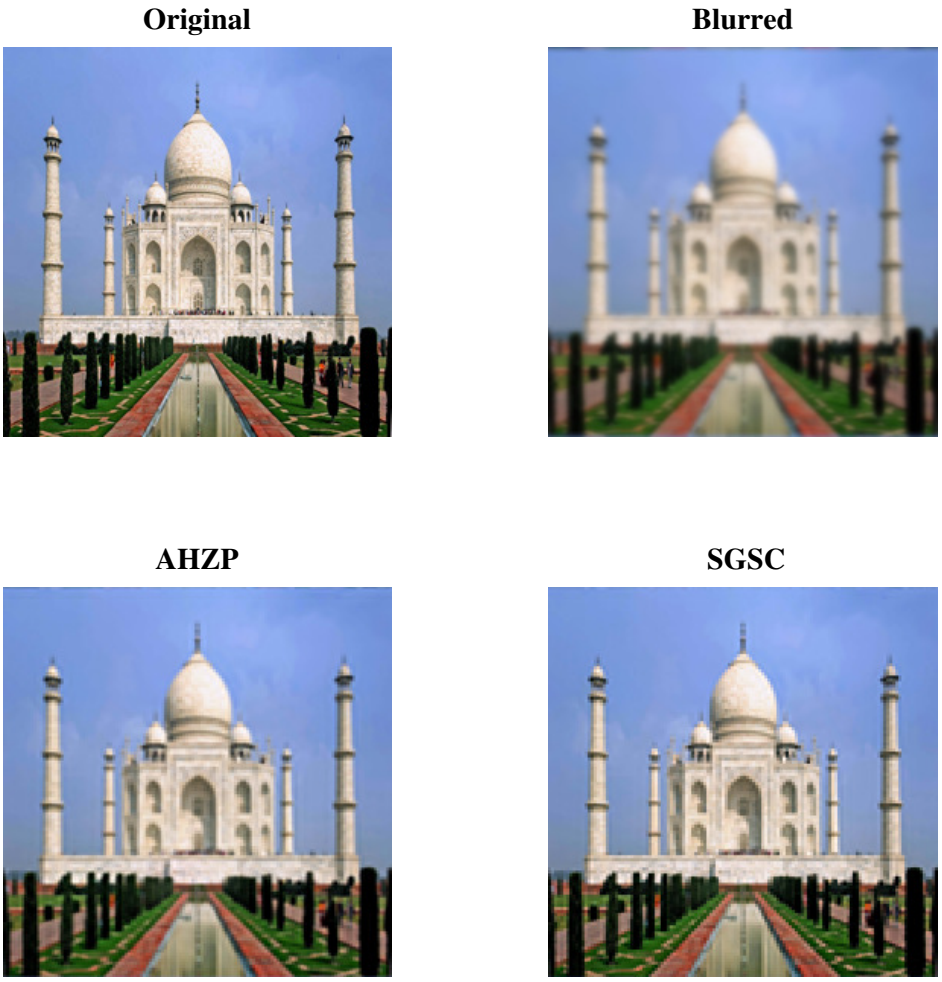

Figure 7: The original image (first row first column), the blurred image (first row second column), the restored image by AHZP (second row first column) and SGCS (second row second column).

Figures (4 – 7) are generated to show the restoration results of different images obtained by AHZP and SGCS methods. Both methods are successful in restoring all four images, but the data in Table 9 clearly show that the proposed method has a higher efficacy. Although the SGCS method has higher SNR values than the AHZP method in Figures 4 and 5, the AHZP method has higher SNR values than the SGCS method in Figures 6 and 7. Figure 4 shows that both methods have the same SSIM values. However, the SSIM values of the SGCS method in Figures (5 – 7) are higher than those of the AHZP method. These results demonstrate that the proposed method is effective at restoring blurred images.

5. Conclusion

Accelerated Hager-Zhang Projection Method for Convex Constrained Monotone Nonlinear Equations with application in image deblurring problems is presented in this paper. We achieved this by proposing

a new Hager-Zhang nonnegative parameter choice. Numerical comparisons were performed using large-scale test problems, and the AHZP method outperformed the DLPM [14] and MHZ2 [29] methods as shown in Tables 1-7 and Figures 1 – 3. Furthermore, the AHZP method is successfully applied to deal with the experiments on the ℓ_1-norm regularization problem in image restoration and compared its performance with the SGCS method [65]. The experiments were carried out on various samples of images (Figures 4 – 7), and the results are recorded in Table 9, which clearly show that the AHZP approach is very efficient. Future work includes modifying the proposed method to solve the inertial-based derivative-free approach for monotone nonlinear equations with Motion control application.

References

[1] Waziri, M.Y., Ahmad, K. and Sabiu, J.: A family of Hager–Zhang conjugate gradient methods for system of monotone nonlinear equations. Applied Mathematics and Computation, 2019; 361: 645–660.

[2] Waziri M.Y., Ahmad K. and Sabiu J.: A Dai–Liao conjugate gradient method via modified secant equation for system of nonlinear equations. Arabian Journal of Mathematics, https://doi.org/10.1007/s40065-019-0264-6

[3] Liu J.K., and Li S.J.: A projection method for convex constrained monotone nonlinear equations with applications, Comput. Math. Appl. 2015; 70(10): 2442–2453.

[4] Waziri M.Y., Ahmad K., Sabiu J., and Halilu, A.S.:Enhanced Dai–Liao conjugate gradient methods for systems of monotone nonlinear equations. SeMA Journal, 2020; 1-37.

[5] Andrei, N.: Open problems in conjugate gradient algorithms for unconstrained optimization. Bull. Malays. Math. Sci. Soc. 2011; 34(2): 319–330.

[6] Barzilai, J. and Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. 1988; 8: 141-148.

[7] Li, D. and Fukushima M.: A global and superlinear convergent Gauss-Newton based BFGS method for symmetric nonlinear equation. SIAM J Numer Anal. 1999; 37: 152-172.

[8] Halilu, A.S., Majumder, A., Waziri, M.Y., Abdullahi,H. Double direction and step length method for solving system of nonlinear equations, Euro. J. Mol. Clinic. Med. 2020; 7 (7): 3899–3913.

[9] Mohammad, H.: A descent derivative-free algorithm for nonlinear monotone equations with convex constraints Rairo-oper. Res.2020; 54: 489–505.

[10] Halilu, A.S., Majumder, A,. Waziri, M.Y., Ahmed, K., Signal recovery with convex constrained nonlinear monotone equations through conjugate gradient hybrid approach, Math. comput. Simulation, 2021, http://doi.org/10.1016/j.matcom.2021.03.020.

[11] Liu, D.Y., Shang, Y.F. A new Perry conjugate gradient method with the generalized conjugacy condition, Comput. Intel. Softw. Eng. (CiSE), 2010 International Conference on Issue Date: 2010; 10–12.

[12] Liu, D.Y., Xu, G.Q. A Perry descent conjugate gradient method with restricted spectrum. Optimization Online, Nonlinear Optimization (unconstrained optimization). 2011; 1–19.

[13] Abubakar, A.B. and Kumam P.: A descent Dai–Liao conjugate gradient method for nonlinear equations. Num. Algor. 2018; 1–14.

[14] Abubakar, A.B. and Kumam P., and Awwal A.M.: A Descent Dai-Liao Projection Method for Convex Constrained Nonlinear Monotone Equations with Applications. Thai J Math. 2018; 128–152.

[15] Babaie-Kafaki, S. Ghanbari, R.: A descent family of Dai–Liao conjugate gradient methods, Optim. Methods Softw. 2014; 29(3) : 583-591.

[16] Babaie-Kafaki, S. Ghanbari, R.: Two modified three-term conjugate gradient methods with sufficient descent property, Optim. Lett. 2014; 8(8) : 2285-2297.

[17] Babaie-Kafaki, S. Ghanbari, R.: Two optimal Dai-Liao conjugate gradient methods, Optimization 2015; 64(11) : 2277-2287.

[18] Andrei, N.: An adaptive conjugate gradient algorithm for large-scale unconstrained optimization, J. Comput. Appl. Math. 2016; 292(1) : 83-91.

[19] Andrei, N.: A Dai-Liao conjugate gradient algorithm with clustering of eigenvalues, Numer. Algor. 2017; 77 : 1273-1282.

[20] Andrei, N. Accelerated adaptive Perry conjugate gradient algorithms based on the self-scaling BFGS update, J. Comput. Appl. Math. 2017; 325: 149-164.

[21] Fatemi, M. A new efficient conjugate gradientmethod for unconstrained optimization, J. Comput. Appl. Math. 2016; 300(1) : 207-216.

[22] Fatemi, M. An optimal parameter for Dai-Liao family of conjugate gradient methods, J. Optim. Theory Appl. 2016; 169(2) : 587-605.

[23] Fatemi M, Babaie-Kafaki, S.: Two extensions of the Daia–Liao method with sufficient desent property based on a penalization scheme, Bull. Comput. Appl. Math, 2016; 4(1) : 7-19.

[24] Dai, Y.H. and Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. 2001; 43(1): 87–101.

[25] Waziri, M.Y., Ahmad, K. and Halilu, A.S.: Adaptive three-term family of conjugate residual methods for system of monotone nonlinear equations. São Paulo Journal of Mathematical Sciences, https://doi.org/10.1007/s40863-022-00293-0

[26] Ahmad, K., Waziri, M.Y., Halilu, A.S., Murtala, S., and Sabi'u, J. : Another Hager-Zhang-type method via singular-value study for constrained monotone equations with application. Numerical Algorithms, https://doi.org/10.1007/s11075-023-01678-8

[27] Ahmad, K., Waziri, M.Y., Murtala, S., Halilu, A.S., and Sabi'u, J. : On a Scaled Symmetric Dai–Liao-Type Scheme for Constrained System of Nonlinear Equations with Applications. Journal of Optimization Theory and Applications, https://doi.org/10.1007/s10957-023-02281-6

[28] Sabi'u, J., Shah, A., Waziri, M.Y.: Two optimal Hager-Zhang conjugate gradient methods for solving monotone nonlinear equations. Appl. Numer. Math. 2020; https://doi.org/10.1016/j.apnum.2020.02.017.
[29] Sabi'u, J., Shah, A., Waziri, M.Y., Ahmed, K.: Modified Hager-Zhang conjugate gradient methods via singular value analysis for solving monotone nonlinear equations with convex constraint, Int. J. Comput. Meth. 2020; https://doi:10.1142/S0219876220500437.
[30] Perry, A.: A modified conjugate gradient algorithm. Oper. Res. Tech. Notes. 1978; 26(6): 1073–1078.
[31] Solodov, M.V., and Svaiter, B.F.: A globally convergent inexact Newton method for systems of monotone equations. In Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Springer. 1999; 355-369.
[32] W.W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, SIAM. J. optim. 2005; 16 (1):170–192.
[33] Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**(1), 35-58(2006).
[34] Halilu A.S., and Waziri M.Y.: A transformed double step length method for solving large-scale systems of nonlinear equations. J Num Math Stoch. 2017; 9: 20-32.
[35] Halilu, A.S. and Waziri, M.Y.: An improved derivative-free method via double direction approach for solving systems of nonlinear equations, J of the Ramanujan Mathl. Soc. 2018; 33: 75-89.
[36] Mohammad, H.: A descent derivative-free algorithm for nonlinear monotone equations with convex constraints Rairo-oper. Res.2020; 54: 489–505.
[37] Aminifard, Z. and Babaie-Kafaki, S.: An optimal parameter choice for the Dai-Liao family of conjugate gradient methods by avoiding a direction of the maximum magnification by the search direction matrix, 4OR 2018; 1–14.
[38] Awwal, A.M, Wang, L, Kumam, P, and Mohammad, H.: A two-step spectral gradient Projection method for system of nonlinear monotone equations and image deblurring problems. Symmetry, 2020: 12: 874. doi:10.3390/sym12060874.
[39] Awwal, A.M, Kumam, P, Wang, L, Huang, S, and Kumam, W : Inertial-based derivative-free method for system of monotone nonlinear equations and application. IEEE, 2020: 1–10. doi: 10.1109/ACCESS.2020.3045493.
[40] Awwal, A.M, Kumam, P, Mohammad, H., Watthayu, W., Abubakar, A.B.: A Perry-type derivative-free algorithm for solving nonlinear system of equations and minimizing ℓ_1 regularized problem. Optimization, 2020; doi: 10.1080/02331934.2020.1808647.
[41] Halilu, A.S. Waziri, M.Y., Yusuf, I. Efficient matrix-free direction method with line search for solving large-scale system of nonlinear equations. Yugoslav Journal of Operations Research, 2020; 30(4): 399–412.
[42] Zarantonello, E.H. Solving functional equations by contractive averaging. Tech. Rep. 160. Madison, Wisconsin: U. S. Army Math. Research Center. 1960.
[43] Dennis J.E and Schnabel R.B.: Numerical Methods for Unconstrained Optimization and NonLinear Equations. Prentice Hall, Englewood Cliffs, NJ, 1983.
[44] Halilu A.S. and Waziri M.Y.: Enhanced matrix-free method via double step length approach for solving systems of nonlinear equations. Int J app Math Res. 2017; 6: 147-156.
[45] Minty, G. J. Monotone (nonlinear) operators in Hilbert space. Duke Math. J. 1962; 29(3):341–346.
[46] Kačurovskii, R. I. On monotone operators and convex functionals. Usp. Math. Nauk. 1960; 15(4):213–215.
[47] Waziri M.Y., Leong W.J., Hassan M.A. Jacobian-Free Diagonal Newton's Method for Solving Nonlinear Systems with Singular Jacobian. Malay J of Mathl. Sci 2011; 5: 241-255.
[48] Yuan G, Lu X. :A new backtracking inexact BFGS method for symmetric nonlinear equations. Comp Math App 2008; 55: 116-129.
[49] Sun, M., Tian, M.Y., Wang, Y.J.: Multi-step discrete-time Zhang neural networks with application to time-varying nonlinear optimization. Discrete Dyn. Nat. Soc. Article ID 4745759, 2019; 1–14.
[50] Meintjes, K., Morgan, A.P.: A methodology for solving chemical equilibrium systems. Appl. Math. Comput. 1987; 22: 333–361.
[51] Xiao, Y.H., Zhu, H.: A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing. J. Math. Anal. Appl. 2013; 405: 310–319.
[52] Halilu A.S., Dauda,M.K., Waziri, M.Y., and Mamat, M.: A derivative-free decent method via acceleration parameter for solving systems of nonlinear equations. Open J. sci.tech. 2019; 2(3): 1-4.
[53] Halilu, A.S., and Waziri, M.Y.: Inexact Double Step Length Method for Solving Systems of Nonlinear Equations. Stat., Optim. Inf. Comput., 2020; 8: 165–174.
[54] H. Abdullahi, A.S. Halilu, and Waziri M. Y. "A Modified Conjugate Gradient Method via a Double Direction Approach for solving large-scale Symmetric Nonlinear Systems",Journal of Numerical Mathematics and Stochastics, 2018; 10(1): 32-44.
[55] Halilu, A.S. and Waziri, M.Y.: Solving systems of nonlinear equations using improved double direction method, J of the Nigerian Mathl. Soc. 2020; 32(2): 287-301.
[56] Hestenes, M.R. and Stiefel, E.: Method of conjugate gradient for solving linear systems, J. Res. Nat. Bur. Stand. 1952; 49 : 409-436.
[57] Wang C, Wang Y, Xu C.: A projection method for a system of nonlinear monotone equations with convex constraints, Math. Methods Oper. Res. 2007; 66(1): 33-46.
[58] Wang CW, Wang YJ.: A superlinearly convergent projection method for constrained systems of nonlinear equations, J. Glob. Optim. 2009; 44(2): 283-296.
[59] Dolan, E. and Moré, J.:Benchmarking optimization software with performance profiles, Journal of Mathematical program, 2002; 91: 201-213.
[60] Elaine T., Wotao, Y., Yin Z.: A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. CAAM TR07-07, Rice University, 2007; 43-44.
[61] Pang, J.S.: Inexact Newton methods for the nonlinear complementarity problem, Math. Program. 1986; 1: 54-71.
[62] Mario A.T., Figueiredo, R., Nowak, D.: An EM algorithm for wavelet-based image restoration. IEEE Transactions on Image Processing, 2003; 12(8): 906-916.
[63] Figueiredo, M., Nowak, R., Wright, S.J.: Gradient projection for sparse reconstruction, application to compressed sensing and other inverse problems, IEEE J-STSP IEEE Press, Piscataway, NJ. 2007; 586-597.
[64] Tibshirani, R.:Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society,1996; 58(1):267–288.

[65] Xiao, Y., Wang, Q., Hu, Q.: Non-smooth equations based methods for $l1-$norm problems with applications to compressed sensing, Nonlinear Anal. Theory Methods Appl. 2011; 74: 3570–3577.
[66] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Trans, Image Process. 2004; 13: 600–612.