Computational and Applied Mathematics (2021) 40:239
https://doi.org/10.1007/s40314-021-01624-1

On solving double direction methods for convex constrained monotone nonlinear equations with image restoration

Abubakar Sani Halilu[1,2,6] · **Arunava Majumder**[1] · **Mohammed Yusuf Waziri**[3,6] · **Aliyu Muhammed Awwal**[4,5,6] · **Kabiru Ahmed**[3,6]

Received: 6 November 2020 / Revised: 20 April 2021 / Accepted: 16 August 2021

Abstract
Vast applications of derivative-free methods to restore the blurred images in compressive sensing has become an important trend in recent years. This research, a double direction method for better image restoration is proposed. Besides this, two double direction algorithms to solve constrained monotone nonlinear equations are presented. The main idea employed in the first algorithm is to approximate the Jacobian matrix via acceleration parameter to propose an effective derivative-free method. The second algorithm involve hybridizing the scheme of the first algorithm with Picard–Mann hybrid iterative scheme. In addition, the step length is calculated using inexact line search technique. The proposed methods are proven to be globally convergent under some mild conditions . The numerical experiment, shown in this paper, depicts the efficiency of the proposed methods. Furthermore, the second 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 previous literature.

Keywords Acceleration parameter · Derivative-free method · Image restoration · Jacobian matrix · Nonlinear monotone equations

✉ Abubakar Sani Halilu
abubakars.halilu@slu.edu.ng

Arunava Majumder
am.arunavamajumder@gmail.com, arunava.23440@lppu.co.in

1 Department of Mathematics, School of Chemical Engineering and Physical Sciences, Lovely Professional University, Phagwara 144411, India

2 Department of Mathematics and Computer Science, Sule Lamido University, Kafin Hausa, Nigeria

3 Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University, Kano, Nigeria

4 Department of Mathematics, Faculty of Science, Gombe State University, Gombe, Nigeria

5 KMUTT-Fixed Point Theory and Applications Research Group, Theoretical and Computational Science Center (TaCS), Science Laboratory Building, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khru, Bangkok 10140, Thailand

6 Numerical Optimization Research Group, Bayero University, Kano, Nigeria

Published online: 08 September 2021

Mathematics Subject Classification Primary 65K05; Secondary 90C30 · 90C53

1 Introduction

Nonlinear problems are of interest to scientists because most problems arising from engineering, biology, mathematics, physics, and many other branches of science, are inherently nonlinear in nature. The standard nonlinear system of equations is represented by

$$F(x) = 0, \tag{1.1}$$

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

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

$$\langle F(x) - F(y),\ x - y\rangle \geq 0. \tag{1.2}$$

Problem (1.1) is called monotone system of nonlinear equations if F satisfies (1.2).

The monotone mappings, which form a class of nonlinear equations were initially researched by Zarantonello (1960), Minty (1962) and Kačurovskii (1960), in Hilbert spaces. The studies of such mappings are practically applied in many scientific fields, like in the system of economic equilibrium (Sun et al. 2019) and chemical equilibrium (Meintjes and Morgan 1987). It has practical application in ℓ_1-norm regularization problem in signal and image recovery (Xiao and Zhu 2013; Halilu et al. 2021) as well. Some iterative approaches for solving these problems include derivative-free methods (Halilu and Waziri 2017a, b, 2020a; Halilu et al. 2019, 2020), Newton and quasi-Newton methods (Dennis and Schnabel 1983; Waziri et al. 2011; Yuan and Lu 2008; Li and Fukushima 1999) and their pertaining iterative procedure is given by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots, \tag{1.3}$$

where $s_k = x_{k+1} - x_k$ and α_k is a step length that is computed using any suitable line search. A line search is a technique that computes the step length α_k along the direction d_k via (1.3) to generate the sequence of iterates $\{x_k\}$ in order to achieve global convergence.

Moreover, the problem of nonlinear equations (1.1) is analogous to the following problem of unconstrained optimization

$$\min f(x), \quad x \in \mathbb{R}^n, \tag{1.4}$$

$f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be twice continuously differentiable (Li and Fukushima 1999). Fermat's extremum theorem stated that, if a point $x^* \in \mathbb{R}^n$ is the local minimizer of the unconstrained optimization problem (1.4) then problem (1.1) holds. Furthermore, let x^* be the minimizer of problem (1.4), then problem (1.1) is the first order necessary condition for the global optimization problem (1.4) with function F as its gradient.

Among the popular methods for finding the solution in (1.1), Newton's method stands out because of its nice properties (Dennis and Schnabel 1983), such as rapid convergence rate from a reasonably good starting point $x_0 \in \mathbb{R}^n$ in a neighborhood of the solution. Newton's search direction d_k is determined by solving the following linear system of equations,

$$F_k + F_k' d_k = 0,$$

where F_k' is the Jacobian matrix of F at x_k. However, in Newton's method, the derivative F' is computed at each iteration, which may be unavailable or could not be obtained exactly. In this case Newton's method cannot be applied directly. For this reason, quasi-Newton's methods were developed to replace the Jacobian matrix or its inverse with an approximation which can be updated at each iteration (Dennis and More 1974; Yuan and Lu 2008), and its search direction is given by

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

where B_k is $n \times n$ matrix that approximate the Jacobian of F at x_k. Furthermore, the most outstanding class of quasi-Newton update B_k needs to satisfy the secant equation

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

where $y_{k-1} = F_k - F(x_{k-1})$ and $s_{k-1} = x_k - x_{k-1}$.

Despite the appealing characteristics of the Newton and quasi-Newton's methods, the derivative F' or its approximation is computed at each iteration, so they are not ideal for solving large-scale problems. To overcome these problems, matrix-free methods are proposed. In Mohammad (2020), the author used Barzilai–Borwein (BB) like method to approximate B_k^{-1} with diagonal matrix (i.e. $B_k^{-1} \approx \tau_k I$), where I is an identity matrix and the search direction is given by

$$d_k = -\tau_k F_k,$$

with τ_k defined as

$$\tau_k = \frac{s_k^T s_k}{y_k^T s_k}. \tag{1.5}$$

Double direction scheme is another variant of matrix-free approach that has been proposed in Duranovic-Milicic (2008) with the iterative procedure presented as

$$x_{k+1} = x_k + \alpha_k d_k^i + \alpha_k^2 d_k^{ii}, \tag{1.6}$$

where x_{k+1} represents the new iterate, x_k is the previous iterate, α_k denotes the step length, while d_k^i and d_k^{ii} generate the search directions.

The concept of double direction technique was developed by Duranovic-Milicic (2008), using multi-step iteration to generate the iterates. Nonetheless, a multi-step algorithm for minimizing a non-differentiable function using double direction approach is presented by Duranovic-Milicic and Gardasevic-Filipovic (2010). Motivated by the work in Duranovic-Milicic (2008) and Duranovic-Milicic and Gardasevic-Filipovic (2010), Petrović and Stanimirovic (2014), proposed a double direction method for solving unconstrained optimization problems. The authors approximated the Hessian matrix $\nabla^2 f(x_k)$ via acceleration parameter $\gamma_k > 0$, i.e., $\nabla^2 f(x_k) \approx \gamma_k I$, where I is an identity matrix and $f : \mathbb{R}^n \to \mathbb{R}$. The proposed acceleration parameter is given as

$$\gamma_{k+1} = 2\frac{f(x_{k+1}) - f(x_k) - \alpha_k g_k^T(\alpha_k d_k - \gamma_k^{-1} g_k)}{(\alpha_k d_k - \gamma_k^{-1} g_k)^T(\alpha_k d_k - \gamma_k^{-1} g_k)},$$

where $g_k = \nabla f(x_k)$ and the step length α_k is computed using backtracking inexact line search technique. In addition, the sequence of iterates $\{x_k\}$ is generated using (1.6) that is converged to the solution linearly. The attractive feature of the method presented in Petrović and Stanimirovic (2014), is that, the two directions used are actually derivative-free and so the method is suitable for solving large-scale problems. Nevertheless, the study of double direction methods for solving system of nonlinear equations is rare in the literature, this motivated (Halilu and Waziri 2018) to use the double direction scheme in (1.6) and proposed a derivative-free method approach for solving system of nonlinear equations. In their work, the Jacobian matrix is approximated via acceleration parameter $\gamma_k > 0$, i.e., $F'_k \approx \gamma_k I$, where I is identity matrix and the acceleration parameter is calculated as

$$\gamma_{k+1} = \frac{y_k^T y_k}{(\alpha_k + \alpha_k^2 \gamma_k) y_k^T d_k}.$$

The two directions presented in Halilu and Waziri (2018), are clearly matrix-free. The first direction approximates the Newton's direction through acceleration parameter and the second one is defined as $d_k = -F_k$. However, the method is proved to be globally convergence after the assumption that, the Jacobian of the function F is bounded and positive definite. The rationale behind double direction method is that, there are two corrections in the scheme, therefore, if one correction fails during iterative process then the second one will correct the system. The performance of double direction scheme is further improved by Abdullahi et al. (2018), where they modified the idea in Halilu and Waziri (2018) based on conjugate gradient approach to solve symmetric nonlinear equations. The method converged globally using the inexact line search proposed by Li and Fukushima (1999). In order to improved the convergence properties of double direction method presented in Halilu and Waziri (2018), Halilu and Waziri also proposed matrix-free direction method that converges globally. This was made possible by making the two directions in (1.6) to be equal and proposed search direction given as $d_k = -(1 + \alpha_k)\gamma_k^{-1} F_k$, α_k and $\gamma_k > 0$ are respectively step length and acceleration parameter. Recently, Halilu and Waziri (2020b) solved the system of nonlinear equations by improving the double direction iteration in (1.6). The global convergence of the method was established under some mild conditions and the numerical experiments demonstrated in the paper showed that the proposed method is very efficient.

Moreover, the convergence properties and numerical results of the double direction method in Petrović and Stanimirovic (2014), is further improved by Petrović et al. (2018) by hybridizing the scheme with Picard–Mann hybrid iterative process proposed by Khan in Safeer (2013). The method in Petrović et al. (2018) generates a sequence of iterates $\{x_k\}$ such that

$$x_{k+1} - x_k = \eta {\alpha_k}^2 d_k - \eta \alpha_k \gamma_k^{-1} g_k,$$

where the correction parameter is defined as $\eta = (\eta_k + 1) \in (1, 2), \forall k$.

The Picard iterative process (Picard 1890) is defined by the sequence $\{u_k\}$ as

$$\begin{cases} u_1 = u \in \Omega, \\ u_{k+1} = T u_k, \quad k \in \mathbb{N}. \end{cases}$$

The Mann iterative process (Mann 1953) is defined by the sequence $\{v_k\}$ as

$$\begin{cases} v_1 = v \in \Omega, \\ v_{k+1} = (1 - \eta_k) v_k + \eta_k T v_k, \quad k \in \mathbb{N}. \end{cases}$$

where $\{\eta_k\} \in (0, 1)$ and the Ishikawa iterative process (Ishikawa 1974) is defined by the sequences $\{u_k\}$ and $\{v_k\}$ as

$$\begin{cases} v_1 = z \in \Omega, \\ u_k = (1 - \beta_k) v_k + \beta_k T v_k, \quad k \in \mathbb{N}, \\ v_{k+1} = (1 - \eta_k) v_k + \eta_k T u_k. \end{cases}$$

where $\{\beta_k\}$ and $\{\eta_k\}$ are the sequences of positive numbers which satisfy the conditions

- $0 \leq \eta_k \leq \beta_k \leq 1, k \geq 0.$
- $\lim_{k\to\infty} \beta_k = 0.$
- $\sum_{k=0}^{\infty} \eta_k \beta_k = \infty.$

The Picard–Mann hybrid iterative process is defined as three relations:

$$\begin{cases} x_1 = x \in \Omega, \\ v_k = (1-\eta_k)x_k + \eta_k T(x_k), \\ x_{k+1} = T(v_k), \quad k \in \mathbb{N}, \end{cases} \tag{1.7}$$

where $T : \Omega \longrightarrow \Omega$ is a mapping defined on nonempty convex subset Ω of a normed space $\mathbf{E}$, x_k and v_k are sequences determined by the iteration (1.7), and $\{\eta_k\}$ is the sequence of positive numbers in (0, 1).

Clearly, one can observe the following:

i. Ishikawa process is a double Mann iterative process.
ii. Picard–Mann process in (1.7) is a hybrid of Picard and Mann iterative schemes.

From the above literature, we can conclude that applying hybridization process is a good approach to improve the convergence properties and numerical experiments of some existing methods.

On the other hand, many researchers developed interest to solve monotone nonlinear equations with convex constraints using the projection method of Solodov and Svaiter (1999). For instance, Wang et al. (2007), incorporated the approach (Solodov and Svaiter 1999) and solved the monotone nonlinear equations with convex constraints, where the linear system of equations is approximately solved after the initialization in order to obtain a point of trial and then use the projection method to achieve the next iteration. The method in Wang et al. (2007) is proved to be globally convergent with linear rate of convergence. To increase the rate of convergence of the approach in Wang et al. (2007), its modification was developed by Wang and Wang (2009) with superlinear rate of convergence. Accordingly, Mohammad and Abubakar (2017) presented a decent method for solving monotone equations with convex constraints by modifying BB parameter in (1.5) as $\bar{\tau}_k = \frac{s_{k-1}^T s_{k-1}}{\gamma_{k-1}^T s_{k-1}}$, with $\gamma_{k-1} = F_k - F_{k-1} + r_{k-1}d_{k-1}$, and $r_{k-1} = 1 + \max\{0, \frac{y_{k-1}^T d_{k-1}}{\|F_{k-1}\|^2}\}$. Interestingly, it has been shown in their work that, $\gamma_{k-1}^T s_{k-1} > 0$ for all k, provided the solution of the problem in (1.1) is yet to be attained. Recently, a two step gradient projection method for solving nonlinear monotone equations is presented by Awwal et al. (2020). The proposed two-step algorithm uses two search directions which are defined using the modified BB spectral parameters which serves as the approximation of the Jacobian matrix with scalar multiple of identity matrix. The numerical results presented in their work have shown the efficiency of the proposed method over some existing methods in the literature. Furthermore, the method has been successfully applied to handle the problem arising in image deblurring.

Inspired by the contributions mentioned above, we aimed at developing a matrix-free method via double direction approach, that is globally convergent, for solving convex constrained system of monotone nonlinear equations, with application in image deblurring problems.

The followings are some of the contributions of this paper.

- This paper present some new iterative methods for solving convex constrained monotone nonlinear equations via the acceleration parameter.

Table 1 Author's contribution table

Author's name	Matrix-free	Double direction	Monotone	Global convergence	Application in compressing sensing
Solodov and Svaiter (1999)			✓	✓	
Li and Fukushima (1999)				✓	
Mohammad (2020)	✓		✓	✓	
Duranovic-Milicic (2008)	✓	✓		✓	
Duranovic-Milicic (2008)	✓	✓		✓	
Halilu et al. (2020)	✓	✓		✓	
Petrović and Stanimirovic (2014)		✓		✓	
Petrović et al. (2018)		✓		✓	
Abdullahi et al. (2018)	✓	✓		✓	
Halilu and Waziri (2020b)	✓	✓		✓	
Halilu and Waziri (2018)	✓	✓		✓	
Wang et al. (2007)	✓		✓	✓	
Wang and Wang (2009)	✓		✓	✓	
Xiao et al. (2011a)	✓		✓	✓	✓
Awwal et al. (2020)	✓		✓	✓	✓
This article	✓	✓	✓	✓	✓

- The new search directions are proposed in such away that they satisfy the decent condition.
- The correction parameter is derived using the Picard–Mann iterative schemes to improve the numerical performance of the proposed methods.
- The second method is successfully applied to handle the ℓ_1-norm regularization in image recovery problems.

The paper is organized as follows. In the next section, we will present the algorithms of the proposed methods. The convergence analysis of the proposed algorithms are reported in Sect. 3. Section 4 lists some numerical experiments and the application of the proposed methods in image deblurring problems. Section 5 is the article's conclusion.

2 Preliminaries and algorithms

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

$$P_\Omega[x] = \arg\min\{\|x - y\| : y \in \Omega\}. \tag{2.1}$$

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

$$\|P_\Omega[x] - P_\Omega[y]\| \leq \|x - y\|, \tag{2.2}$$

consequently, we have

$$\|P_\Omega[x] - y\| \leq \|x - y\|, \quad \forall y \in \Omega. \tag{2.3}$$

Now, to present the iterative scheme of our methods, we suggest the search directions d_k^i and d_k^{ii} in (1.6) to be, respectively, defined as

$$d_k^i = -\gamma_k^{-1} F_k, \tag{2.4}$$

and

$$d_k^{ii} = -F_k, \tag{2.5}$$

where $\gamma_k > 0$. By putting (2.4) and (2.5) into (1.6), we obtained

$$x_{k+1} = x_k - (\alpha_k + \alpha_k^2 \gamma_k)\gamma_k^{-1} F_k. \tag{2.6}$$

The main idea of our scheme is to derive the update of the acceleration parameter γ_k. This is made possible using the first order Taylor's expansion given by

$$F_{k+1} \approx F_k + F'(\xi)(x_{k+1} - x_k). \tag{2.7}$$

By multiplying (2.7) through by τ_k we have

$$\tau_k F_{k+1} \approx \tau_k F_k + \tau_k F'(\xi)(\alpha_k + \alpha_k^2 \gamma_k) d_k^i, \tag{2.8}$$

where τ_k is defined in (1.5) and the parameter ξ fulfills the conditions $\xi \in [x_k, x_{k+1}]$,

$$\xi = x_k + \delta(x_{k+1} - x_k), \quad 0 \leq \delta \leq 1. \tag{2.9}$$

Taking $\delta = 1$ in (2.9) and obtained $\xi = x_{k+1}$.

Therefore, we are interested to approximate the Jacobian matrix via

$$\tau_k F'(\xi) \approx \gamma_{k+1} I. \tag{2.10}$$

Now, from (2.8) and (2.10) its not difficult to verify the modified secant equation:

$$\gamma_{k+1}s_k = \tau_k y_k, \tag{2.11}$$

where $y_k = F_{k+1} - F_k$, $s_k = (\alpha_k + \alpha_k^2\gamma_k)d_k^i$.

By multiplying y_k^T to the both side of (2.11) the proposed acceleration parameter is defined as:

$$\gamma_{k+1} = \frac{\|s_k\|^2\|y_k\|^2}{(\alpha_k + \alpha_k^2\gamma_k)^2(y_k^T d_k)^2}. \tag{2.12}$$

From (2.4) and (2.6) we have our first iterative scheme as:

$$x_{k+1} = x_k + (\alpha_k + \alpha_k^2\gamma_k)d_k^i. \tag{2.13}$$

Algorithm 1: Derivative-free double direction method(DDDM)

Input: Given $x_0 \in \Omega$, $d_0^i = -F_0$, $\rho \in (0, 1)$, $\sigma > 0$, $\epsilon = 10^{-6}$, $\gamma_0 = 1$ and $\xi > 0$, set $k = 0$.

Step 1: Compute F_k. If $\|F_k)\| \leq \epsilon$ then stop, else go to Step 2.

Step 2: Let $\mu_k = (\alpha_k + \alpha_k^2\gamma_k)$ where, $\alpha_k = \xi\rho^{m_k}$, with m_k being the smallest nonnegative integer m such that

$$-F(x_k + \mu_k d_k^i)^T d_k^i \geq \sigma\mu_k\|d_k^i\|^2. \tag{2.14}$$

Step 3: Set $z_k = x_k + \mu_k d_k^i$.

Step 4: If $z_k \in \Omega$ and $\|F(z_k)\| \leq \epsilon$, stop, otherwise go to Step 5.

Step 5: Compute the next iterate by

$$x_{k+1} = P_\Omega[x_k - \lambda_k F(z_k)], \tag{2.15}$$

where, $\lambda_k = \dfrac{(x_k - z_k)^T F(z_k)}{\|F(z_k)\|^2}$.

Step 6: Determine $\gamma_{k+1} = \dfrac{\|s_k\|^2\|y_k\|^2}{(\alpha_k + \alpha_k^2\gamma_k)^2(y_k^T d_k^i)^2}$, where $s_k = z_k - x_k$ and $y_k = F(z_k) - F(x_k)$.

Step 7: Compute the search direction d_{k+1}^i using (2.4).

Step 8: Consider $k = k + 1$ and go to Step 2.

To present the hybrid type of derivative-free double direction method, the mapping T in (1.7) is redefined by $T(v_k) = v_k - (\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k$. By this definition and (1.7) we have

$$\begin{cases} x_1 = x \in \Omega, \\ v_k = (1 - \eta_k)x_k + \eta_k T(x_k) \\ \quad = (1 - \eta_k)x_k + \eta_k(x_k - (\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k) \\ \quad = x_k - \eta_k(\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k, \\ x_{k+1} = T(v_k) = v_k - (\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k, \quad k \in \mathbb{N}. \end{cases} \tag{2.16}$$

From the second and third equations in (2.16) we obtain the second iterative scheme,

$$x_{k+1} = x_k - t_k(\alpha_k + \alpha_k^2\gamma_k)\gamma_k^{-1}F_k, \tag{2.17}$$

where $t_k = (\eta_k + 1)$ and $\eta_k \in (0, 1)$ is a correction parameter. We however choose $\eta k = \eta \forall k$, so that $t = (\eta + 1) \in (1, 2)$. From (2.17), we can easily show that, the search direction is defined as

$$d_k = \begin{cases} -F_k, & \text{if} \quad k = 0, \\ -t\gamma_k^{-1} F_k, & \text{if} \quad k \geq 1. \end{cases} \tag{2.18}$$

Algorithm 2: Derivative-free double direction method(HDDM)

Input: Given $x_0 \in \Omega$, $\gamma_0 = 1$, $d_0 = -F_0$, $t \in (1, 2)$, $\rho \in (0, 1)$, $\sigma > 0$, $\epsilon = 10^{-6}$, and $\xi > 0$, set $k = 0$.

Step 1: Compute F_k. If $\|F_k\| \leq \epsilon$ then stop, else go to Step 2.

Step 2: Let $\mu_k = (\alpha_k + \alpha_k^2 \gamma_k)$ where, $\alpha_k = \xi \rho^{m_k}$, with m_k being the smallest nonnegative integer m such that

$$-F(x_k + \mu_k d_k)^T d_k \geq \sigma \mu_k \|d_k\|^2. \tag{2.19}$$

Step 3: Set $z_k = x_k + \mu_k d_k$.

Step 4: If $z_k \in \Omega$ and $\|F(z_k)\| \leq \epsilon$, stop, otherwise go to Step 5.

Step 5: Compute the next iterate by

$$x_{k+1} = P_\Omega[x_k - \lambda_k F(z_k)], \tag{2.20}$$

where, $\lambda_k = \dfrac{(x_k - z_k)^T F(z_k)}{\|F(z_k)\|^2}$.

Step 6: $\gamma_{k+1} = \dfrac{\|s_k\|^2 \|y_k\|^2}{(\alpha_k + \alpha_k^2 \gamma_k)^2 (y_k^T d_k)^2}$.

Step 7: Compute the search direction $d_{k+1} = -t\gamma_{k+1}^{-1} F_{k+1}$.

Step 8: Consider $k = k + 1$ and go to Step 2.

3 Convergence analysis

In this section, the analysis of the global convergence of HDDM algorithm is presented. We consider the following assumptions:

Assumption 3.1 (A1) The mapping F is Lipschitz continuous, namely there exists a positive constant L such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{3.1}$$

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

$$(x - y)^T (F(x) - F(y)) \geq c\|x - y\|^2, \quad \forall x, y \in \mathbb{R}^n. \tag{3.2}$$

(A3) The solution set of (1.1) is nonempty and is denoted by S^x.

Remark 3.2 We give the following remarks

(a) From (3.2), we have

$$y_{k-1}^T s_{k-1} = s_{k-1}^T (F(z_{k-1}) - F(x_{k-1})) \geq c\|s_{k-1}\|^2 > 0. \tag{3.3}$$

From (3.1) and Cauchy–Schwarz inequality, we have

$$y_{k-1}^T s_{k-1} = (F(z_{k-1}) - F(x_{k-1}))^T (z_{k-1} - x_{k-1}) \leq L\|s_{k-1}\|^2. \tag{3.4}$$

Also, from (3.3) and (3.4) it is simple to check that, $L\|s_{k-1}\|^2 \geq y_{k-1}^T s_{k-1} \geq c\|s_{k-1}\|^2$, this implies that $\frac{\|s_{k-1}\|^2}{y_{k-1}^T s_{k-1}} \leq \frac{1}{c}$ and $\frac{\|s_{k-1}\|^2}{y_{k-1}^T s_{k-1}} \geq \frac{1}{L}$. This means τ_k in (1.5) is well defined. Therefore, we have

$$\frac{1}{L} \leq \tau_k \leq \frac{1}{c}. \tag{3.5}$$

(b) From assumption (A1), we have

$$\|y_{k-1}\| = \|F(z_{k-1}) - F(x_{k-1})\| \leq L\|s_{k-1}\| = L\mu_{k-1}\|d_{k-1}\|. \tag{3.6}$$

(c) From assumption (A2), we have

$$y_{k-1}^T d_{k-1} = (F(z_{k-1}) - F(x_{k-1}))^T \frac{s_{k-1}}{\mu_{k-1}} \geq \frac{c\|s_{k-1}\|^2}{\mu_{k-1}} = c\mu_{k-1}\|d_{k-1}\|^2 > 0. \tag{3.7}$$

Since $y_{k-1}^T d_{k-1} > 0$, the proposed γ_k in (2.12) is well defined.

(d) From assumption (A1) and (A2), we have

$$\gamma_k = \frac{\|y_{k-1}\|^2\|s_{k-1}\|^2}{(y_{k-1}^T s_{k-1})^2} \leq \frac{L^2\|s_{k-1}\|^4}{c^2\|s_{k-1}\|^4} = \frac{L^2}{c^2}. \tag{3.8}$$

Lemma 3.3 *Suppose Assumptions (A1)–(A3) hold and $\{x_k\}$ be generated by HDDM Algorithm, then the search direction d_k fulfills the descent property, i.e.,*

$$F_k^T d_k < 0. \tag{3.9}$$

Proof For $k = 0$, from (2.18) we have $F_0^T d_0 = -\|F_0\|^2 < 0$. For $k \geq 1$ since $t \in (1, 2)$, from (2.12) and (2.18), we have

$$F_k^T d_k = -\frac{t(y_k^T s_{k-1})^2}{\|y_{k-1}\|^2\|s_{k-1}\|^2}\|F_k\|^2 < 0. \tag{3.10}$$

□

Lemma 3.4 *Suppose Assumptions (A1)–(A3) hold. Then there exists a step length $\mu_k = (\alpha_k + \alpha_k^2\gamma_k)$ satisfying*

$$-F(x_k + \mu_k d_k)^T d_k \geq \sigma\mu_k\|d_k\|^2, \tag{3.11}$$

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

$$-F(x_{k_0} + (\xi\rho^{m_{k_0}} + (\xi\rho^{m_{k_0}})^2\gamma_{k_0})d_{k_0})^T d_{k_0} < \sigma(\xi\rho^{m_{k_0}} + (\xi\rho^{m_{k_0}})^2\gamma_{k_0})\|d_{k_0}\|^2. \tag{3.12}$$

Since $\rho \in (0, 1)$, using assumption $(A2)$, (3.11) and letting $m \to \infty$, we get

$$-F(x_{k_0})^T d_{k_0} \leq 0. \tag{3.13}$$

which clearly contradicts (3.10). 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 HDDM Algorithm, then $\{x_k\}$ and $\{z_k\}$ are bounded. In addition, we have*

$$\|d_k\| \leq M. \tag{3.14}$$

$$\lim_{k\to\infty} \|x_k - z_k\| = 0. \tag{3.15}$$

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{3.16}$$

Proof First, we show the boundedness of the sequences $\{x_k\}$ and $\{z_k\}$. Let $\bar{x} \in S^x$ be any solution of (1.1). Then by monotonicity of F we can write

$$(x_k - \bar{x})^T F(z_k) \geq (x_k - z_k)^T F(z_k). \tag{3.17}$$

Using the line search condition (3.11) and definition of z_k, we have

$$(x_k - z_k)^T F(z_k) \geq \sigma \mu_k^2 \|d_k\|^2 > 0. \tag{3.18}$$

Also, using (2.3) and fact that $x_{k+1} = P_\Omega[x_k - \lambda_k F(z_k)]$, we have

$$\begin{aligned}
\|x_{k+1} - \bar{x}\|^2 &= \|P_\Omega(x_k - \lambda_k F(z_k)) - \bar{x}\|^2 \\
&\leq \|x_k - \lambda_k F(z_k) - \bar{x}\|^2 \\
&= \|x_k - \bar{x}\|^2 - 2\lambda_k (x_k - \bar{x})^T F(z_k) + \lambda_k^2 \|F(z_k)\|^2 \\
&\leq \|x_k - \bar{x}\|^2 - 2\lambda_k (x_k - z_k)^T F(z_k) + \lambda_k^2 \|F(z_k)\|^2 \\
&= \|x_k - \bar{x}\|^2 - \frac{((x_k - z_k)^T F(z_k))^2}{\|F(z_k)\|^2} \\
&\leq \|x_k - \bar{x}\|^2,
\end{aligned} \tag{3.19}$$

for which we obtain

$$\|x_{k+1} - \bar{x}\| \leq \|x_k - \bar{x}\|. \tag{3.20}$$

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 (3.20) we have

$$\|F_k\| = \|F_k - F(\bar{x})\| \leq L\|x_k - \bar{x}\| \leq L\|x_0 - \bar{x}\|. \tag{3.21}$$

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

$$\|F_k\| \leq m_1. \tag{3.22}$$

For $k = 0$, by (2.18) and (3.22), we have $\|d_0\| = -\|F_0\| < m_1$. For $k \geq 1$, from (2.18) (3.22) and remarks (a)–(d) we have

$$\|d_k\| = \left\| \frac{-t(y_{k-1}^T s_{k-1})^2 F_k}{\|y_{k-1}\|^2 \|s_{k-1}\|^2} \right\| \leq \frac{t\|y_{k-1}\|^2 \|s_{k-1}\|^2 \|F_k\|}{\|y_{k-1}\|^2 \|s_{k-1}\|^2} \leq t\|F_k\| \leq tm_1.$$

Taking $M = tm_1$, we have (3.14).

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 (3.21), there exists some constants, say $\kappa > 0$, such that

$$\|F(z_k)\| \leq \kappa. \tag{3.23}$$

Now, from (3.19), we have

$$((x_k - z_k)^T F(z_k))^2 \leq \|F(z_k)\|^2(\|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2). \tag{3.24}$$

From the linesearch (2.19), we have

$$\sigma^2 \mu_k^4 \|d_k\|^4 \leq \mu_k^2 (F(z_k)^T d_k)^2. \tag{3.25}$$

Combining (3.19) and (3.25), it holds

$$\sigma^2 \mu_k^4 \|d_k\|^4 \leq \|F(z_k)\|^2(\|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2). \tag{3.26}$$

Since the sequence $\{\|x_k - \bar{x}\|\}$ is convergent and $\{F(z_k)\}$ is bounded, taking limit on both sides of (3.26) yields

$$\sigma^2 \lim_{k \to \infty} \mu_k^4 \|d_k\|^4 \leq 0,$$

and hence it holds that

$$\lim_{k \to \infty} \mu_k \|d_k\| = 0. \tag{3.27}$$

Combining (3.26) with the definition of z_k implies (3.15) holds. On the other hand, from (2.3) and the definition of λ_k, we have

$$\begin{aligned} \|x_{k+1} - x_k\| &= \|P_\Omega(x_k - \lambda_k F(z_k)) - x_k\| \\ &\leq \|x_k - \lambda_k F(z_k) - x_k\| \\ &= \|\lambda_k F(z_k)\| \\ &\leq \|x_k - z_k\|. \end{aligned} \tag{3.28}$$

This implies (3.16) . □

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

$$\liminf_{k \to \infty} \|F_k\| = 0. \tag{3.29}$$

Proof For the sake of contradiction, suppose that (3.29) is not true, then there exists $n_0 > 0$ such that

$$\|F_k\| \geq \delta_0 \quad \text{holds}, \quad \forall k > 0. \tag{3.30}$$

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

$$\|d_k\| \geq \delta_1. \tag{3.31}$$

If $\mu_k \neq \xi$, then by the definition of $\mu_k = (\alpha_k + \alpha_k^2 \gamma_k)$, $\rho^{-1}\mu_k$ does not satisfies (3.11), i.e.,

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

Now, combining with (3.10) gives

$$\begin{aligned} \frac{t(y_{k-1}^T s_{k-1})^2}{\|y_{k-1}\|^2 \|s_{k-1}\|^2} \|F(x_k)\|^2 &= -F_k^T d_k, \\ &= (F(x_k + \rho^{-1}\mu_k d_k) - F(x_k))^T d_k - F(x_k + \rho^{-1}\mu_k d_k)^T d_k, \end{aligned}$$

$$\leq L\rho^{-1}\mu_k\|d_k\|^2 + \sigma\rho^{-1}\mu_k\|d_k\|^2.$$
$$= \mu_k\|d_k\|(L+\sigma)\rho^{-1}\|d_k\|. \tag{3.32}$$

Therefore, from (3.32), we have

$$\begin{aligned}
\mu_k\|d_k\| &\geq \frac{t(y_{k-1}^T s_{k-1})^2}{\|y_{k-1}\|^2\|s_{k-1}\|^2}\frac{\rho\|F_k\|^2}{(L+\sigma)\|d_k\|} \\
&\geq \frac{tc^2\|s_{k-1}\|^4}{\|y_{k-1}\|^2\|s_{k-1}\|^2}\frac{\rho\|F_k\|^2}{(L+\sigma)\|d_k\|} \\
&\geq \frac{tc^2\|s_{k-1}\|^4}{L^2\|s_{k-1}\|^4}\frac{\rho\|F_k\|^2}{(L+\sigma)\|d_k\|} \\
&\geq \frac{tc^2}{L^2}\frac{\rho\delta_0^2}{(L+\sigma)M}.
\end{aligned} \tag{3.33}$$

The second and third inequalities follow from (3.3) and (3.7), respectively. The last inequality follows from (3.14) and (3.30).

The inequality (3.33) contradicts (3.27). Therefore (3.29) holds. Hence, the proof is complete. □

Remark 3.7 If the correction parameter $\eta = 0$, then the convergence result of DDDM algorithm follows.

4 Numerical experiments

Some numerical results are provided in the first part of this section, to show the effectiveness of our methods, i.e.,

- Algorithm 2.1: Derivative-free double direction method (DDDM).
- Algorithm 2.2: Hybrid derivative-free double direction method (HDDM).

Since the proposed algorithms are derivative-free, therefore their performances are compared with the following derivative-free method existing in the literature.

- A descent derivative-free algorithm for nonlinear monotone equations with convex constraints (DDPM) (Mohammad 2020).

In the second part, HDDM algorithm is applied to solve ℓ_1-norm regularization problem in image deblurring problems. The computer 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.

4.1 Numerical results

When implementing the algorithms in this experiments, the following parameters are set; $\xi = 1$, $\sigma = 10^{-4}$ and $\rho = 0.9$, we however set $t = 1.2$, in HDDM Algorithm. The parameters of DDPM algorithm, are taken as in Mohammad (2020). The iteration is set to stop for all the three methods if the following conditions occur: (i) when $\|F(x_k)\| \leq 10^{-6}$, (ii) when $\|F(z_k)\| \leq 10^{-6}$, or (iii) when the iterations exceed 1000 but no point of x_k satisfying the stopping criterion is obtained. We have tried the three methods on the previous six test problems with different initial guess and dimension (n values). To show the extensive

Table 2 Starting points used to test problems

Initial points (IP)	Values
$x1$	$(1, 1, \ldots, 1)^T$
$x2$	$\left(\frac{1}{2}, \frac{1}{2^2}, \ldots, \frac{1}{2^n}\right)^T$
$x3$	$\left(0, \frac{1}{2}, \frac{2}{3}, \ldots, 1 - \frac{1}{n}\right)^T$
$x4$	$\left(1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{n}\right)^T$
$x5$	$(2, 2, \ldots, 2)^T$
$x6$	$\left(\frac{1}{4}, \frac{-1}{4}, \ldots, \frac{(-1)^n}{4}\right)^T$

numerical experiments of DDDM, HDDM and DDPM methods, the experimentation was carried out with three different dimensions namely, 1000, 50,000 and 100,000. The following initial points were used for the test problems:

The following test problems were used in the experiments:

Problem 1 Awwal et al. (2020)
$F_1(x) = e^{x_1} - 1,$
$F_i(x) = e^{x_i} + x_{i-1} - 1, \quad i = 2, 3, \ldots, n-1,$
where $\Omega = \mathbb{R}^n_+$.

Problem 2 Mohammad (2020)
$F_i(x) = log(x_i + 1) - \frac{x_i}{n}, \quad i = 2, 3, \ldots, n-1,$
where $\Omega = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq n, x_i > -1, \quad i = 1, 2, \ldots, n\}$.

Problem 3 Awwal et al. (2020)
$F_i(x) = 2x_i - \sin |x_i|, \quad i = 1, 2, \ldots, n,$
where $\Omega = \mathbb{R}^n_+$.
Clearly, Problem 3 is nonsmooth at $x = 0$.

Problem 4 Mohammad (2020)
$F_i(x) = e^{x_i} - 1, \quad i = 1, 2, \ldots, n,$
where $\Omega = \mathbb{R}^n_+$.

Problem 5 Awwal et al. (2020)
$F_i(x) = 2x_i - \sin |x_i - 1|, \quad i = 1, 2, \ldots, n,$
where $\Omega = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq n, x_i > -1, \quad i = 1, 2, \ldots, n\}$.
Clearly, Problem 5 is nonsmooth at $x = 1$.

Problem 6 Awwal et al. (2020)
$F_1(x) = x_1 - e^{\cos(h(x_1 + x_2))},$
$F_i(x) = x_i - e^{\cos(h(x_{i-1} + x_i + x_{i+1}))}, \quad i = 2, 3, \ldots, n-1,$
$F_n(x) = x_n - e^{\cos(h(x_{n-1} + x_n))},$
where $h = \frac{1}{n+1}$ and $\Omega = \mathbb{R}^n_+$.

The reported numerical results of the three (3) methods are shown in Tables 3, 4, 5, 6, 7 and 8. From the Tables, "Iter" represents the total number of iterations, "Time" represents the CPU time (in seconds), "IP" represents the initial points and "FVAL" represents the functions

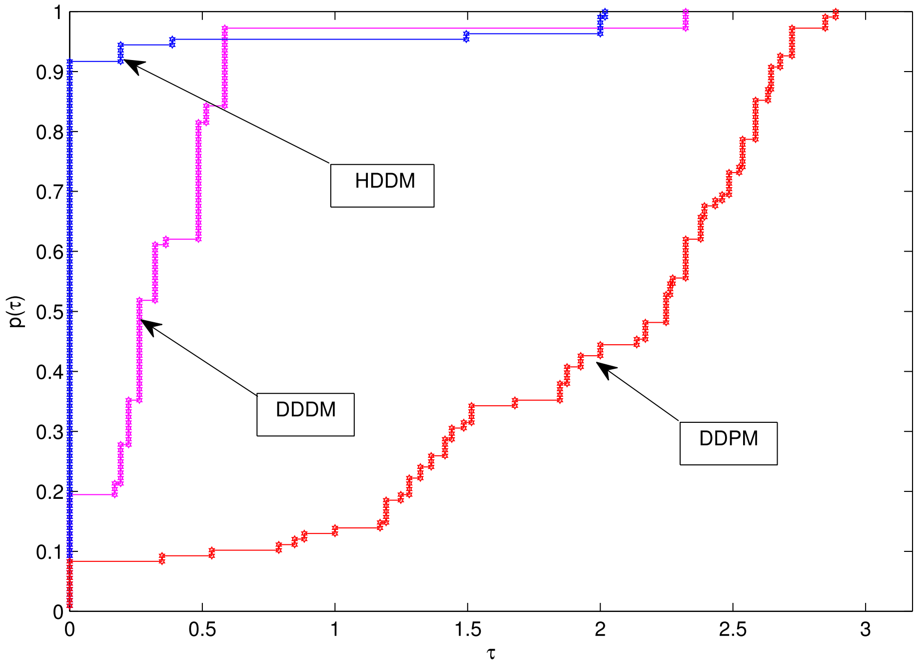

Fig. 1 Performance profile for the number of iterations

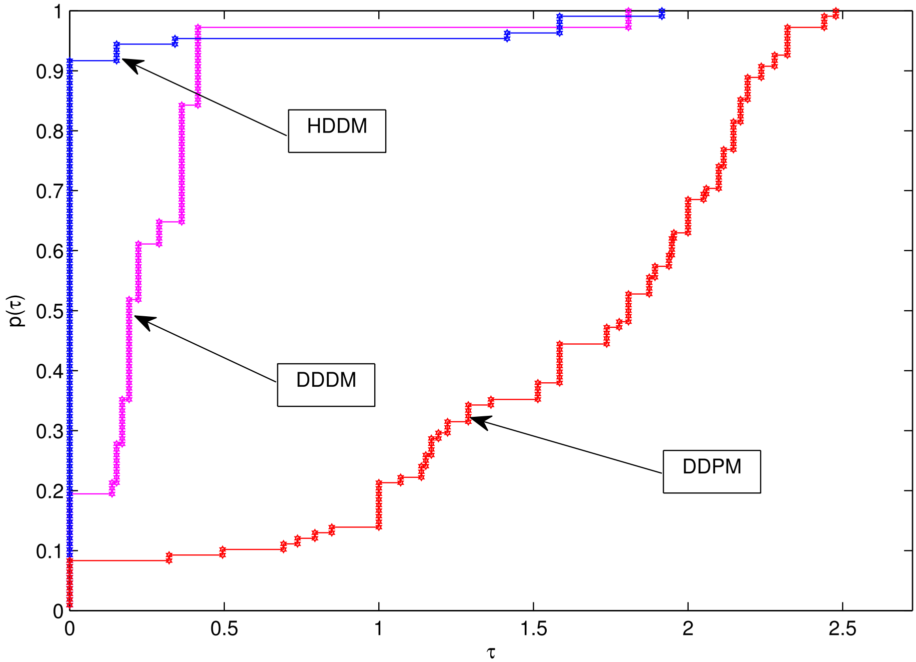

Fig. 2 Performance profile for the functions evaluations

Table 3 Numerical results of DDDM, HDDM and DDPM methods for problem 1

Dimension	IP	DDDM				HDDM				DDPM			
		Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm
1000	x1	6	8	0.01579	1.14E−07	4	6	0.021898	7.32E−07	16	18	0.013754	3.57E−07
	x2	9	10	0.017381	7.65E−08	5	7	0.012606	4.74E−07	17	18	0.008548	4.51E−07
	x3	7	9	0.015226	1.69E−07	5	7	0.015344	5.46E−07	16	18	0.011436	3.58E−07
	x4	8	10	0.015319	8.82E−08	7	9	0.016014	1.97E−08	16	18	0.009857	5.85E−07
	x5	6	8	0.012441	1.41E−07	4	6	0.0137	7.15E−07	16	18	0.01057	7.28E−07
	x6	5	7	0.011342	2E−07	4	6	0.009925	6.42E−08	1	2	0.003141	0
50,000	x1	6	8	0.234366	3.56E−07	4	6	0.194617	5.71E−08	19	21	0.242217	8.67E−07
	x2	9	10	0.283763	7.65E−08	5	7	0.181244	4.74E−07	17	18	0.19226	4.51E−07
	x3	6	8	0.241063	3.8E−07	4	6	0.210873	3.96E−07	19	21	0.247919	8.66E−07
	x4	8	10	0.265288	8.81E−08	7	9	0.24463	1.97E−08	16	18	0.199716	5.85E−07
	x5	7	9	0.249878	6.29E−08	5	7	0.204272	5.25E−08	22	24	0.285129	6.24E−07
	x6	5	7	0.13476	1.98E−07	4	6	0.130291	6.11E−08	1	2	0.027491	0
100,000	x1	6	8	0.48908	4.55E−07	4	6	0.340777	7.89E−08	21	23	0.478644	4.22E−07
	x2	9	10	0.499119	7.65E−08	5	7	0.344758	4.74E−07	17	18	0.343464	4.51E−07
	x3	6	8	0.425237	4.74E−07	4	6	0.321529	2.81E−07	21	23	0.480251	4.22E−07
	x4	8	10	0.537506	8.81E−08	7	9	0.5049	1.97E−08	16	18	0.380719	5.85E−07
	x5	7	9	0.599913	8.9E−08	5	7	0.386616	7.43E−08	24	26	0.598374	5.53E−07
	x6	5	7	0.248611	1.98E−07	4	6	0.219518	6.11E−08	1	2	0.051124	0

Table 4 Numerical results of DDDM, HDDM and DDPM methods for problem 2

Dimension	IP	DDDM				HDDM				DDPM			
		Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm
1000	x1	6	8	0.035927	4.76E−07	5	7	0.008075	1.4E−07	24	26	0.020136	5.08E−07
	x2	5	7	0.013451	9.65E−07	5	7	0.010812	9.2E−08	18	20	0.014231	6.32E−07
	x3	13	15	0.021244	5.11E−07	16	18	0.029107	2.45E−08	24	26	0.021171	5.03E−07
	x4	7	9	0.01367	2.3E−07	7	9	0.011141	4.5E−07	20	22	0.016412	5.2E−07
	x5	6	8	0.010638	4.59E−07	5	7	0.011358	1.35E−07	25	27	0.019453	7.65E−07
	x6	5	7	0.012076	3.97E−07	4	6	0.011281	1.19E−07	22	24	0.02353	7.68E−07
50,000	x1	7	9	0.243708	1.86E−07	5	7	0.181388	8.25E−07	28	30	0.494609	5.79E−07
	x2	5	7	0.156054	9.31E−07	5	7	0.170949	9.13E−08	18	20	0.319672	6.29E−07
	x3	18	20	0.551114	2.99E−07	88	90	2.626965	1.53E−07	28	30	0.541877	5.79E−07
	x4	7	9	0.212973	2.25E−07	7	9	0.23127	7.61E−07	20	22	0.352528	5.18E−07
	x5	7	9	0.215001	1.75E−07	5	7	0.18158	7.64E−07	29	31	0.528937	7.28E−07
	x6	6	8	0.211737	1.57E−07	4	6	0.16431	7.01E−07	25	27	0.458729	6.71E−07
100,000	x1	7	9	0.442629	2.63E−07	6	8	0.504699	2.72E−08	29	31	0.97942	6.35E−07
	x2	5	7	0.314231	9.31E−07	5	7	0.369069	9.13E−08	18	20	0.588916	6.29E−07
	x3	18	20	1.001162	2.16E−07	75	77	4.725202	3.04E−07	29	31	0.99589	6.34E−07
	x4	7	9	0.39876	2.25E−07	7	9	0.434612	7.66E−07	20	22	0.658945	5.18E−07
	x5	7	9	0.419536	2.46E−07	6	8	0.410346	2.52E−08	30	32	1.029389	5.14E−07
	x6	6	8	0.3679	2.21E−07	4	6	0.304972	9.89E−07	25	27	0.86033	9.49E−07

Table 5 Numerical results of DDDM, HDDM and DDPM methods for problem 3

Dimension	IP	DDDM				HDDM				DDPM			
		Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm
1000	x1	6	8	0.015569	9.07E−08	4	6	0.009941	5.25E−07	24	26	0.014296	7.56E−07
	x2	5	7	0.007412	1.51E−07	4	6	0.007669	4.34E−08	19	21	0.011837	5.31E−07
	x3	6	8	0.011093	1.11E−07	4	6	0.008335	6.63E−07	24	26	0.014005	7.54E−07
	x4	4	6	0.00622	5.3E−07	4	6	0.00931	7.72E−07	20	22	0.009989	5.75E−07
	x5	6	8	0.010519	1.51E−07	4	6	0.008023	8.52E−07	23	25	0.015776	7.62E−07
	x6	1	2	0.00584	0	1	2	0.004187	0	1	2	0.003839	0
50,000	x1	6	8	0.14953	6.42E−07	5	7	0.144335	8.67E−08	28	30	0.310846	7.61E−07
	x2	5	7	0.147519	1.51E−07	4	6	0.112941	4.34E−08	19	21	0.215895	5.31E−07
	x3	6	8	0.154206	6.45E−07	5	7	0.147688	8.73E−08	28	30	0.34001	7.61E−07
	x4	4	6	0.122533	5.25E−07	4	6	0.119342	7.72E−07	20	22	0.225217	5.76E−07
	x5	7	9	0.187996	6.51E−08	5	7	0.146349	1.41E−07	31	33	0.357249	6.27E−07
	x6	1	2	0.049631	0	1	2	0.049287	0	1	2	0.032177	0
100,000	x1	6	8	0.28952	9.07E−07	5	7	0.257853	1.23E−07	29	31	0.628223	7.94E−07
	x2	5	7	0.220009	1.51E−07	4	6	0.241617	4.34E−08	19	21	0.406173	5.31E−07
	x3	6	8	0.275188	9.1E−07	5	7	0.255319	1.23E−07	29	31	0.606163	7.94E−07
	x4	4	6	0.192486	5.25E−07	4	6	0.224371	7.72E−07	20	22	0.412978	5.76E−07
	x5	7	9	0.29537	9.21E−08	5	7	0.259356	1.99E−07	33	35	0.789034	7.38E−07
	x6	1	2	0.087854	0	1	2	0.098928	0	1	2	0.057458	0

Table 6 Numerical results of DDDM, HDDM and DDPM methods for problem 4

Dimension	IP	DDDM				HDDM				DDPM			
		Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm
1000	x1	5	7	0.013752	7.83E−07	5	7	0.008646	6.87E−08	25	27	0.012583	5.99E−07
	x2	5	7	0.007757	2.22E−07	4	6	0.008513	9.53E−08	18	20	0.008533	9.22E−07
	x3	7	9	0.006986	4.87E−07	5	7	0.008774	2.43E−07	25	27	0.01383	5.99E−07
	x4	6	8	0.011449	1.13E−07	7	9	0.012199	1.12E−07	22	24	0.016128	6.5E−07
	x5	6	8	0.012186	4.38E−07	5	7	0.008843	8.4E−08	26	28	0.013009	5.98E−07
	x6	1	2	0.004079	0	1	2	0.004508	0	1	2	0.002897	0
50,000	x1	6	8	0.128542	3.37E−07	5	7	0.123549	4.85E−07	29	31	0.292743	5.1E−07
	x2	5	7	0.106402	2.22E−07	4	6	0.097253	9.53E−08	18	20	0.180574	9.22E−07
	x3	6	8	0.134465	4.31E−07	5	7	0.127108	5.23E−07	29	31	0.318688	5.1E−07
	x4	6	8	0.1664	1.13E−07	7	9	0.192778	1.11E−07	22	24	0.225365	6.5E−07
	x5	7	9	0.152226	1.88E−07	5	7	0.136408	5.94E−07	31	33	0.2956	9.63E−07
	x6	1	2	0.02421	0	1	2	0.0219	0	1	2	0.015788	0
100,000	x1	6	8	0.25947	4.76E−07	5	7	0.243199	6.87E−07	30	32	0.543538	5.07E−07
	x2	5	7	0.191936	2.22E−07	4	6	0.177432	9.53E−08	18	20	0.309274	9.22E−07
	x3	6	8	0.223481	5.48E−07	5	7	0.256258	7.15E−07	30	32	0.568598	5.07E−07
	x4	6	8	0.223235	1.13E−07	7	9	0.293253	1.11E−07	22	24	0.372303	6.5E−07
	x5	7	9	0.281456	2.66E−07	5	7	0.240241	8.4E−07	33	35	0.659843	8.31E−07
	x6	1	2	0.038523	0	1	2	0.038031	0	1	2	0.028365	0

Table 7 Numerical results of DDDM, HDDM and DDPM methods for problem 5

Dimension	IP	DDDM				HDDM				DDPM			
		Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm
1000	x1	7	9	0.016248	8.05E−07	6	8	0.012379	3.59E−07	16	18	0.01181	7.39E−07
	x2	7	9	0.017203	8.13E−07	6	8	0.012694	4.1E−07	16	18	0.012994	8.26E−07
	x3	8	10	0.018875	1.9E−07	7	9	0.014774	2.79E−07	16	18	0.011449	7.31E−07
	x4	9	11	0.017352	2.53E−07	8	10	0.016785	9.87E−08	16	18	0.012042	8.17E−07
	x5	7	9	0.012254	9.93E−07	6	8	0.011468	4.4E−07	16	18	0.013446	6.79E−07
	x6	6	8	0.013715	1.26E−07	5	7	0.011815	9.4E−08	14	16	0.00959	5.51E−07
50,000	x1	8	10	0.279344	6.81E−07	7	9	0.287708	1.87E−07	18	20	0.276878	6.33E−07
	x2	8	10	0.282651	6.69E−07	7	9	0.273813	1.94E−07	18	20	0.330285	7.09E−07
	x3	8	10	0.279367	7.52E−07	8	10	0.314475	9.55E−08	18	20	0.297498	6.33E−07
	x4	10	12	0.379323	6.93E−07	10	12	0.360856	1.02E−07	18	20	0.294785	7.09E−07
	x5	8	10	0.271348	8.39E−07	7	9	0.264383	2.29E−07	20	22	0.316913	3.5E−07
	x6	6	8	0.223637	8.92E−07	5	7	0.248618	6.64E−07	19	21	0.296125	3.88E−07
100,000	x1	8	10	0.599236	9.63E−07	7	9	0.503587	2.65E−07	19	21	0.590877	7.35E−07
	x2	8	10	0.556636	9.45E−07	7	9	0.549131	2.74E−07	19	21	0.555972	3.49E−07
	x3	9	11	0.631565	1.22E−07	8	10	0.620913	1.52E−07	19	21	0.556349	7.35E−07
	x4	10	12	0.683173	9.98E−07	13	15	0.905601	1.32E−07	19	21	0.595335	3.49E−07
	x5	9	11	0.590531	1.42E−07	7	9	0.482389	3.24E−07	20	22	0.596198	4.95E−07
	x6	7	9	0.478583	1.51E−07	5	7	0.428161	9.4E−07	19	21	0.567323	9.66E−07

Table 8 Numerical results of DDDM, HDDM and DDPM methods for problem 6

Dimension	IP	DDDM				HDDM				DDPM			
		Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm	Iter	FVAL	Time	Norm
1000	x1	6	8	0.018569	1.66E−07	4	6	0.00993	9.72E−07	25	27	0.019235	8.09E−07
	x2	6	8	0.013581	2.63E−07	5	7	0.012451	3.58E−08	26	28	0.019958	6.40E−07
	x3	6	8	0.014415	1.67E−07	4	6	0.011838	9.77E−07	25	27	0.020869	8.13E−07
	x4	6	8	0.010919	2.63E−07	5	7	0.012611	3.57E−08	26	28	0.024025	6.38E−07
	x5	6	8	0.015888	6.95E−08	4	6	0.010797	4.06E−07	24	26	0.018638	6.76E−07
	x6	6	8	0.013901	2.87E−07	5	7	0.011791	3.91E−08	26	28	0.023805	6.99E−07
50,000	x1	7	9	0.262561	7.20E−08	5	7	0.225942	1.62E−07	30	32	0.545067	8.07E−07
	x2	7	9	0.267122	1.14E−07	5	7	0.252171	2.57E−07	33	35	0.618142	5.83E−07
	x3	7	9	0.280711	7.21E−08	5	7	0.222714	1.62E−07	30	32	0.545653	8.08E−07
	x4	7	9	0.300448	1.14E−07	5	7	0.238128	2.57E−07	33	35	0.632667	5.83E−07
	x5	6	8	0.24229	4.95E−07	5	7	0.22748	6.79E−08	27	29	0.475108	5.98E−07
	x6	7	9	0.270401	1.24E−07	5	7	0.243771	2.81E−07	33	35	0.619143	9.87E−07
100,000	x1	7	9	0.580075	1.02E−07	5	7	0.412347	2.30E−07	32	34	1.222911	8.23E−07
	x2	7	9	0.53134	1.61E−07	5	7	0.423014	3.63E−07	36	38	1.466238	5.96E−07
	x3	7	9	0.49702	1.02E−07	5	7	0.458689	2.30E−07	32	34	1.252447	8.23E−07
	x4	7	9	0.545991	1.61E−07	5	7	0.41576	3.63E−07	36	38	1.469043	5.96E−07
	x5	6	8	0.489506	7.00E−07	5	7	0.45221	9.60E−08	28	30	1.018146	7.96E−07
	x6	7	9	0.496382	1.76E−07	5	7	0.456999	3.97E−07	37	39	1.537679	6.04E−07

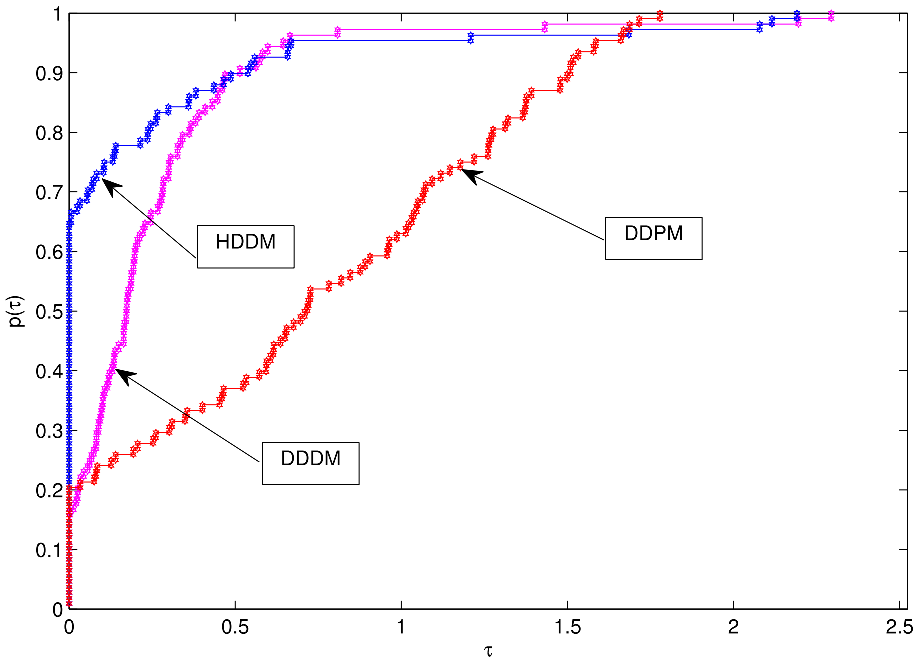

Fig. 3 Performance profile for the CPU time (in second)

evaluations, and "Norm" represents $\|F_k\|$ at the termination point. From Tables 3, 4, 5, 6, 7 and 8, one can easily see that, the three methods are trying to solve the problem in (1.1), but the improved efficacy of the proposed methods are clear from the tables. In fact, the HDDM approach significantly outperforms the DDDM method for nearly all the problems assessed, since it has the least number of iterations, functions evaluations and CPU time, which are below the number of iterations, functions evaluations and the CPU time for the DDDM methods. This is apparently due to the contribution of correction parameter η in the HDDM iteration. Furthermore, the numerical results reported in Tables 3, 4, 5, 6, 7 and 8 were evidently showed that HDDM and DDDM methods, are far better than DDPM method, because they have number of iterations, functions evaluations and CPU time, that far below the ones for DDPM method.

We generate Figs. 1, 2 and 3 using the performance profiles Dolan and Moré (2002) to show the performance of each of the three methods. This is done by plotting the fraction p(τ) of the problems for which each method is within τ of the smallest number of iterations, CPU time and function evaluations respectively. Observe that, from Figs. 1, 2 and 3, the curves corresponding to the HDDM and DDDM methods remain above the other curves representing the DDPM method. This shows that the methods proposed in this paper, outperforms the method compared in terms of fewer iterations, functions evaluation and CPU time (in seconds). However, the curves in the figures indicated that HDDM methods is the most efficient among the three method. Finally, from three figures and the results in Tables 3, 4, 5, 6, 7 and 8 it is obvious that our approaches are very successful in solving large-scale nonlinear problems.

4.2 Applications in image deblurring

This part is dedicated to application of the HDDM algorithm to image deblurring in compressive sensing. This is the process for the efficient acquisition and reconstruction of a signal. It compresses the signal acquired at the time of the sensing. Its application rises in many applications including the statistics and signal processing (Figueiredo et al. 2007; Tibshirani 1996). The most prominent approach requires optimizing problems in sparse recovery is represented by a convex unconstrained optimization problem:

$$\min_x \frac{1}{2}\|w - Qx\|_2^2 + \tau\|x\|_1, \tag{4.1}$$

where $x \in \mathbb{R}^n$, $w \in \mathbb{R}^k$, $Q \in \mathbb{R}^{k\times n}(k << n)$ denotes a linear operator, the parameter $\tau \geq 0$, and $\|\mathrm{x}\|_1 = \sum_{i=1}^n |x_i|$. In the literature of compressive sensing, (4.1) is often called Basis Pursuit Denoising Problem (BPDN) or ℓ_1-regularized least square problem. Some iterative methods for solving (4.1), can be found in Elaine et al. (2007), Mario et al. (2003) and Figueiredo et al. (2007). But, gradient-based methods are the most prominent (Figueiredo et al. 2007). In the scheme, problem (4.1) is expressed as follows. Any vector $x \in \mathbb{R}^n$ is parted into the positive and negative portion as

$$x = p - q, \quad p \geq 0, \quad q \geq 0, \quad p, q \in \mathbb{R}^n. \tag{4.2}$$

Let $p_i = (x_i)_+$, and $q_i = (-x_i)_+$ for $i = 1, 2, \ldots, n$, where $(.)_+$ is the positive operator, which is defined as $(x)_+ = \max\{0, x\}$. Applying the definition of the l_1-norm, we have $\|x\|_1 = e_n^T p + e_n^T q$, with $e_n = (1, 1, 1, \ldots, 1)^T \in \mathbb{R}^n$. So, problem (4.1) can be reformulated as the following

$$\min_{p,q} \frac{1}{2}\|w - Q(p-q)\|_2^2 + \tau e_n^T p + \tau e_n^T q, \quad p, q \geq 0. \tag{4.3}$$

It was shown in Figueiredo et al. (2007) that problem (4.3) can be formulated in more standard bound-constrained quadratic program as follows

$$\min_z \frac{1}{2} z^T H z + c^T z, \quad \text{s.t} \quad z \geq 0, \tag{4.4}$$

where
$z = \begin{pmatrix} p \\ q \end{pmatrix}$, $c = \tau e_{2n} + \begin{pmatrix} -h \\ h \end{pmatrix}$, $h = Q^T w$, $H = \begin{pmatrix} Q^TQ & -Q^TQ \\ -Q^TQ & Q^TQ \end{pmatrix}$. Clearly, the matrix H is positive semi-definite. So, problem in (4.4) is a convex quadratic programming problem, that is translated into the following problem of linear variable inequality (LVI) (Xiao et al. 2011a). Find $z \in \mathbb{R}^n$, such that

$$(z' - z)^T (Hz + c) \geq 0 \quad \forall z' \geq 0. \tag{4.5}$$

Moreover, problem in (4.4) is equivalent to the following linear complementary problem (Xiao et al. 2011a). Find $z \in \mathbb{R}^n$,

$$z \geq 0, \quad Hz + c \geq 0, \quad \text{and} \quad z^T(Hz + c) = 0. \tag{4.6}$$

where $z \in \mathbb{R}^n$ is the solution of problem in (4.6) if and only if it satisfies the nonlinear equations defined by

$$F(z) = \min\{z, Hz + c\} = 0, \tag{4.7}$$

Fig. 4 The original images (first column), the blurred images (second column), the restored images by HDDM (third column) and SGCS (last column)

where F is a vector-valued function that is Lipschitz continuous and monotone as proved in Pang (1986) and Xiao et al. (2011a), and the "min" interpreted as component-wise minimum. Therefore, problem (4.1) can be translated into (1.1). This shows that HDDM algorithm can be applied to solve it.

To further highlight the performance of the HDDM scheme, some numerical experiments was carried out. To highlight the efficiency of the proposed algorithm, its performance to restore some blurred images is compared with a spectral gradient method for ℓ_1-norm problems in compressed sensing (SGCS) (Xiao et al. 2011b). When implementing HDDM algorithm in this experiments, and the following parameters are set $\xi = 1$, $\sigma = 10^{-4}$, $\rho = 0.9$, and $t = 1.2$. The parameters of SGCS algorithm, are taken as in Xiao et al. (2011b). The iteration is set to stop for both methods if the following conditions occur:

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

Table 9 Numerical results for HDDM and SGCS in image restoration

Image	Size	HDDM				SGCS			
		Iter	Time (s)	SNR	SSIM	Iter	Time (s)	SNR	SSIM
Lena	256 × 256	26	1.03	24.41	0.90	270	10.84	24.14	0.90
House	256 × 256	23	0.91	27.66	0.89	228	9.17	27.50	0.89
Barbara	512 × 512	28	6.11	19.64	0.79	257	16.22	19.68	0.79
Pepper	256 × 256	26	0.97	22.84	0.87	296	12.02	22.80	0.87

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

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

where $\hat{x}$ and x are the original image and the restored image, respectively. Furthermore, Structural Similarity (SSIM) index is used in this paper to measure the quality of the restored images (Wang et al. 2004). The MATLAB implementation of the SSIM index can be obtained at http://www.cns.nyu.edu/~lcv/ssim/.

Figure 4 is generated to show the restoration results of different images obtained by HDDM and SGCS methods. From Table 9, both methods are successful in restoration of all the four images, but the improved efficacy of the proposed method is very clear from the table. Although, both method have the same values of SSIM, but SNR values reported in the table indicated that HDDM method is slightly restored the four blurred images with better quality than SGCS method except for Barbara. In fact, the HDDM method remarkably outperforms the SGCS method for restoring all the images assessed, since it has the least number of iterations and CPU time (in second), which are far below the number of iterations and the CPU time for the SGCS method. These results show that the proposed method can restore corrupted images efficiently.

5 Conclusion

In this paper, two double direction methods (DDDM and HDDM) for convex constrained monotone nonlinear equations are presented. The DDDM scheme was achieved by proposing the acceleration parameter that approximated the Jacobian with diagonal matrix in a sufficient manner. We further improved the numerical performances of DDDM by applying Picard–Mann hybrid iterative process (Safeer 2013), and derived HDDM scheme. The proposed methods are completely matrix-free that are globally convergent under certain appropriate conditions. Numerical comparisons have been made using large-scale test problems. Furthermore, Tables 3, 4, 5, 6, 7 and 8 and Figs. 1, 2 and 3 showed that the proposed methods are practically quite welcome, because they have the least number of iteration and CPU time compared to DDPM method (Mohammad 2020). In addition, HDDM method is successfully applied to deal with the experiments on the ℓ_1-norm regularization problem in image restoration and compared its performance with SGCS method (Xiao et al. 2011b). The experiments were carried out with various samples of images, as shown in Table 9, which clearly demonstrated a higher efficiency for the HDDM method. Finally, Fig. 4, indicated that the HDDM method had better restoration of the blurred images because it had the lowest mean iteration number and CPU time than SGCS method. Future work includes, deriving the appropriate value of the parameter $t = t_k \in (1, 2)$ so that t_k can be updated in each iteration to improve the convergence properties and numerical results of HDDM method.

References

Abdullahi H, Halilu AS, Waziri MY (2018) A modified conjugate gradient method via a double direction approach for solving large-scale symmetric nonlinear systems. J Numer Math Stoch 10(1):32–44

Awwal AM, Wang L, Kumam P, Mohammad H (2020) A two-step spectral gradient Projection method for system of nonlinear monotone equations and image deblurring problems. Symmetry 12:874. https://doi.org/10.3390/sym12060874

Dennis JE, More JJ (1974) A characterization of superlinear convergence and its application to quasi-newton methods. Math Comput 28:549–560

Dennis JE, Schnabel RB (1983) Numerical methods for unconstrained optimization and non-linear equations. Prentice Hall, Englewood Cliffs

Dolan E, Moré J (2002) Benchmarking optimization software with performance profiles. J Math Program 91:201–213

Duranovic-Milicic NI (2008) A multi step curve search algorithm in nonlinear optimization. Yugoslav J Oper Res 18:47–52

Duranovic-Milicic NI, Gardasevic-Filipovic M (2010) A multi step curve search algorithm in nonlinear convex case. Facta Univ Ser Math Inform 25:11–24

Elaine T, Wotao Y, Yin Z (2007) A fixed-point continuation method for ell_1-regularized minimization with applications to compressed sensing. CAAM TR07-07, Rice University, pp 43–44

Figueiredo M, Nowak R, Wright SJ (2007) Gradient projection for sparse reconstruction, application to compressed sensing and other inverse problems. IEEE J-STSP IEEE Press, Piscataway, pp 586–597

Halilu AS, Waziri MY, Yusuf I (2020) Efficient matrix-free direction method with line search for solving large-scale system of nonlinear equations. Yugoslav J Oper Res 30(4):399–412

Halilu AS, Waziri MY (2017) Enhanced matrix-free method via double step length approach for solving systems of nonlinear equations. Int J Appl Math Res 6:147–156

Halilu AS, Waziri MY (2017) A transformed double step length method for solving large-scale systems of nonlinear equations. J Numer Math Stoch 9:20–32

Halilu AS, Waziri MY (2018) An improved derivative-free method via double direction approach for solving systems of nonlinear equations. J Ramanujan Math Soc 33:75–89

Halilu AS, Waziri MY (2020) Inexact double step length method for solving systems of nonlinear equations. Stat Optim Inf Comput 8:165–174

Halilu AS, Waziri MY (2020) Solving systems of nonlinear equations using improved double direction method. J Niger Math Soc 32(2):287–301

Halilu AS, Dauda MK, Waziri MY, Mamat M (2019) A derivative-free decent method via acceleration parameter for solving systems of nonlinear equations. Open J Sci Tech 2(3):1–4

Halilu AS, Majumder A, Waziri MY, Abdullahi H (2020) Double direction and step length method for solving system of nonlinear equations. Eur J Mol Clin Med 7(7):3899–3913

Halilu AS, Majumder A, Waziri MY, Ahmed K (2021) Signal recovery with convex constrained nonlinear monotone equations through conjugate gradient hybrid approach. Math Comput Simul 187:520–539

Ishikawa S (1974) Fixed points by a new iteration method. Proc Am Math Soc 44:147–150

Kačurovskii RI (1960) On monotone operators and convex functionals. Usp Math Nauk 15(4):213–215

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

Mann WR (1953) Mean value methods in iterations. Proc Am Math Soc 4:506–510

Mario AT, Figueiredo R, Nowak D (2003) An EM algorithm for wavelet-based image restoration. IEEE Trans Image Process 12(8):906–916

Meintjes K, Morgan AP (1987) A methodology for solving chemical equilibrium systems. Appl Math Comput 22:333–361

Minty GJ (1962) Monotone (nonlinear) operators in Hilbert space. Duke Math J 29(3):341–346

Mohammad H (2020) A descent derivative-free algorithm for nonlinear monotone equations with convex constraints. RAIRO-Oper Res 54:489–505

Mohammad H, Abubakar AB (2017) Barzilai–Borwein-like method for solving large-scale nonlinear systems of equations. J Niger Math Soc 36:71–83

Pang JS (1986) Inexact Newton methods for the nonlinear complementarity problem. Math Program 1:54–71

Petrović MJ, Stanimirović PS, Kontrec N, Mladenović J (2018) Hybrid modification of accelerated double direction method. Math Probl Eng Article ID 1523267, 8. https://doi.org/10.1155/2018/1523267

Petrović MJ, Stanimirovic PS (2014) Accelerated double direction method for solving unconstrained optimization problems. Math Probl Eng 2014:1–8

Picard E (1890) Memoire sur la theorie des equations aux derivees partielles et la meth-ode des approximations successives. J Math Pures Appl 6:145–210

Safeer HK (2013) A Picard–Mann hybrid iterative process. Fixed Point Theory Appl 69:10

Solodov MV, Svaiter BF (1999) A globally convergent inexact Newton method for systems of monotone equations. Reformulation: nonsmooth, piecewise smooth, semismooth and smoothing methods. Springer, Berlin, pp 355–369

Sun M, Tian MY, Wang YJ (2019) Multi-step discrete-time Zhang neural networks with application to time-varying nonlinear optimization. Discret Dyn Nat Soc Article ID 4745759:1–14

Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc 58(1):267–288

Wang CW, Wang YJ (2009) A superlinearly convergent projection method for constrained systems of nonlinear equations. J Glob Optim 44(2):283–296

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:600–612

Wang C, Wang Y, Xu C (2007) A projection method for a system of nonlinear monotone equations with convex constraints. Math Methods Oper Res 66(1):33–46

Waziri MY, Leong WJ, Hassan MA (2011) Jacobian-free diagonal Newton's method for solving nonlinear systems with singular jacobian. Malay J Math Sci 5:241–255

Xiao YH, Zhu H (2013) A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing. J Math Anal Appl 405:310–319

Xiao Y, Wang Q, Hu Q (2011) Non-smooth equations based method for ℓ_1-norm problems with applications to compressed sensing. Nonlinear Anal Theory Methods Appl 74(11):3570–3577

Xiao Y, Wang Q, Hu Q (2011) Non-smooth equations based methods for $l1-$norm problems with applications to compressed sensing. Nonlinear Anal Theory Methods Appl 74:3570–3577

Yuan G, Lu X (2008) A new backtracking inexact BFGS method for symmetric nonlinear equations. Comput Math Appl 55:116–129

Zarantonello EH (1960) Solving functional equations by contractive averaging. Tech. Rep. 160. U. S. Army Math. Research Center, Madison, Wisconsin