Ibrahim et al. *Journal of Inequalities and Applications* (2021) 2021:189
https://doi.org/10.1186/s13660-021-02719-3

Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH

Open Access

A method with inertial extrapolation step for convex constrained monotone equations

Abdulkarim Hassan Ibrahim[1], Poom Kumam[1,2,3*], Auwal Bala Abubakar[4,5] and Jamilu Abubakar[6]

*Correspondence:
poom.kum@kmutt.ac.th
[1]KMUTTFixed Point Research Laboratory, Room SCL 802 Fixed Point Laboratory, Science Laboratory Building, Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khru, Bangkok 10140, Thailand
[2]Center of Excellence in Theoretical and Computational Science (TaCS-CoE), Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha Uthit Rd., Bang Mod, Thung Khru, Bangkok 10140, Thailand
Full list of author information is available at the end of the article

Abstract

In recent times, various algorithms have been incorporated with the inertial extrapolation step to speed up the convergence of the sequence generated by these algorithms. As far as we know, very few results exist regarding algorithms of the inertial derivative-free projection method for solving convex constrained monotone nonlinear equations. In this article, the convergence analysis of a derivative-free iterative algorithm (Liu and Feng in Numer. Algorithms 82(1):245–262, 2019) with an inertial extrapolation step for solving large scale convex constrained monotone nonlinear equations is studied. The proposed method generates a sufficient descent direction at each iteration. Under some mild assumptions, the global convergence of the sequence generated by the proposed method is established. Furthermore, some experimental results are presented to support the theoretical analysis of the proposed method.

Keywords: Iterative method; Inertial algorithm; Nonlinear equations; Derivative-free method; Projection method

1 Introduction

Our main aim in this paper is to find the approximate solutions of the systems of monotone nonlinear equations with convex constraints; precisely, the problem

$$\text{find } x \in \mathcal{C} \text{ s.t. } h(x) = 0, \tag{1}$$

where $h : \mathcal{R}^n \to \mathcal{R}^n$ is assumed to be a monotone and Lipschitz continuous operator, while $\mathcal{C}$ is a nonempty, closed, and convex subset of $\mathcal{R}^n$.

Monotone operator was first introduced by Minty [2]. The concept has aided several studies such as the abstract study of electrical networks [2]. Interest in the study of the systems of monotone nonlinear equations with convex constraint (1) stems mainly from their several applications in various fields. For instance, in power flow equations [3], economic equilibrium problems [4], chemical equilibrium [5], and compressive sensing [6]. These applications have attracted the attention of many researchers. Thus, numerous iterative methods have been proposed by many authors to approximate solutions of (1) (see [7–35] and the references therein).

Among the early methods introduced and studied in the literature are Newton method, quasi-Newton method, Gauss–Newton method, Levenberg–Marquardt method, and their modifications (see, e.g., [36–39] and the references therein). These methods have fast local convergence but are not efficient for solving large scale nonlinear monotone equations, because they involve the computation of the Jacobian matrix or its approximation per iteration, which is well known to require a large amount of storage. To overcome this problem, various alternatives and modifications of the early methods have been proposed by several authors. Amongst these methods are conjugate gradient methods, spectral conjugate gradient methods, and spectral gradient methods. Extensions of the conjugate gradient method and its variant to solve large scale nonlinear equations have been obtained by several authors. For instance, motivated by the stability and efficiency of the Dai–Yuan (DY) conjugate gradient method [40] for solving unconstrained optimization problems, Liu and Feng [1] proposed a derivative-free projection method based on the structures of the DY conjugate gradient method [40]. This method inherits the stability of the DY method and greatly improves its computing performance.

In practical applications, it is always desirable to have iterative algorithms that have a high rate of convergence [41–46]. An increasingly important acceleration method is the inertial extrapolation type algorithms [47, 48]. They use an iterative procedure in which subsequent terms are obtained using the preceding two terms. This idea was first introduced by Polyak [49] and was inspired by an implicit discretization of a second-order-in-time dissipative dynamical system, so-called 'Heavy Ball with Friction':

$$\nu''(t) + \gamma \nu'(t) + \nabla f\big(\nu(t)\big) = 0, \tag{2}$$

where $\gamma > 0$ and $f : \mathcal{R}^n \to \mathcal{R}$ is differentiable. System (2) is discretized so that, having the terms x_{k-1} and x_k, the next term x_{k+1} can be determined using

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{j^2} + \gamma \frac{x_k - x_{k-1}}{j} + \nabla f(x_k) = 0, \quad k \geq 1, \tag{3}$$

where j is the step size. Equation (3) yields the following iterative algorithm:

$$x_{k+1} = x_k + \beta(x_k - x_{k-1}) - \alpha \nabla f(x_k), \quad k \geq 1, \tag{4}$$

where $\beta = 1 - \gamma j$, $\alpha = j^2$ and $\beta(x_k - x_{k-1})$ is called the inertial extrapolation term which is intended to speed up the convergence of the sequence generated by equation (4).

Several algorithms with inertial extrapolation term have been tested in the solution of several problems (for example, imaging/data analysis problems and motion of a body in a potential field), and the test showed that the inertial steps remarkably increase the convergence speed of these algorithms (see [47, 48, 50] and other references therein). Therefore, this property is very important. As far as we know, there are not many results regarding algorithms of inertial derivative-free projection for solving (1).

Our concern now is the following: *Based on the derivative-free iterative algorithm of Liu and Feng* [1], *can we construct an inertial derivative-free method for solving the system of monotone nonlinear equations with convex constraints?*

In this paper, we give a positive answer to the aforementioned question. Motivated and inspired by the algorithm in [1], we introduce an inertial derivative-free algorithm for solving (1). Our proposed method is a combination of inertial extrapolation step and the derivative-free iterative method for nonlinear monotone equations with convex constraints [1]. We obtain the global convergence result under mild assumptions. Using a set of test problems, we illustrate the numerical behaviors of the algorithm in [1] and compare it with the algorithm presented in this paper. The results indicate that the proposed algorithm with the inertial step is superior in terms of the number of iterations and function evaluations.

The rest of paper is organized as follows. The next section contains some preliminaries. The proposed inertial algorithm is presented in Sect. 3, and its convergence is presented in the fourth section. The last section is devoted to presentation of examples and numerical results.

2 Preliminaries

We recall some known definitions and results which will be used in the sequel. First, let us denote by $\mathbf{SOL}(\mathbf{h},\mathcal{C})$ the solution set of (1).

Definition 2.1 Let $\mathcal{C}$ be a nonempty closed convex subset of $\mathcal{R}^n$. A mapping $h:\mathcal{R}^n\to\mathcal{R}^n$ is said to be:

(i) monotone on $\mathcal{C}$ if

$$\big(h(x)-h(z)\big)^T(x-z)\geq 0,\quad \forall x,z\in\mathcal{C}.$$

(ii) L-Lipschitz continuous on $\mathcal{C}$, if there exists $L>0$ such that

$$\big\|h(x)-h(z)\big\|\leq L\|x-z\|,\quad \forall x,z\in\mathcal{C}.$$

Definition 2.2 Let $\mathcal{C}\subset\mathcal{R}^n$ be a closed and convex set, some vector $x\in\mathcal{R}^n$, the orthogonal projection of x onto $\mathcal{C}$ denoted by $P_{\mathcal{C}}(x)$, is defined by

$$P_{\mathcal{C}}(x)=\arg\min\big\{\|z-x\|\mid z\in\mathcal{C}\big\},$$

where $\|x\|=\sqrt{x^Tx}$.

The following lemma gives some well-known characteristics of the projection operator.

Lemma 2.3 *Let $\mathcal{C}\subset\mathcal{R}^n$ be a nonempty closed and convex set. Then the following statements hold*:

(i) $(x-P_{\mathcal{C}}(x))^T(P_{\mathcal{C}}(x)-z)\geq 0$, $\forall x\in\mathcal{R}^n$, $\forall z\in\mathcal{C}$.

(ii) $\|P_{\mathcal{C}}(x)-P_{\mathcal{C}}(z)\|\leq\|x-z\|$, $\forall x,z\in\mathcal{R}^n$.

(iii) $\|P_{\mathcal{C}}(x)-z\|^2\leq\|x-z\|^2-\|x-P_{\mathcal{C}}(x)\|^2$, $\forall x\in\mathcal{R}^n$, $\forall z\in\mathcal{C}$.

Lemma 2.4 ([51]) *Let $\mathcal{R}^n$ be an Euclidean space. Then the following inequality holds*:

$$\|x+z\|^2\leq\|x\|^2+2z^T(x+z),\quad \forall x,z\in\mathcal{R}^n.$$

Lemma 2.5 ([52]) *Let* $\{x_k\}$ *and* $\{z_k\}$ *be sequences of nonnegative real numbers satisfying the following relation*:

$$x_{k+1} \leq x_k + z_k,$$

where $\sum_{k=1}^{\infty} z_k < \infty$, *then* $\lim_{k\to\infty} x_k$ *exists*.

3 Proposed method

Based on the Liu and Feng [1] derivative-free iterative method for monotone nonlinear equation with convex constraint, in the sequel, we present an inertial extrapolation algorithm for solving the system of nonlinear monotone equations (1). The corresponding algorithm, which we refer to as the inertial projected Dai–Yuan (IPDY) algorithm, uses a strategy which tracks the optimal x-value by starting with an initial x-value x_0 and thereafter updating the x by performing iterations of the form

$$x_{k+1} = x_k + \alpha_k d_k, \quad k \geq 0, \tag{5}$$

where α_k is a positive step size obtained by a line search procedure, and d_k is the search direction implemented so that

$$h(x_k)^T d_k = -c\big\|h(x_k)\big\|^2, \quad c > 0, \tag{6}$$

is fulfilled. Next, we give a precise statement for our method as follows.

Algorithm 1 (Inertial projected Dai–Yuan algorithm (IPDY))

1 **(S.0)** Choose $x_0, x_1 \in \mathcal{C}$, $Tol \in (0,1), a \in (0,1], \sigma > 0, \theta \in [0,1) r \in (0,1)$. Set $k := 1$.

2 **(S.1)** Compute

$$w_k = x_k + \theta_k(x_k - x_{k-1}),$$

where $0 \leq \theta_k \leq \tilde{\theta}_k$ with

$$\tilde{\theta}_k := \begin{cases} \min\{\theta, \frac{1}{k^2\|x_k - x_{k-1}\|^2}\} & \text{if } x_k \neq x_{k-1}, \\ \theta, & \text{otherwise.} \end{cases} \tag{7}$$

3 **(S.2)** Compute $h(w_k)$. If $\|h(w_k)\| \leq Tol$, stop. Otherwise, generate the search direction d_k by

4

$$d_k := \begin{cases} -h(w_k) & \text{if } k = 1, \\ -\zeta_k h(w_k) + \beta_k^{\mathrm{IPDY}} d_{k-1} & \text{if } k > 1, \end{cases} \tag{8}$$

where

$$\begin{aligned}
&\beta_k^{\mathrm{IPDY}} := \frac{\|h(w_k)\|^2}{d_{k-1}^T y_{k-1}}, \qquad \zeta_k := c_0 + \frac{h(w_k)^T d_{k-1}}{d_{k-1}^T y_{k-1}}, \quad c_0 > 0, \\
&v_{k-1} := h(w_k) - h(w_{k-1}), \qquad\qquad (9)\\
&y_{k-1} := v_{k-1} + t_{k-1} d_{k-1}, \qquad t_{k-1} := 1 + \max\left\{0, -\frac{d_{k-1}^T v_{k-1}}{d_{k-1}^T d_{k-1}}\right\}.
\end{aligned}$$

(S.3) Find $z_k = w_k + \alpha_k d_k$, where $\alpha_k = ar^i$ with i being the smallest nonnegative integer such that

$$-h(w_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \left\| h(w_k + \alpha_k d_k) \right\| \|d_k\|^2. \tag{10}$$

(S.4) If $z_k \in \mathcal{C}$ and $\|h(z_k)\| \leq Tol$, stop. Otherwise, compute the next iterate by

$$x_{k+1} = P_{\mathcal{C}}\left[w_k - \lambda_k h(z_k)\right], \tag{11}$$

where

$$\lambda_k := \frac{h(z_k)^T (w_k - z_k)}{\|h(z_k)\|^2}.$$

(S.5) Set $k \leftarrow k + 1$, and return to (S.1).

Remark 3.1 For all $k \geq 0$, it can be observed from equation (7) that $\theta_k \|x_k - x_{k-1}\|^2 \leq \frac{1}{k^2}$. This implies that

$$\sum_{k=1}^{\infty} \theta_k \|x_k - x_{k-1}\|^2 < \infty.$$

Throughout this paper, we make use of the following assumptions.

Assumption 1

(A1) The solution set $\mathcal{C}^*$ of (1) is nonempty.

(A2) h is monotone on $\mathcal{C}$.

(A3) h is Lipschitz continuous on $\mathcal{C}$.

4 Convergence result

In this section, convergence analysis of our algorithm is presented. We start by proving some lemmas followed by the proof of the main theorem.

Lemma 4.1 *Let d_k be generated by Algorithm 1. Then d_k always satisfies the sufficient descent condition, that is,*

$$h(w_k)^T d_k = -c_0 \|h(w_k)\|^2, \quad c_0 > 0. \tag{12}$$

Proof For $k = 1$, multiplying both sides of (8) by $h(w_0)^T$, we have

$$h(w_0)^T d_0 = -\|h(w_0)\|^2.$$

Also for $k > 1$, multiplying both sides of (8) by $h(w_k)^T$, we get

$$\begin{aligned} h(w_k)^T d_k &= -\zeta_k \|h(w_k)\|^2 + \beta_k h(w_k)^T d_{k-1} \\ &= -\left(c_0 + \frac{h(w_k)^T d_{k-1}}{d_{k-1}^T y_{k-1}}\right)\|h(w_k)\|^2 + \frac{\|h(w_k)\|^2}{d_{k-1}^T y_{k-1}} h(w_k)^T d_{k-1} \\ &= -c_0 \|h(w_k)\|^2. \end{aligned}$$

□

Remark 4.2 From the definition of y_{k-1} and t_{k-1}, it holds that

$$d_{k-1}^T y_{k-1} \geq d_{k-1}^T v_{k-1} + \|d_{k-1}\|^2 - d_{k-1}^T v_{k-1} = \|d_{k-1}\|^2,$$

then from (12) we have

$$d_{k-1}^T y_{k-1} \geq c_0^2 \|h(w_{k-1})\|^2.$$

This indicates that $d_{k-1}^T y_{k-1}$ is always positive when the solution of (1) is not achieved, which means that the parameters ζ_k and β_k are well defined.

Lemma 4.3 *The line search condition (10) is well defined. That is, for all $k \geq 1$, there exists a nonnegative integer i satisfying (10).*

Proof The proof of Lemma 4.3 can be obtained in the same way as [1] with the difference that the sequence $\{x_k\}$ is replaced with the inertial extrapolation term w_k. □

Lemma 4.4 *Suppose that h is a monotone and Lipschitz continuous mapping, and $\{w_k\}$ and $\{z_k\}$ are sequences generated by Algorithm 1, then*

$$\alpha_k > \max\left\{a, \frac{rc_0\|h(w_k)\|^2}{(L + \sigma\|h(w_k + \tilde{\alpha}_k d_k)\|)\|d_k\|^2}\right\}. \tag{13}$$

Proof From line search (10), if $\alpha_k \neq a$, then $\tilde{\alpha}_k r^{-1}$ does not satisfy the line search. That is,

$$-h(w_k + \tilde{\alpha}_k d_k)^T d_k < \sigma \tilde{\alpha}_k \|h(w_k + \tilde{\alpha}_k d_k)\| \|d_k\|^2.$$

This fact, in combination with the Lipschitz continuity assumption (A3) and the sufficient descent condition (12), expresses

$$\begin{aligned} c_0\|h(w_k)\|^2 &= -h(w_k)^T d_k \\ &= \big(h(w_k+\tilde{\alpha_k}d_k)-h(w_k)\big)^T d_k - h(w_k+\tilde{\alpha_k}d_k)^T d_k \\ &< \tilde{\alpha_k}\big(L+\sigma\|h(w_k+\tilde{\alpha_k}d_k)\|\big)\|d_k\|^2. \end{aligned}$$

This yields the desired inequality (13). □

Lemma 4.5 *Let* $\{x_k\}$ *and* $\{z_k\}$ *be generated by Algorithm 1. If* $x^* \in \mathbf{SOL}(\mathbf{h},\mathcal{C})$, *then under Assumption 1, it holds that*

$$\|x_{k+1}-x^*\|^2 \le \|w_k-x^*\|^2 - \sigma^2\|w_k-z_k\|^4. \tag{14}$$

Moreover, the sequence $\{x_k\}$ *is bounded and*

$$\sum_{k=1}^{\infty}\|w_k-z_k\|^4 < \infty. \tag{15}$$

Proof By the monotonicity of the mapping h, we have

$$\begin{aligned} h(z_k)^T(w_k-x^*) &= h(z_k)^T(w_k-z_k)+h(z_k)^T(z_k-x^*) \\ &\ge h(z_k)^T(w_k-z_k)+h(x^*)^T(z_k-x^*) \\ &= h(z_k)^T(w_k-z_k) \qquad (16) \\ &\ge \sigma\|h(z_k)\|\|w_k-z_k\|^2. \qquad (17) \end{aligned}$$

By Lemma 2.3(iii), (16), and (17), it holds that, for any $x^* \in \mathbf{SOL}(\mathbf{h},\mathcal{C})$,

$$\begin{aligned} \|x_{k+1}-x^*\|^2 &= \|P_{\mathcal{C}}(w_k-\lambda_k h(z_k))-x^*\|^2 \\ &\le \|(w_k-\lambda_k h(z_k))-x^*\|^2 - \|(w_k-\lambda_k h(z_k))-P_{\mathcal{C}}(w_k-\lambda_k h(z_k))\|^2 \\ &\le \|w_k-\lambda_k h(z_k)-x^*\|^2 \\ &\le \|w_k-x^*\|^2 - 2\lambda_k h(z_k)^T(w_k-x^*)+\lambda_k^2\|h(z_k)\|^2 \\ &\le \|w_k-x^*\|^2 - 2\lambda_k h(z_k)^T(w_k-z_k)+\lambda_k^2\|h(z_k)\|^2 \\ &\le \|w_k-x^*\|^2 - \frac{[h(z_k)^T(w_k-z_k)]^2}{\|h(z_k)\|^2} \\ &\le \|w_k-x^*\|^2 - \sigma^2\|w_k-z_k\|^4. \qquad (18) \end{aligned}$$

From inequality (18), we can deduce that

$$\begin{aligned} \|x_{k+1}-x^*\| &\le \|w_k-x^*\| \\ &= \|x_k+\theta_k(x_k-x_{k-1})-x^*\| \\ &\le \|x_k-x^*\| + \theta_k\|x_k-x_{k-1}\|. \qquad (19) \end{aligned}$$

From Remark 3.1, noting that $\sum_{k=1}^{\infty}\theta_k\|x_k - x_{k-1}\| < \infty$, by Lemma 2.5, we deduce that the sequence $\{\|x_k - x^*\|\}$ is bounded by a positive number, say M_0. Therefore, for all k, we have that

$$\|x_k - x^*\| \leq M_0. \tag{20}$$

Thus, we can infer that $\|x_k - x_{k-1}\| \leq 2M_0$. Using the aforementioned facts, we have

$$\begin{aligned}\|w_k - x^*\|^2 &= \|x_k + \theta_k(x_k - x_{k-1}) - x^*\|^2 \\ &\leq \|x_k - x^*\|^2 + 2\theta_k(x_k - x_{k-1})^T\bigl(x_k + \theta_k(x_k - x_{k-1}) - x^*\bigr) \\ &\leq \|x_k - x^*\|^2 + 2\theta_k\|x_k - x_{k-1}\|\bigl(\|x_k - x^*\| + \theta_k\|x_k - x_{k-1}\|\bigr) \\ &\leq \|x_k - x^*\|^2 + 2M_0\theta_k\|x_k - x_{k-1}\| + 4M_0\theta_k\|x_k - x_{k-1}\| \\ &= \|x_k - x^*\|^2 + 6M_0\theta_k\|x_k - x_{k-1}\|. \end{aligned} \tag{21}$$

Combining (21) with (18), we have

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 + 6M_0\theta_k\|x_k - x_{k-1}\| - \sigma^2\|w_k - z_k\|^4. \tag{22}$$

Thus, we have

$$\sigma^2\|w_k - z_k\|^4 \leq \|x_k - x^*\|^2 + 6M_0\theta_k\|x_k - x_{k-1}\| - \|x_{k+1} - x^*\|^2. \tag{23}$$

Adding (23) for $k = 1, 2, 3, \ldots$, we have

$$\sigma^2\sum_{k=1}^{\infty}\|w_k - z_k\|^4 \leq \sum_{k=1}^{\infty}\bigl(\|x_k - x^*\|^2 + 6M_0\theta_k\|x_k - x_{k-1}\| - \|x_{k+1} - x^*\|^2\bigr).$$

But $\sum_{k=1}^{\infty}(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2)$ is finite since the sequence $\{\|x_{k+1} - x^*\|\}$ is convergent and $\sum_{k=1}^{\infty}\theta_k\|x_k - x_{k-1}\| < \infty$. It implies that

$$\sigma^2\sum_{k=1}^{\infty}\|w_k - z_k\|^4 \leq \sum_{k=1}^{\infty}\bigl(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + 6M_0\theta_k\|x_k - x_{k-1}\|\bigr) < \infty.$$

Therefore,

$$\lim_{k\to\infty}\|w_k - z_k\| = 0. \tag{24}$$

□

Remark 4.6 By the definition of $\{z_k\}$ and (24), we have

$$\lim_{k\to\infty}\alpha_k\|d_k\| = 0. \tag{25}$$

Theorem 4.7 *Suppose that the conditions of Assumption 1 hold. If $\{x_k\}$ is the sequence generated by (11) in Algorithm 1, then*

$$\liminf_{k\to\infty} \|h(x_k)\| = 0. \tag{26}$$

Furthermore, $\{x_k\}$ converges to a solution of (1).

Proof We first prove that

$$\liminf_{k\to\infty} \|h(w_k)\| = 0. \tag{27}$$

Suppose that equality (27) does not hold. Then there exists a constant $\varepsilon > 0$ such that

$$\|h(w_k)\| \geq \varepsilon, \quad \forall k \geq 1.$$

This fact, in combination with the sufficient descent condition (12), implies that

$$\|d_k\| \geq c_0\varepsilon, \quad \forall k \geq 1. \tag{28}$$

This shows that

$$\lim_{k\to\infty} \alpha_k = 0. \tag{29}$$

On the other hand, by the Lipschitz continuity assumption (A3) and (20), we have

$$\begin{aligned} \|h(w_k)\| &= \|h(w_k) - h(x^*)\| \leq L\|w_k - x^*\| \\ &\leq L(\|x_k - x^*\| + \|x_k - x_{k-1}\|) \leq 3LM_0 = M_h. \end{aligned} \tag{30}$$

By using the Cauchy–Schwarz inequality, Remark 4.2, and (28), it follows from (8)–(9) that, for all $k > 1$,

$$\|d_k\| \leq \gamma_d.$$

Then we get from (13) that

$$\begin{aligned} \alpha_k\|d_k\| &> \max\left\{a, \frac{rc_0\|h(w_k)\|^2}{(L + \sigma\|h(w_k + \tilde{\alpha}_k d_k)\|)\|d_k\|^2}\right\}\|d_k\| \\ &> \max\left\{ac_0\varepsilon, \frac{rc_0\varepsilon^2}{(L + \sigma M_h^*)\gamma_d}\right\} > 0, \end{aligned}$$

which contradicts (29). Thus, (27) holds. Now, since we know that

$$\|x_k - w_k\| = \|x_k - (x_k + \theta_k(x_k - x_{k-1}))\| = \theta_k\|x_k - x_{k-1}\| \to 0, \tag{31}$$

by the continuity of h, we have that

$$\liminf_{k\to\infty}\|h(x_k)\| = 0. \tag{32}$$

From the continuity of h, the boundedness of $\{x_k\}$, and (32), it implies that the sequence $\{x_k\}$ generated by Algorithm 1 has an accumulation point x^* such that $h(x^*) = 0$. On the other hand, the sequence $\{x_k - x^*\}$ is convergent by Lemma 2.5, which means that the whole sequence $\{x_k\}$ globally converges to the solution x^* of system (1). □

5 Numerical experiments

In this section, an efficiency comparison between the proposed method called IPDY and the method proposed Liu and Feng in [1] called PDY is presented. Recall that the IPDY is a modification of the method in PDY by introducing the inertial term. The metrics considered for the comparison are the number of iterations (NI) and function evaluations (NF). This means that the method with the least NI and NF is the best method. The following were considered for the experimental comparison:

- Dimensions: 1000, 5000, 10,000, 50,000, 100,000.
- Parameters: For IPDY, we select $\theta = 0.8$, $a = 1$, $r = 0.7$, $\sigma = 0.01$, $c_0 = 1$. As for PDY, all parameters are selected as in [1].
- Terminating criterion: When $\|h(w_k)\| \leq 10^{-6}$.
- Implementation software: All methods are coded in MATLAB R2019ba and run on a PC with an intel COREi3 processor, 8 GB of RAM and CPU 2.30 GHz.

The two methods were compared based on the following test problems, where $h = (h_1, h_2, \ldots, h_n)^T$.

Problem 1 (Modified exponential function [53])

$$h_1(x) = e^{x_1} - 1$$
$$h_i(x) = e^{x_i} + x_i - 1, \quad i = 1, 2, \ldots, n-1,$$
$$\mathcal{C} = \mathcal{R}^n_+.$$

Problem 2 (Logarithmic function [53])

$$h_i(x_i) = \log(x_i + 1) - \frac{x_i}{n}, \quad i = 1, 2, \ldots, n,$$
$$\mathcal{C} = \mathcal{R}^n_+.$$

Problem 3 (Nonsmooth function [54])

$$h_i(x) = 2x_i - \sin(|x_i|), \quad \text{for } i = 1, 2, \ldots, n,$$
$$\mathcal{C} = \left\{x \in \mathcal{R}^n_+ : x \geq 0, \sum_{i=1}^{n} x_i \leq n\right\}.$$

Problem 4 ([55])

$$h_i(x) = \min\big(\min\big(|x_i|, x_i^2\big), \max\big(|x_i|, x_i^3\big)\big), \quad i = 1, 2, \ldots, n,$$
$$\mathcal{C} = \mathcal{R}_+^n.$$

Problem 5 (Strictly convex function I [53])

$$h_i(x) = e^{x_i} - 1, \quad i = 1, 2, \ldots, n,$$
$$\mathcal{C} = \mathcal{R}_+^n.$$

Problem 6 (Strictly convex function II [53])

$$h_i(x) = \left(\frac{i}{n}\right)e^{x_i} - 1, \quad i = 1, 2, \ldots, n,$$
$$\mathcal{C} = \mathcal{R}_+^n.$$

Problem 7 (Tridiagonal exponential function [53])

$$h_1(x) = x_1 - e^{\cos(l(x_1+x_2))}$$
$$h_i(x) = x_i - e^{\cos(l(x_{i-1}+x_i+x_{i+1}))}, \quad i = 2, \ldots, n-1,$$
$$h_n(x) = x_n - e^{\cos(l(x_{n-1}+x_n))},$$
$$l = \frac{1}{n+1} \quad \text{and} \quad \mathcal{C} = \mathcal{R}_+^n.$$

Problem 8 (Nonsmooth function II [56])

$$h_i(x) = x_i - \sin\big(|x_i - 1|\big), \quad \text{for } i = 1, 2, \ldots, n,$$
$$\mathcal{C} = \left\{x \in \mathcal{R}_+^n : x \geq -1, \sum_{i=1}^{n} x_i \leq n\right\}.$$

Problem 9 (Trig-Exp function [57])

$$h_1(x) = 3x_1^3 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2),$$
$$h_i(x) = 3x_i^3 + 2x_{i+1} - 5 + \sin(x_i - x_{i+1})\sin(x_i + x_{i+1}) + 4x_i - x_{i-1}e^{(x_{i-1}-x_i)} - 3, \quad \text{for } 1 < i < n,$$
$$h_n(x) = 4x_n - x_{n-1}e^{(x_{n-1}-x_n)} - 3,$$
$$\mathcal{C} = \mathcal{R}_+^n.$$

Table 1 List of starting points

S/N	Starting points
1	$x_0 = (0.2, 0.2, \dots, 0.2)^T, x_1 = (0.1, 0.1, \dots, 0.1)^T$
2	$x_0 = (0.2, 0.2, \dots, 0.2)^T, x_1 = (0.2, 0.2, \dots, 0.2)^T$
3	$x_0 = (0.5, 0.5, \dots, 0.5)^T, x_1 = (0.5, 0.5, \dots, 0.5)^T$
4	$x_0 = (1.2, 1.2, \dots, 1.2)^T, x_1 = (1.2, 1.2, \dots, 1.2)^T$
5	$x_0 = (1.5, 1.5, \dots, 1.5)^T, x_1 = (1.5, 1.5, \dots, 1.5)^T$
6	$x_0 = (2, 2, \dots, 2)^T, x_1 = (2, 2, \dots, 2)^T$
7	$x_0 = \text{rand}(n, 1), x_1 = \text{rand}(n, 1)$

Problem 10 (Penalty function I [58])

$$\xi_i = \sum_{i=1}^{n} x_i^2, \qquad c = 10^{-5},$$

$$h_i(x) = 2c(x_i - 1) + 4(\xi_i - 0.25)x_i, \quad i = 1, 2, \dots, n,$$

$$\mathcal{C} = \mathcal{R}_+^n.$$

Above is a list of the seven starting points in Table 1.

The numerical results are given in Tables 2–11 in the Appendix section for the sake of comparison. From the table, it can be observed that the IPDY method has lower NI and NF than the PDY in most of the problems. This is the result of the inertial effect possessed by the IPDY method. For all initial points used, it can be observed that the IPDY method was able to solve the test problems. However, it can be seen that for Problem 3, using the randomly selected initial points, the IPDY method failed for dimension 5000 and 10,000. On the overall, to visualize the performance of IPDY verses the PDY method, we employ the well-known performance profiles of Dolan and Moré [59] defined as:

$$\rho(\tau) := \frac{1}{|T_P|} \left| \left\{ t_p \in T_P : \log_2 \left(\frac{t_{p,q}}{\min\{t_{p,q} : q \in Q\}} \right) \leq \tau \right\} \right|,$$

where T_P is the test set, $|T_P|$ is the number of problems in the test set T_P, Q is the set of optimization solvers, and $t_{p,q}$ is the NI (or the NF) for $t_p \in T_P$ and $q \in Q$. Figures 1 and 2 were obtained using the above performance profiles.

From Figs. 1 and 2, the IPDY method has the least NI and NF in over 80% of the problem, respectively. This can be seen on the *y-axis* of the plots. As a conclusion, it can be said that the purpose of introducing the inertial effect was achieved as the IPDY method recorded the lowest number of iterations and function evaluations.

6 Conclusion

The paper has proposed an inertial derivative-free algorithm, called IPDY, for solving systems of monotone nonlinear equations with convex constraints in the Euclidean space. Under some suitable conditions imposed on parameters, we established the global convergence of the algorithm. In all our comparisons, the numerical results as shown in Tables 2–11 and Figs. 1, 2 demonstrate that our method converges faster and is more efficient than the PDY algorithm. In the future, we plan to study different variants of derivative-free methods with the inertial extrapolation step and apply them in various directions like image deblurring and signal processing problems.

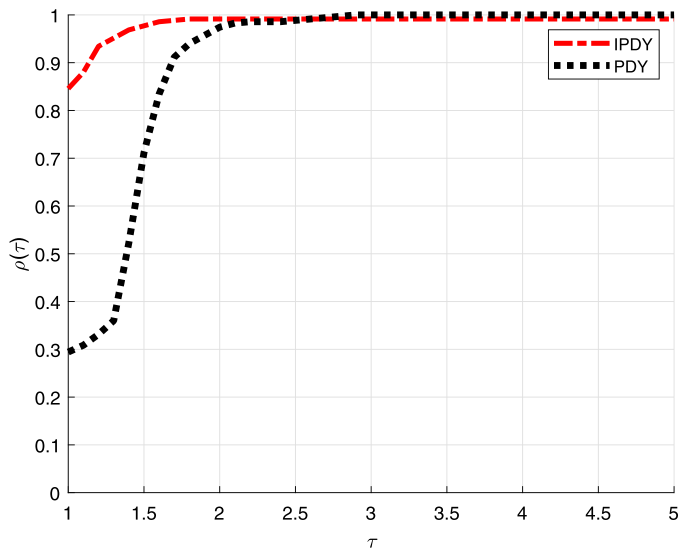

Figure 1 Performance profiles based on the number of iterations

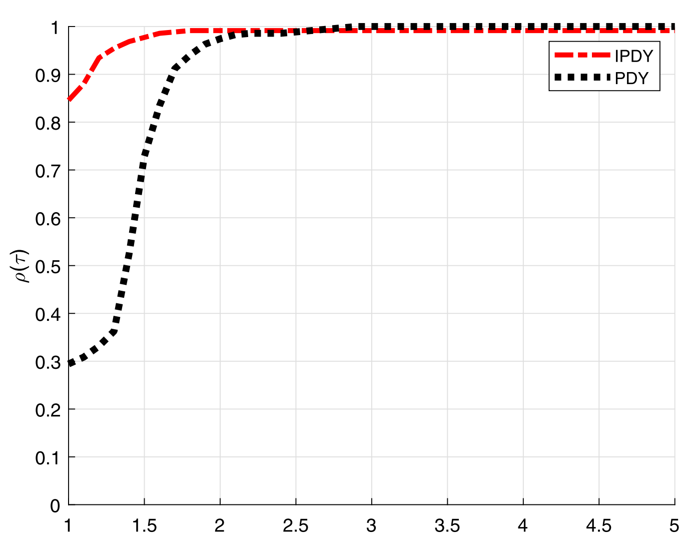

Figure 2 Performance profiles based on the number of function evaluations

Appendix

Table 2 Test result for Problem 1

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	10	40	0.029733	16	64	0.03464
	(x_2,x_2)	11	44	0.030847	16	64	0.023968
	(x_3,x_3)	12	48	0.010731	17	68	0.014996
	(x_4,x_4)	12	48	0.024497	18	72	0.015807
	(x_5,x_5)	12	48	0.021592	18	72	0.032048
	(x_6,x_6)	12	48	0.022392	18	72	0.015106
	(x_7,x_7)	18	72	0.031427	17	68	0.011476
5000	(x_1,x_2)	10	40	0.042238	16	64	0.04849
	(x_2,x_2)	11	44	0.067285	17	68	0.061148
	(x_3,x_3)	12	48	0.044846	18	72	0.072658
	(x_4,x_4)	12	48	0.054505	19	76	0.052417
	(x_5,x_5)	13	52	0.057993	18	72	0.055616
	(x_6,x_6)	12	48	0.05459	18	72	0.058231
	(x_7,x_7)	20	80	0.083833	18	72	0.062164
10,000	(x_1,x_2)	9	36	0.0635	17	68	0.10741
	(x_2,x_2)	12	48	0.10059	17	68	0.094942
	(x_3,x_3)	12	48	0.080824	18	72	0.12731
	(x_4,x_4)	12	48	0.13745	19	76	0.10435
	(x_5,x_5)	14	56	0.10695	20	80	0.13305
	(x_6,x_6)	13	52	0.12456	19	76	0.13785
	(x_7,x_7)	19	76	0.14196	18	72	0.13154
50,000	(x_1,x_2)	9	36	0.31707	17	68	0.4311
	(x_2,x_2)	12	48	0.29351	18	72	0.37639
	(x_3,x_3)	13	52	0.35328	19	76	0.42555
	(x_4,x_4)	15	60	0.48101	20	80	0.79858
	(x_5,x_5)	16	64	0.64442	22	88	0.74776
	(x_6,x_6)	16	64	0.59247	23	92	1.3893
	(x_7,x_7)	14	56	0.47954	19	76	0.54864
100,000	(x_1,x_2)	9	36	0.54463	18	72	1.3933
	(x_2,x_2)	12	48	0.68154	18	72	1.0358
	(x_3,x_3)	14	56	0.76761	19	76	1.1113
	(x_4,x_4)	16	64	0.99936	23	92	1.5626
	(x_5,x_5)	17	68	1.2856	23	92	1.4079
	(x_6,x_6)	19	76	1.672	26	104	1.6467
	(x_7,x_7)	15	60	0.87727	20	80	1.0802

Table 3 Test result for Problem 2

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	7	27	0.096872	13	51	0.019005
	(x_2,x_2)	10	39	0.020041	15	59	0.013681
	(x_3,x_3)	11	43	0.010628	16	63	0.019522
	(x_4,x_4)	11	43	0.014032	18	71	0.015199
	(x_5,x_5)	12	47	0.032128	18	71	0.020678
	(x_6,x_6)	12	47	0.031786	18	71	0.018411
	(x_7,x_7)	17	67	0.05388	17	67	0.022057
5000	(x_1,x_2)	8	31	0.057562	14	55	0.49934
	(x_2,x_2)	11	43	0.14625	15	59	0.069077
	(x_3,x_3)	12	47	0.103	17	67	0.073856
	(x_4,x_4)	12	47	0.24808	18	71	0.095211
	(x_5,x_5)	12	47	0.039387	19	75	0.069529
	(x_6,x_6)	12	47	0.078057	19	75	0.09134
	(x_7,x_7)	23	91	0.17661	17	67	0.052495
10,000	(x_1,x_2)	8	31	0.043896	14	55	0.1585
	(x_2,x_2)	11	43	0.12549	16	63	0.089382
	(x_3,x_3)	12	47	0.12935	17	67	0.11205
	(x_4,x_4)	12	47	0.12544	19	75	0.11729
	(x_5,x_5)	12	47	0.11877	19	75	0.14566
	(x_6,x_6)	12	48	0.12846	19	76	0.13503
	(x_7,x_7)	21	83	0.20705	18	71	0.31225
50,000	(x_1,x_2)	8	31	0.24247	15	59	0.48899
	(x_2,x_2)	11	43	0.55387	16	63	0.35285
	(x_3,x_3)	12	47	0.77037	18	71	0.44647
	(x_4,x_4)	14	56	0.73475	21	84	0.5386
	(x_5,x_5)	14	56	0.65943	21	84	1.1396
	(x_6,x_6)	15	60	0.46108	21	84	0.54725
	(x_7,x_7)	22	87	0.87983	19	75	0.46819
100,000	(x_1,x_2)	8	31	0.36339	15	59	0.71881
	(x_2,x_2)	11	43	0.48992	17	67	0.76688
	(x_3,x_3)	13	52	0.61507	18	72	0.88845
	(x_4,x_4)	15	60	0.85806	22	88	1.1812
	(x_5,x_5)	15	60	1.0527	22	88	1.1886
	(x_6,x_6)	15	60	1.5379	22	88	1.1909
	(x_7,x_7)	19	76	1.1514	20	80	1.0089

Table 4 Test result for Problem 3

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	9	36	0.13993	15	60	0.009012
	(x_2,x_2)	10	40	0.016829	16	64	0.012426
	(x_3,x_3)	11	44	0.01805	16	64	0.013702
	(x_4,x_4)	11	44	0.013523	17	68	0.011571
	(x_5,x_5)	11	44	0.011749	18	72	0.016509
	(x_6,x_6)	12	48	0.031454	18	72	0.012098
	(x_7,x_7)	259	1036	0.62134	17	68	0.010712
5000	(x_1,x_2)	9	36	0.60992	16	64	0.13774
	(x_2,x_2)	11	44	0.050266	16	64	0.12547
	(x_3,x_3)	11	44	0.057844	17	68	0.037555
	(x_4,x_4)	12	48	0.052319	18	72	0.1555
	(x_5,x_5)	11	44	0.12611	18	72	0.2368
	(x_6,x_6)	13	52	0.11407	18	72	0.050688
	(x_7,x_7)	–	–	–	17	68	0.044552
10,000	(x_1,x_2)	9	36	0.49181	16	64	0.090626
	(x_2,x_2)	11	44	0.14241	17	68	0.10902
	(x_3,x_3)	11	44	0.26172	17	68	0.74438
	(x_4,x_4)	13	52	0.38111	18	72	0.17919
	(x_5,x_5)	12	48	0.17741	20	80	0.10613
	(x_6,x_6)	13	52	0.083221	19	76	0.12769
	(x_7,x_7)	–	–	–	18	72	0.093064
50,000	(x_1,x_2)	10	40	0.27835	17	68	0.32105
	(x_2,x_2)	11	44	0.24875	17	68	0.30922
	(x_3,x_3)	12	48	0.35821	18	72	0.87847
	(x_4,x_4)	13	52	0.4112	20	80	0.43688
	(x_5,x_5)	13	52	0.38957	21	84	0.46666
	(x_6,x_6)	13	52	0.48965	21	84	1.7167
	(x_7,x_7)	15	60	0.51664	18	72	1.4507
100,000	(x_1,x_2)	10	40	0.45116	17	68	0.89372
	(x_2,x_2)	11	44	1.1194	18	72	0.86119
	(x_3,x_3)	13	52	1.804	19	76	0.8354
	(x_4,x_4)	14	56	1.0128	22	88	1.1409
	(x_5,x_5)	15	60	1.0246	22	88	0.96899
	(x_6,x_6)	15	60	0.93187	22	88	1.06
	(x_7,x_7)	15	60	1.0907	20	80	1.0702

Table 5 Test result for Problem 4

DIM	(x_0, x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1, x_2)	2	6	0.047603	2	6	0.00627
	(x_2, x_2)	2	6	0.005977	2	6	0.003122
	(x_3, x_3)	2	6	0.004191	2	6	0.004575
	(x_4, x_4)	2	6	0.010495	2	6	0.003937
	(x_5, x_5)	2	6	0.005327	2	6	0.003846
	(x_6, x_6)	2	6	0.011041	2	6	0.002927
	(x_7, x_7)	2	6	0.005492	2	6	0.003468
5000	(x_1, x_2)	2	6	0.031877	2	6	0.096079
	(x_2, x_2)	2	6	0.014782	2	6	0.020704
	(x_3, x_3)	2	6	0.017518	2	6	0.011814
	(x_4, x_4)	2	6	0.056106	2	6	0.008442
	(x_5, x_5)	2	6	0.033729	2	6	0.01061
	(x_6, x_6)	2	6	0.013977	2	6	0.018209
	(x_7, x_7)	2	6	0.019215	2	6	0.011608
10,000	(x_1, x_2)	2	6	0.037551	2	6	0.078449
	(x_2, x_2)	2	6	0.025921	2	6	0.054769
	(x_3, x_3)	2	6	0.02794	2	6	0.013808
	(x_4, x_4)	2	6	0.017372	2	6	0.045487
	(x_5, x_5)	2	6	0.14848	2	6	0.050758
	(x_6, x_6)	2	6	0.089886	2	6	0.018096
	(x_7, x_7)	2	6	0.21488	2	6	0.016159
50,000	(x_1, x_2)	2	6	0.11667	2	6	0.076694
	(x_2, x_2)	2	6	0.087319	2	6	0.06213
	(x_3, x_3)	2	6	0.093389	2	6	0.060762
	(x_4, x_4)	2	6	0.072841	2	6	0.045131
	(x_5, x_5)	2	6	0.17981	2	6	0.096853
	(x_6, x_6)	2	6	0.19816	2	6	0.11699
	(x_7, x_7)	2	7	0.12229	2	7	0.08756
100,000	(x_1, x_2)	2	6	0.42528	2	6	0.15701
	(x_2, x_2)	2	6	0.27468	2	6	0.18368
	(x_3, x_3)	2	6	0.29093	2	6	0.37375
	(x_4, x_4)	2	6	0.14828	2	6	0.16735
	(x_5, x_5)	2	6	0.22	2	6	0.08926
	(x_6, x_6)	2	6	0.23811	2	6	0.12826
	(x_7, x_7)	2	7	0.40503	2	7	0.17219

Table 6 Test result for Problem 5

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	9	36	0.054204	15	60	0.014197
	(x_2,x_2)	10	40	0.017794	16	64	0.014104
	(x_3,x_3)	11	44	0.016521	16	64	0.019131
	(x_4,x_4)	11	44	0.024269	15	60	0.013714
	(x_5,x_5)	11	44	0.014339	17	68	0.013673
	(x_6,x_6)	11	44	0.01843	17	68	0.015563
	(x_7,x_7)	21	84	0.074477	17	68	0.012997
5000	(x_1,x_2)	9	36	0.089441	16	64	0.055828
	(x_2,x_2)	11	44	0.12174	16	64	0.034553
	(x_3,x_3)	11	44	0.034213	17	68	0.39755
	(x_4,x_4)	11	44	0.094632	16	64	0.059572
	(x_5,x_5)	12	48	0.26357	17	68	0.039368
	(x_6,x_6)	12	48	0.099053	19	76	0.055029
	(x_7,x_7)	22	88	0.15821	18	72	0.085236
10,000	(x_1,x_2)	9	36	0.27567	16	64	0.17337
	(x_2,x_2)	11	44	0.10903	17	68	0.081115
	(x_3,x_3)	11	44	0.13007	17	68	0.058728
	(x_4,x_4)	12	48	0.063544	19	76	0.1323
	(x_5,x_5)	12	48	0.062452	18	72	0.079782
	(x_6,x_6)	12	48	0.10251	19	76	0.6484
	(x_7,x_7)	24	96	0.20202	18	72	0.11402
50,000	(x_1,x_2)	10	40	0.22994	17	68	0.30098
	(x_2,x_2)	12	48	0.36531	17	68	0.23896
	(x_3,x_3)	13	52	0.23288	18	72	0.30252
	(x_4,x_4)	14	56	0.64547	20	80	0.40897
	(x_5,x_5)	15	60	0.31616	20	80	1.1317
	(x_6,x_6)	16	64	0.75731	22	88	0.45799
	(x_7,x_7)	21	84	0.47342	19	76	0.26783
100,000	(x_1,x_2)	10	40	0.41965	17	68	0.56148
	(x_2,x_2)	12	48	0.46951	18	72	0.60414
	(x_3,x_3)	12	48	0.42241	19	76	0.61453
	(x_4,x_4)	15	60	0.98056	22	88	0.81583
	(x_5,x_5)	15	60	1.0268	24	96	1.0713
	(x_6,x_6)	17	68	1.0367	26	104	1.1566
	(x_7,x_7)	20	80	0.81391	19	76	0.58601

Table 7 Test result for Problem 6

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	18	71	0.083017	19	75	0.012928
	(x_2,x_2)	20	79	0.031182	19	75	0.019364
	(x_3,x_3)	26	103	0.028924	20	79	0.023384
	(x_4,x_4)	24	96	0.032465	20	80	0.018849
	(x_5,x_5)	22	88	0.028834	20	80	0.020314
	(x_6,x_6)	30	120	0.077844	21	84	0.01811
	(x_7,x_7)	33	131	0.049431	28	111	0.026045
5000	(x_1,x_2)	23	91	0.12845	20	79	0.10752
	(x_2,x_2)	25	99	0.06932	20	79	0.080367
	(x_3,x_3)	27	107	0.11594	21	83	0.088736
	(x_4,x_4)	27	108	0.088566	21	84	0.066533
	(x_5,x_5)	22	88	0.078413	21	84	0.063351
	(x_6,x_6)	25	100	0.19629	21	84	0.061728
	(x_7,x_7)	42	167	0.24797	27	107	0.12475
10,000	(x_1,x_2)	25	99	0.49274	20	79	0.10424
	(x_2,x_2)	27	107	0.11762	20	79	0.087288
	(x_3,x_3)	25	99	0.22357	22	87	0.10489
	(x_4,x_4)	32	128	0.31481	23	92	0.12033
	(x_5,x_5)	21	84	0.15018	21	84	0.099959
	(x_6,x_6)	23	92	0.22802	21	84	0.12034
	(x_7,x_7)	37	148	0.34333	25	100	0.11022
50,000	(x_1,x_2)	27	108	0.95113	23	92	0.4633
	(x_2,x_2)	26	104	0.65128	23	92	0.4594
	(x_3,x_3)	24	96	0.60615	22	88	0.49417
	(x_4,x_4)	23	92	0.9149	24	96	0.59564
	(x_5,x_5)	34	136	1.7539	24	96	0.47907
	(x_6,x_6)	40	160	1.661	23	92	0.56435
	(x_7,x_7)	30	120	1.2256	26	104	0.62177
100,000	(x_1,x_2)	27	108	1.5181	24	96	1.0094
	(x_2,x_2)	23	92	1.1859	24	96	0.95376
	(x_3,x_3)	35	140	2.2552	23	92	0.91113
	(x_4,x_4)	38	152	2.6421	25	100	1.0668
	(x_5,x_5)	34	136	2.9522	25	100	1.0427
	(x_6,x_6)	44	176	3.966	26	104	1.1329
	(x_7,x_7)	24	96	1.4499	24	96	1.0176

Table 8 Test result for Problem 7

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	11	44	0.068624	18	72	0.030227
	(x_2,x_2)	11	44	0.023476	18	72	0.023403
	(x_3,x_3)	11	44	0.017342	18	72	0.020935
	(x_4,x_4)	11	44	0.025539	17	68	0.017201
	(x_5,x_5)	11	44	0.022939	17	68	0.024173
	(x_6,x_6)	11	44	0.019481	17	68	0.026493
	(x_7,x_7)	11	44	0.017316	18	72	0.039273
5000	(x_1,x_2)	13	52	0.16108	19	76	0.098763
	(x_2,x_2)	13	52	0.077516	19	76	0.096008
	(x_3,x_3)	13	52	0.072916	18	72	0.12134
	(x_4,x_4)	12	48	0.11408	18	72	0.087261
	(x_5,x_5)	11	44	0.06039	18	72	0.14077
	(x_6,x_6)	11	44	0.19888	17	68	0.15511
	(x_7,x_7)	13	52	0.093287	18	72	0.22382
10,000	(x_1,x_2)	10	40	0.14961	21	84	0.22682
	(x_2,x_2)	10	40	0.17445	21	84	0.16402
	(x_3,x_3)	14	56	0.12871	20	80	0.17297
	(x_4,x_4)	13	52	0.1861	18	72	0.12425
	(x_5,x_5)	12	48	0.147	18	72	0.21042
	(x_6,x_6)	11	44	0.096169	18	72	0.27754
	(x_7,x_7)	14	56	0.16325	20	80	0.14635
50,000	(x_1,x_2)	18	72	1.0847	24	96	1.6976
	(x_2,x_2)	16	64	1.0441	24	96	0.96248
	(x_3,x_3)	16	64	0.9691	23	92	0.99229
	(x_4,x_4)	15	60	0.7331	21	84	0.8611
	(x_5,x_5)	10	40	0.43613	21	84	0.84527
	(x_6,x_6)	13	52	0.48876	18	72	0.92241
	(x_7,x_7)	16	64	0.94373	23	92	0.98198
100,000	(x_1,x_2)	21	84	3.0149	29	116	2.8437
	(x_2,x_2)	20	80	2.8402	28	112	2.6468
	(x_3,x_3)	19	76	3.7723	26	104	2.3032
	(x_4,x_4)	16	64	2.2599	23	92	1.9833
	(x_5,x_5)	14	56	2.5742	22	88	1.8357
	(x_6,x_6)	10	40	1.6625	20	80	1.5495
	(x_7,x_7)	19	76	4.7247	26	104	2.3423

Table 9 Test result for Problem 8

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	13	52	0.047278	17	68	0.017428
	(x_2,x_2)	12	48	0.018759	17	68	0.017513
	(x_3,x_3)	5	20	0.037937	5	20	0.006891
	(x_4,x_4)	7	28	0.01145	18	72	0.013338
	(x_5,x_5)	13	52	0.018218	19	76	0.017099
	(x_6,x_6)	13	51	0.019231	18	71	0.024026
	(x_7,x_7)	20	80	0.072003	19	76	0.024216
5000	(x_1,x_2)	13	52	0.1766	18	72	0.31792
	(x_2,x_2)	13	52	0.062472	17	68	0.071077
	(x_3,x_3)	5	20	0.042518	5	20	0.016293
	(x_4,x_4)	7	28	0.18377	19	76	0.075083
	(x_5,x_5)	14	56	0.082247	20	80	0.05464
	(x_6,x_6)	14	55	0.051198	19	75	0.071584
	(x_7,x_7)	21	84	0.091359	19	76	0.49819
10,000	(x_1,x_2)	13	52	0.071794	18	72	0.34353
	(x_2,x_2)	13	52	0.096188	18	72	0.10846
	(x_3,x_3)	5	20	0.08997	5	20	0.026613
	(x_4,x_4)	7	28	0.049831	20	80	0.094635
	(x_5,x_5)	14	56	0.1608	20	80	0.13325
	(x_6,x_6)	14	56	0.10315	21	84	0.23725
	(x_7,x_7)	22	88	0.21037	20	80	0.14349
50,000	(x_1,x_2)	14	56	1.3873	19	76	0.8772
	(x_2,x_2)	13	52	0.48713	19	76	0.39603
	(x_3,x_3)	6	24	0.19012	5	20	0.12496
	(x_4,x_4)	8	32	0.212	21	84	0.51621
	(x_5,x_5)	15	60	0.47484	21	84	0.55585
	(x_6,x_6)	15	60	0.46325	21	84	0.56854
	(x_7,x_7)	23	92	0.71778	21	84	0.58178
100,000	(x_1,x_2)	8	32	0.38132	20	80	0.90233
	(x_2,x_2)	14	56	0.84004	19	76	0.75693
	(x_3,x_3)	6	24	0.26154	5	20	0.18686
	(x_4,x_4)	16	64	1.0809	22	88	2.3305
	(x_5,x_5)	16	64	1.0547	22	88	1.1825
	(x_6,x_6)	16	64	1.0315	22	88	1.4144
	(x_7,x_7)	24	96	1.6254	20	80	1.1643

Table 10 Test result for Problem 9

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	26	104	0.25727	36	144	0.23507
	(x_2,x_2)	25	100	0.34424	35	140	0.20293
	(x_3,x_3)	20	80	0.23255	35	140	0.18885
	(x_4,x_4)	19	76	0.20691	33	132	0.29419
	(x_5,x_5)	16	64	0.19554	31	124	0.18894
	(x_6,x_6)	27	108	0.28929	24	96	0.19307
	(x_7,x_7)	18	72	0.15421	27	108	0.18974
5000	(x_1,x_2)	16	64	0.66807	34	136	0.96072
	(x_2,x_2)	20	80	1.0047	34	136	0.98855
	(x_3,x_3)	18	72	0.74354	34	136	0.91487
	(x_4,x_4)	19	76	0.83177	31	124	0.92888
	(x_5,x_5)	17	68	0.77948	30	120	0.80674
	(x_6,x_6)	24	96	1.1869	24	96	0.66125
	(x_7,x_7)	18	72	0.85028	27	108	0.75491
10,000	(x_1,x_2)	20	80	1.6479	34	136	1.6326
	(x_2,x_2)	30	120	2.6217	34	136	1.7051
	(x_3,x_3)	24	96	1.8867	33	132	1.6473
	(x_4,x_4)	16	64	1.3469	30	120	1.544
	(x_5,x_5)	17	68	1.4065	30	120	1.5837
	(x_6,x_6)	27	108	2.2891	24	96	1.2598
	(x_7,x_7)	18	72	1.4826	28	112	1.6064
50,000	(x_1,x_2)	26	104	12.2321	34	136	7.4057
	(x_2,x_2)	20	80	6.9478	33	132	7.1126
	(x_3,x_3)	17	68	5.6733	32	128	6.9792
	(x_4,x_4)	19	76	6.8442	24	96	5.7097
	(x_5,x_5)	18	72	6.3233	29	116	6.4302
	(x_6,x_6)	19	76	6.8044	31	124	6.8804
	(x_7,x_7)	20	80	7.1252	26	104	5.7188
100,000	(x_1,x_2)	33	132	23.4997	33	132	14.5783
	(x_2,x_2)	20	80	13.6641	33	132	14.6661
	(x_3,x_3)	21	84	14.5443	40	160	18.5338
	(x_4,x_4)	18	72	12.7254	30	120	13.2054
	(x_5,x_5)	19	76	13.7186	28	112	12.9061
	(x_6,x_6)	28	112	20.2095	26	104	11.7886
	(x_7,x_7)	19	76	14.2087	28	112	12.1064

Table 11 Test result for Problem 10

DIM	(x_0,x_1)	IPDY			PDY		
		NI	NF	CPU	NI	NF	CPU
1000	(x_1,x_2)	8	29	0.031115	11	42	0.013176
	(x_2,x_2)	8	30	0.011849	11	42	0.007171
	(x_3,x_3)	8	30	0.007197	11	42	0.014527
	(x_4,x_4)	8	30	0.013392	11	42	0.010997
	(x_5,x_5)	8	30	0.011848	11	42	0.013188
	(x_6,x_6)	8	30	0.011115	12	46	0.010747
	(x_7,x_7)	8	30	0.012579	11	42	0.014138
5000	(x_1,x_2)	8	31	0.084149	8	31	0.13649
	(x_2,x_2)	8	31	0.052078	8	31	0.033025
	(x_3,x_3)	8	31	0.054572	8	31	0.072224
	(x_4,x_4)	8	31	0.071554	9	35	0.044811
	(x_5,x_5)	9	35	0.075278	9	35	0.042668
	(x_6,x_6)	9	35	0.29077	9	35	0.049779
	(x_7,x_7)	8	31	0.10995	8	31	0.29745
10,000	(x_1,x_2)	8	31	0.11789	11	43	0.29977
	(x_2,x_2)	8	31	0.12087	11	43	0.11724
	(x_3,x_3)	8	31	0.50803	11	43	0.1141
	(x_4,x_4)	9	35	0.47272	12	47	0.23368
	(x_5,x_5)	9	35	0.25186	13	51	0.18279
	(x_6,x_6)	10	39	0.22833	13	51	0.19171
	(x_7,x_7)	8	31	0.11345	11	43	0.10251
50,000	(x_1,x_2)	8	32	0.54884	10	40	0.506
	(x_2,x_2)	8	32	0.57793	10	40	1.0076
	(x_3,x_3)	9	36	1.006	11	44	0.69436
	(x_4,x_4)	11	44	1.5944	13	52	1.0485
	(x_5,x_5)	11	44	1.9997	14	56	1.3131
	(x_6,x_6)	13	52	2.2809	16	64	1.5874
	(x_7,x_7)	9	36	0.90394	11	44	0.5891
100,000	(x_1,x_2)	8	32	1.1894	9	36	0.95893
	(x_2,x_2)	8	32	1.3967	9	36	0.96424
	(x_3,x_3)	9	36	2.0912	11	44	1.6021
	(x_4,x_4)	12	48	4.044	14	56	2.5748
	(x_5,x_5)	13	52	4.7375	16	64	3.3814
	(x_6,x_6)	15	60	6.0328	18	72	4.1306
	(x_7,x_7)	10	40	2.4042	11	44	1.5925

Acknowledgements
We are grateful to the anonymous referees for their useful comments which have made the paper clearer and more comprehensive than the earlier version. The first author was supported by the "Petchra Pra Jom Klao Ph.D. Research Scholarship from King Mongkut's University of Technology Thonburi" (Grant no. 16/2561). The authors also acknowledge the financial support provided by the Center of Excellence in Theoretical and Computational Science (TaCS-CoE), KMUTT. Moreover, this research project is supported by National Council of Thailand (NRCT) under Research Grants for Talented Mid-Career Researchers (Contract no. N41A640089). The third author acknowledges with thanks the Department of Mathematics and Applied Mathematics at the Sefako Makgatho Health Sciences University.

Funding
No funding received.

Availability of data and materials
Not applicable.

Declarations

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
All authors contributed equally to the manuscript and read and approved the final manuscript.

Author details
[1]KMUTTFixed Point Research Laboratory, Room SCL 802 Fixed Point Laboratory, Science Laboratory Building, Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thung Khru, Bangkok 10140, Thailand. [2]Center of Excellence in Theoretical and Computational Science (TaCS-CoE), Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha Uthit Rd., Bang Mod, Thung Khru, Bangkok 10140, Thailand. [3]Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan. [4]Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University Kano, Kano, Nigeria. [5]Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, Medunsa 0204, South Africa. [6]Department of Mathematics, Usmanu Danfodiyo University, Sokoto, Nigeria.

Publisher's Note

Received: 16 April 2021 Accepted: 26 October 2021 Published online: 06 December 2021

References

1. Liu, J., Feng, Y.: A derivative-free iterative method for nonlinear monotone equations with convex constraints. Numer. Algorithms **82**(1), 245–262 (2019)
2. Minty, G.J.: Monotone networks. Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci. **257**(1289), 194–212 (1960)
3. Aj, W., Wollenberg, B.: Power Generation, Operation and Control, p. 592. Wiley, New York (1996)
4. Dirkse, S.P., Ferris, M.C.: MCPLIB: a collection of nonlinear mixed complementarity problems. Optim. Methods Softw. **5**(4), 319–345 (1995)
5. Meintjes, K., Morgan, A.P.: A methodology for solving chemical equilibrium systems. Appl. Math. Comput. **22**(4), 333–361 (1987)
6. Figueiredo, M.A., Nowak, R.D., Wright, S.J.: Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Signal Process. **1**(4), 586–597 (2007)
7. Abubakar, A.B., Kumam, P., Ibrahim, A.H., Chaipunya, P., Rano, S.A.: New hybrid three-term spectral-conjugate gradient method for finding solutions of nonlinear monotone operator equations with applications. Math. Comput. Simul. (2021, in press)
8. Ibrahim, A.H., Deepho, J., Bala Abubakar, A., Adamu, A.: A three-term Polak–Ribière–Polyak derivative-free method and its application to image restoration. Sci. Afr. **13**, e00880 (2021). https://www.sciencedirect.com/science/article/pii/S2468227621001848
9. Ibrahim, A.H., Kumam, P., Hassan, B.A., Abubakar, A.B., Abubakar, J.: A derivative-free three-term Hestenes–Stiefel type method for constrained nonlinear equations and image restoration. Int. J. Comput. Math. (2021). https://doi.org/10.1080/00207160.2021.1946043
10. Ibrahim, A.H., Deepho, J., Abubakar, A.B., Aremu, K.O.: A modified Liu–Storey-conjugate descent hybrid projection method for convex constrained nonlinear equations and image restoration. Numer. Algebra Control Optim. (2021). https://doi.org/10.3934/naco.2021022
11. Ibrahim, A.H., Garba, A.I., Usman, H., Abubakar, J., Abubakar, A.B.: Derivative-free RMIL conjugate gradient method for convex constrained equations. Thai J. Math. **18**(1), 212–232 (2019)
12. Abubakar, A.B., Rilwan, J., Yimer, S.E., Ibrahim, A.H., Ahmed, I.: Spectral three-term conjugate descent method for solving nonlinear monotone equations with convex constraints. Thai J. Math. **18**(1), 501–517 (2020)
13. Ibrahim, A.H., Kumam, P., Abubakar, A.B., Jirakitpuwapat, W., Abubakar, J.: A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing. Heliyon **6**(3), e03466 (2020)
14. Ibrahim, A.H., Kumam, P., Abubakar, A.B., Abubakar, J., Muhammad, A.B.: Least-square-based three-term conjugate gradient projection method for ℓ_1-norm problems with application to compressed sensing. Mathematics **8**(4), 602 (2020)
15. Ibrahim, A.H., Kumam, P., Abubakar, A.B., Yusuf, U.B., Rilwan, J.: Derivative-free conjugate residual algorithms for convex constraints nonlinear monotone equations and signal recovery. J. Nonlinear Convex Anal. **21**(9), 1959–1972 (2020)
16. Abubakar, A.B., Ibrahim, A.H., Muhammad, A.B., Tammer, C.: A modified descent Dai–Yuan conjugate gradient method for constraint nonlinear monotone operator equations. Appl. Anal. Optim. **4**, 1–24 (2020)
17. Abubakar, A.B., Kumam, P., Ibrahim, A.H., Rilwan, J.: Derivative-free HS–DY-type method for solving nonlinear equations and image restoration. Heliyon **6**(11), e05400 (2020)
18. Ibrahim, A.H., Kumam, P., Kumam, W.: A family of derivative-free conjugate gradient methods for constrained nonlinear equations and image restoration. IEEE Access **8**, 162714–162729 (2020)
19. Ibrahim, A.H., Kumam, P., Abubakar, A.B., Yusuf, U.B., Yimer, S.E., Aremu, K.O.: An efficient gradient-free projection algorithm for constrained nonlinear equations and image restoration. AIMS Math. **6**(1), 235 (2020)
20. Abubakar, A.B., Muangchoo, K., Ibrahim, A.H., Muhammad, A.B., Jolaoso, L.O., Aremu, K.O.: A new three-term Hestenes–Stiefel type method for nonlinear monotone operator equations and image restoration. IEEE Access **9**, 18262–18277 (2021)
21. Ibrahima, A.H., Muangchoob, K., Mohamedc, N.S., Abubakard, A.B.: Derivative-free SMR conjugate gradient method for constraint nonlinear equations. J. Math. Comput. Sci. **24**(2), 147–164 (2022)
22. Abubakar, A.B., Muangchoo, K., Ibrahim, A.H., Abubakar, J., Rano, S.A.: FR-type algorithm for finding approximate solutions to nonlinear monotone operator equations. Arab. J. Math. **10**, 261–270 (2021)
23. Abubakar, A.B., Kumam, P., Mohammad, H., Ibrahim, A.H.: PRP-like algorithm for monotone operator equations. Jpn. J. Ind. Appl. Math. **38**, 805–822 (2021)
24. Ibrahim, A.H., Muangchoo, K., Abubakar, A.B., Adedokun, A.D., Spectral, M.H.: Conjugate gradient like method for signal reconstruction. Thai J. Math. **18**(4), 2013–2022 (2020)
25. Ibrahim, A.H., Kumam, P.: Re-modified derivative-free iterative method for nonlinear monotone equations with convex constraints. Ain Shams Eng. J. **12**(2), 2205–2210 (2021)

26. Mohammad, H.: Barzilai–Borwein-like method for solving large-scale non-linear systems of equations. J. Niger. Math. Soc. **36**(1), 71–83 (2017)
27. Abubakar, A.B., Kumam, P.: A descent Dai–Liao conjugate gradient method for nonlinear equations. Numer. Algorithms **81**(1), 197–210 (2019)
28. Abubakar, A.B., Kumam, P.: An improved three-term derivative-free method for solving nonlinear equations. Comput. Appl. Math. **37**(5), 6760–6773 (2018)
29. Abubakar, A.B., Muangchoo, K., Ibrahim, A.H., Fadugba, S.E., Aremu, K.O., Jolaoso, L.O.: A modified scaled spectral-conjugate gradient-based algorithm for solving monotone operator equations. J. Math. **2021**, Article ID 5549878 (2021)
30. Waziri, M.Y., Ahmed, K., Sabi'u, J.: A family of Hager–Zhang conjugate gradient methods for system of monotone nonlinear equations. Appl. Math. Comput. **361**, 645–660 (2019)
31. Waziri, M., Ahmed, K., Sabi'u, J.: A Dai–Liao conjugate gradient method via modified secant equation for system of nonlinear equations. Arab. J. Math. **9**, 443–457 (2020)
32. 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. Methods **2020**, 2050043 (2020)
33. Waziri, M.Y., Hungu, K.A., Descent, S.J.: Perry conjugate gradient methods for systems of monotone nonlinear equations. Numer. Algorithms **85**(3), 763–785 (2020)
34. Waziri, M.Y., Muhammad, H.U., Halilu, A.S., Ahmed, K.: Modified matrix-free methods for solving system of nonlinear equations. Optimization (2020). https://doi.org/10.1080/02331934.2020.1778689
35. 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. Simul. **187**, 520–539 (2021)
36. Dennis, J.E., Moré, J.J.: A characterization of superlinear convergence and its application to quasi-Newton methods. Math. Comput. **28**(126), 549–560 (1974)
37. Li, D., Fukushima, M.: A globally and superlinearly convergent Gauss–Newton-based BFGS method for symmetric nonlinear equations. SIAM J. Numer. Anal. **37**(1), 152–172 (1999)
38. Zhou, G., Toh, K.C.: Superlinear convergence of a Newton-type algorithm for monotone equations. J. Optim. Theory Appl. **125**(1), 205–221 (2005)
39. Zhou, W.J., Li, D.H.: A globally convergent BFGS method for nonlinear monotone equations without any merit functions. Math. Comput. **77**(264), 2231–2240 (2008)
40. Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**(1), 177–182 (1999)
41. Chen, P., Huang, J., Zhang, X.: A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration. Inverse Probl. **29**(2), 025011 (2013)
42. Iiduka, H.: Iterative algorithm for triple-hierarchical constrained nonconvex optimization problem and its application to network bandwidth allocation. SIAM J. Optim. **22**(3), 862–878 (2012)
43. Jolaoso, L.O., Alakoya, T., Taiwo, A., Mewomo, O.: Inertial extragradient method via viscosity approximation approach for solving equilibrium problem in Hilbert space. Optimization **70**(2), 387–412 (2021)
44. Abubakar, J., Kumam, P., Ibrahim, A.H., Relaxed, P.A.: Inertial Tseng's type method for solving the inclusion problem with application to image restoration. Mathematics **8**(5), 818 (2020)
45. Abubakar, J., Kumam, P., Ibrahim, A.H., et al.: Inertial iterative schemes with variable step sizes for variational inequality problem involving pseudomonotone operator. Mathematics **8**(4), 609 (2020)
46. Abubakar, J., Sombut, K., Ibrahim, A.H., et al.: An accelerated subgradient extragradient algorithm for strongly pseudomonotone variational inequality problems. Thai J. Math. **18**(1), 166–187 (2019)
47. Alvarez, F.: Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert space. SIAM J. Optim. **14**(3), 773–782 (2004)
48. Alvarez, F., Attouch, H.: An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping. Set-Valued Anal. **9**(1), 3–11 (2001)
49. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys. **4**(5), 1–17 (1964)
50. Ogwo, G., Izuchukwu, C., Mewomo, O.: Inertial methods for finding minimum-norm solutions of the split variational inequality problem beyond monotonicity. Numer. Algorithms **88**, 1419–1456 (2021)
51. Takahashi, W.: Introduction to Nonlinear and Convex Analysis. Yokohama Publishers, Yokohama (2009)
52. Auslender, A., Teboulle, M., Ben-Tiba, S.: A logarithmic-quadratic proximal method for variational inequalities. In: Computational Optimization, pp. 31–40. Springer, Berlin (1999)
53. La Cruz, W., Martínez, J.M., Raydan, M.: Spectral residual method without gradient information for solving large-scale nonlinear systems: theory and experiments. Citeseer. Technical report RT-04-08 (2004). https://www.ime.unicamp.br/~martinez/lmrreport.pdf
54. Li, Q., Li, D.H.: A class of derivative-free methods for large-scale nonlinear monotone equations. IMA J. Numer. Anal. **31**(4), 1625–1635 (2011)
55. La Cruz, W.: A spectral algorithm for large-scale systems of nonlinear monotone equations. Numer. Algorithms **76**(4), 1109–1130 (2017)
56. Yu, Z., Lin, J., Sun, J., Xiao, Y., Liu, L., Li, Z.: Spectral gradient projection method for monotone nonlinear equations with convex constraints. Appl. Numer. Math. **59**(10), 2416–2423 (2009)
57. Lukšan, L., Matonoha, C., Vlcek, J.: Problems for nonlinear least squares and nonlinear equations. Technical report (2018)
58. Ibrahim, A.H., Kumam, P., Abubakar, A.B., Jirakitpuwapat, W., Abubakar, J.: A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing. Heliyon **6**(3), e03466 (2020)
59. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)