Nonlinear Functional Analysis and Applications
Vol. 26, No. 4 (2021), pp. 839-853
ISSN: 1229-1595(print), 2466-0973(online)

https://doi.org/10.22771/nfaa.2021.26.04.11
http://nfaa.kyungnam.ac.kr/journal-nfaa

A THREE-TERM INERTIAL DERIVATIVE-FREE PROJECTION METHOD FOR CONVEX CONSTRAINED MONOTONE EQUATIONS

Supansa Noinakorn[1], Abdukarim Hassan Ibrahim[2], Auwal Bala Abubakar[3,4] and Nuttapol Pakkaranang[1]

[1]Mathematics and Computing Science Program, Faculty of Science and Technology
Phetchabun Rajabhat University, Phetchabun 67000, Thailand
e-mail: supansa.noi@pcru.ac.th

[2]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
e-mail: ibrahimkarym@gmail.com

[3]Department of Mathematical Sciences, Faculty of Physical Sciences
Bayero University Kano, Kano, Nigeria
e-mail: ababubakar.mth@buk.edu.ng

[4]Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, Pretoria, Medunsa-0204, South Africa
e-mail: ababubakar.mth@buk.edu.ng

Abstract. Let $\mathfrak{R}^n$ be an Euclidean space and $g : \mathfrak{R}^n \to \mathfrak{R}^n$ be a monotone and continuous mapping. Suppose the convex constrained nonlinear monotone equation problem $x \in \mathfrak{C}$ s.t $g(x) = 0$ has a solution. In this paper, we construct an inertial-type algorithm based on the three-term derivative-free projection method (TTMDY) for convex constrained monotone nonlinear equations. Under some standard assumptions, we establish its global convergence to a solution of the convex constrained nonlinear monotone equation. Furthermore, the proposed algorithm converges much faster than the existing non-inertial algorithm (TTMDY) for convex constrained monotone equations.

[0]Received May 6, 2021. Revised October 20, 2021. Accepted November 2, 2021.
[0]2010 Mathematics Subject Classification: 90C06, 90C56, 65K05, 65K10.
[0]Keywords: Iterative method, inertial algorithm, nonlinear equations, derivative-free method, projection method.
[0]Corresponding author: N. Pakkaranang(nuttapol.pak@pcru.ac.th).

1. Introduction

Projection method is one of the most common and efficient methods for solving large-scale nonlinear equations. In this paper, our interest is on the monotone nonlinear equations with convex constraints, that is

$$\text{find } x \in \mathfrak{C} \text{ s.t } g(x) = 0, \tag{1.1}$$

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

Chemical equilibrium systems [31], economic equilibrium problems [13], and power flow systems [34] are only a few examples of practical problems that can be converted into convex constraint nonlinear monotone equations (1.1). This is why there have been keen interest in solving the problem (1.1) [35]. In solving unconstrained optimization problem [1, 2], the three-term conjugate gradient method is commonly used because of its descent property, computational efficiency, and stable convergence. Many researchers are now turning to the frameworks of three-term conjugate gradient methods to solve nonlinear equations (1.1). For example, based on the projection technique [33], Cao [12] introduced a three-term derivative-free method for large-scale nonlinear equations, which is based on the structures of the famous Dai-Yuan (DY) conjugate gradient method, and the three-term conjugate gradient method of Gao and He [15]. This method inherits the stability of the DY method, and greatly improves its computing performance. For more algorithms for solving (1.1), see [3]-[9], [16]-[28].

In recent years, it has always been of great interest to speed up the convergence of iterative algorithms. The addition of inertial terms to algorithms is one of the most recent methods of speeding up convergence. The use of "inertial" term can be traced back to Polyak [32], who studied the following second-order system of differential equations:

$$v'' + \gamma v' + \nabla f(v) = 0, \ \gamma > 0 \tag{1.2}$$

in the context of optimization. In two-dimensional case, system (1.2) describes, roughly, the motion of a heavy ball that rolls under its own inertial over the graph of f until it is impeded by friction. For results concerning inertial algorithms, see [10, 11, 29, 30].

Our interest in this paper is to introduce an inertial algorithm for finding the solutions of the convex constraint nonlinear monotone equation. The proposed method combines the inertial term with the three-term derivative-free projection method (TTMDY) for convex constrained monotone nonlinear equations [12]. Our algorithm converges much faster than the existing non-inertial algorithm (TTMDY). We give numerical examples to support this claim. Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm of a vector.

2. Preliminaries and algorithm

In the sequel, we first give some well-known concepts that will be used.

Definition 2.1. A mapping $g : \mathfrak{R}^n \to \mathfrak{R}^n$ is called

(a) monotone on $\mathfrak{R}^n$ if

$$(g(x) - g(y))^T (x - y) \geq 0, \ \forall x, y \in \mathfrak{R}^n.$$

(b) L-Lipschitz continuous on $\mathfrak{R}^n$ if there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \ \forall x, y \in \mathfrak{R}^n.$$

To describe our algorithm, we also use the projection operator $P_{\mathfrak{C}}$ which is defined as follows:

For every element $x \in \mathfrak{R}^n$, there exists a unique nearest point in $\mathfrak{C}$, denoted by $P_{\mathfrak{C}}(x)$, such that

$$P_{\mathfrak{C}}(x) = \arg\min_{y \in \mathfrak{C}} \|x - y\|.$$

$P_{\mathfrak{C}}$ is called the orthogonal (or metric) projection of $\mathfrak{R}^n$ onto $\mathfrak{C}$. The metric projection $P_{\mathfrak{C}}$ has the following basic property:

$$\|P_{\mathfrak{C}}(x) - y\|^2 \leq \|x - y\|^2 - \|x - P_{\mathfrak{C}}(x)\|^2, \ \forall x \in \mathfrak{R}^n, \forall y \in \mathfrak{C}. \tag{2.1}$$

Lemma 2.2. ([28]) *Let $\mathfrak{R}^n$ be an Euclidean space. Then the following inequality hold:*

$$\|x + y\|^2 \leq \|x\|^2 + 2y^T (x + y).$$

Lemma 2.3. ([28]) *Let $\{x_k\}$ and $\{y_k\}$ be sequences of nonnegative real numbers satisfying the following relation*

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

where $\sum_{k=0}^{\infty} y_k < \infty$, then $\lim_{k \to \infty} x_k$ exists.

Lemma 2.4. ([28]) *A point q is in the solution set $\mathfrak{C}^*$ of (1.1) if and only if $q = P_{\mathfrak{C}}(q - \rho u)$ for some $u = g(q)$ and $\rho > 0$.*

In the following, based on the TTMDY method [12], we present the inertial TTMDY method. The specific steps of our proposed method are presented in the algorithm below.

Algorithm A: Inertial-TTMDY method (ITTMDY)

(S.0) Take the positive constants: $Tol > 0,\ r > 0\ \rho \in (0,1),\ \zeta > 0, \sigma > 0,$ $\psi_k \in [0,1)$. Select arbitrary points x_{-1}, $x_0 \in \mathfrak{C}$. Set $k := 0$.

(S.1) Set $c_k = x_k + \psi_k(x_k - x_{k-1})$.

(S.2) Compute $g(c_k)$. If $\|g(c_k)\| \leq Tol$, stop. Otherwise, generate the search direction d_k by

$$d_k := \begin{cases} -g(c_k) & \text{if } k = 0, \\ -g(c_k) + \beta_k^{ImDY} d_{k-1} + \vartheta_k y_{k-1} & \text{if } k > 0, \end{cases} \tag{2.2}$$

where,

$$\beta_k^{ImDY} := \frac{\|g(c_k)\|^2}{d_{k-1}^T w_{k-1}},\ \vartheta_k := \frac{g(c_k)^T d_{k-1}}{d_{k-1}^T w_{k-1}},$$
$$y_{k-1} := g(c_k) - g(c_{k-1}),$$
$$w_{k-1} := y_{k-1} + t_{k-1} d_{k-1},\ t_{k-1} := r\frac{\|g(c_{k-1})\|}{\|d_{k-1}\|} + \max\left\{0, -\frac{d_{k-1}^T y_{k-1}}{\|d_{k-1}\|^2}\right\}. \tag{2.3}$$

(S.3) Determine the step-size $\varepsilon_k = \zeta\rho^i$ where i is the least nonnegative integer satisfying

$$-g(c_k + \varepsilon_k d_k)^T d_k \geq \sigma\varepsilon_k \|d_k\|^2. \tag{2.4}$$

(S.4) Compute $v_k = c_k + \varepsilon_k d_k$, where v_k is a trial point.

(S.5) If $v_k \in \mathfrak{C}$ and $\|g(v_k)\| \leq Tol$, stop. Otherwise, compute the next iterate by

$$x_{k+1} = P_{\mathfrak{C}}\left[c_k - \lambda_k g(v_k)\right], \tag{2.5}$$

where

$$\lambda_k := \frac{g(v_k)^T(c_k - v_k)}{\|g(v_k)\|^2}.$$

(S.6) Set $k \leftarrow k+1$, and return to (S.1).

Next, we give the assumptions that we will use throughout this paper.

Assumption 2.5. *Suppose that the following conditions hold:*

(a) *The feasible set* $\mathfrak{C}$ *is a nonempty, closed and convex subset of the Euclidean space* $\mathfrak{R}^n$.

(b) $g : \mathfrak{R}^n \to \mathfrak{R}^n$ *is monotone and L-Lipschitz continuous.*
(c) *The solution set* $\mathfrak{C}^*$ *of* (1.1) *is nonempty.*

Assumption 2.6. *Let* $\{\psi_k\}$ *be a sequence of nonnegative real number satisfying the condition:*

$$\sum_{k=0}^{\infty} \psi_k \|x_k - x_{k-1}\| < \infty.$$

3. Main result

In this section, we will analyze the convergence of Algorithm A. We start with the following lemma which plays an important role in proving the convergence of the proposed algorithm.

Lemma 3.1. *Let* d_k *be generated by Algorithm A. Then,* d_k *always satisfies the sufficient descent condition, that is,*

$$g(c_k)^T d_k = -p\|g(c_k)\|^2, p > 0. \tag{3.1}$$

Proof. The proof is the same as that in [12]. Therefore we omit it. □

Lemma 3.2. *Let* $\{x_k\}$ *and* $\{v_k\}$ *be generated by Algorithm A. If* $q \in \mathfrak{C}^*$*, then under Assumption* 2.5 *and* 2.6*, it holds that*

$$\|x_{k+1} - q\|^2 \leq \|c_k - q\| - \frac{\sigma^2 \|c_k - v_k\|^4}{\|g(v_k)\|^2}. \tag{3.2}$$

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

$$\sum_{k=0}^{\infty} \|c_k - v_k\|^4 < \infty. \tag{3.3}$$

Proof. By the monotonicity of the mapping g, we have

$$\begin{aligned} g(v_k)^T (c_k - q) &= g(v_k)^T (c_k - v_k) + g(v_k)^T (v_k - q) \\ &\geq g(v_k)^T (c_k - v_k) + g(q)^T (v_k - q) \\ &= g(v_k)^T (c_k - v_k) \\ &\geq \sigma \|c_k - v_k\|^2. \end{aligned} \tag{3.4}$$

From the nonexpansiveness of the projection operator and (3.4), it holds that for any $q \in \mathfrak{C}^*$,

$$\begin{aligned}
\|x_{k+1} - q\|^2 &= \|P_{\mathfrak{C}}(c_k + \lambda_k g(v_k)) - q\|^2 \\
&\leq \|c_k - \lambda_k g(v_k) - q\|^2 \\
&= \|c_k - q\|^2 - 2\lambda_k g(v_k)^T (c_k - q) + \lambda_k^2 \|g(v_k)\|^2 \\
&\leq \|c_k - q\|^2 - 2\lambda_k g(v_k)^T (c_k - v_k) + \lambda_k^2 \|g(v_k)\|^2 \\
&\leq \|c_k - q\|^2 - \frac{g(v_k)^T (c_k - v_k)^2}{\|g(v_k)\|^2} \\
&\leq \|c_k - q\|^2 - \frac{\sigma^2 \|c_k - v_k\|^4}{\|g(v_k)\|^2}. \qquad (3.5)
\end{aligned}$$

From (3.5), we can deduce that

$$\begin{aligned}
\|x_{k+1} - q\| &\leq \|c_k - q\| \\
&= \|x_k + \psi_k (x_k - x_{k-1}) - q\| \\
&\leq \|x_k - q\| + \psi_k \|x_k - x_{k-1}\|. \qquad (3.6)
\end{aligned}$$

Noting that $\sum_{k=0}^{\infty} \psi_k \|x_k - x_{k-1}\| < \infty$, we deduce that $\{x_k - q\}$ is bounded by a positive number say M_0. Thus, for all k, we have that

$$\|x_k - x_{k-1}\| \leq 2M_0.$$

Using the above bounds and the definition of c_k, we have

$$\begin{aligned}
\|c_k - q\|^2 &= \|x_k + \psi_k (x_k - x_{k-1}) - q\|^2 \\
&= \|x_k - q\|^2 + 2\psi_k (x_k - x_{k-1})^T (x_k + \psi_k (x_k - x_{k-1}) - q) \\
&\leq \|x_k - q\|^2 + 2\psi_k \|x_k - x_{k-1}\| (\|x_k - q\| + \psi_k \|x_k - x_{k-1}\|) \\
&\leq \|x_k - q\|^2 + 2M\psi_k \|x_k - x_{k-1}\| + 4M_0 \psi_k \|x_k - x_{k-1}\| \\
&= \|x_k - q\|^2 + 6M_0 \psi_k \|x_k - x_{k-1}\|. \qquad (3.7)
\end{aligned}$$

Combining (3.5) with (3.7), we have

$$\|x_{k+1} - q\|^2 \leq \|x_k - q\|^2 + 6M_0 \psi_k \|x_k - x_{k-1}\| - \sigma^2 \|c_k - v_k\|^4. \qquad (3.8)$$

Thus, we have Thus, we have

$$\sigma^2 \|c_k - v_k\|^4 \leq \|x_k - q\|^2 + 6M_0 \psi_k \|x_k - x_{k-1}\| - \|x_{k+1} - q\|^2. \qquad (3.9)$$

Adding (3.9) for $k = 0, 1, 2, \cdots$, we have

$$\sigma^2 \sum_{k=0}^{\infty} \|c_k - v_k\|^4 \leq \sum_{k=0}^{\infty} \left(\|x_k - q\|^2 + 6M_0 \psi_k \|x_k - x_{k-1}\| - \|x_{k+1} - q\|^2 \right).$$

Now, since

$$\sum_{k=0}^{\infty}\left(\|x_k - q\|^2 - \|x_{k+1} - q\|^2\right) = \|x_0 - q\|^2 < \infty$$

and

$$\infty\psi_k\|x_k - x_{k-1}\| < \infty,$$

it implies that

$$\begin{aligned}\sigma^2\sum_{k=0}^{\infty}\|c_k - v_k\|^4 &\leq \sum_{k=0}^{\infty}\left(\|x_k - q\|^2 - \|x_{k+1} - q\|^2 + 6M_0\psi_k\|x_k - x_{k-1}\|\right)\\ &< \infty.\end{aligned}$$

Therefore,

$$\lim_{k\to\infty}\|c_k - v_k\| = 0. \tag{3.10}$$

□

Remark 3.3. By the definition of $\{v_k\}$ and (3.3), we have

$$\lim_{k\to\infty}\varepsilon_k\|d_k\| = 0. \tag{3.11}$$

Lemma 3.4. *Suppose Assumptions* 2.5-2.6 *hold and the sequence* $\{x_k\}$ *and* $\{c_k\}$ *are generated by Algorithm A. Then we have*

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

Proof. We know that

$$\|x_k - c_k\| = \|x_k - (x_k + \psi_k(x_k - x_{k-1}))\| = \psi_k\|x_k - x_{k-1}\|.$$

Therefore,

$$\lim_{k\to\infty}\|x_k - c_k\| = 0. \tag{3.12}$$

Similarly, we have

$$\|x_k - v_k\| = \|x_k - c_k + c_k - v_k\| \leq \|x_k - c_k\| + \|c_k - v_k\|.$$

Thus, by (3.10) and (3.12), it follows that

$$\lim_{k\to\infty}\|x_k - v_k\| = 0. \tag{3.13}$$

From the nonexpansiveness of the projection operator, we have

$$\begin{aligned}
\|x_{k+1} - x_k\| &= \|P_{\mathfrak{C}}[c_k - \lambda_k g(v_k)] - x_k\| \\
&\leq \|c_k - \lambda_k g(v_k) - x_k\| \\
&\leq \|c_k - x_k\| + \|\lambda_k g(v_k)\| \\
&= \|c_k - x_k\| + \|\lambda_k g(v_k)\| \\
&= \|c_k - x_k\| + \left\| \frac{g(v_k)^T (c_k - v_k)}{\|g(v_k)\|^2} g(v_k) \right\| \\
&\leq \|c_k - x_k\| + \|c_k - v_k\|. \qquad (3.14)
\end{aligned}$$

Therefore,

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \qquad (3.15)$$

Thus, it is easy to see that

$$\begin{aligned}
\|x_{k+1} - c_k\| &= \|x_{k+1} - (x_k + \psi_k(x_k - x_{k-1}))\| \\
&\leq \|x_{k+1} - x_k\| + \psi_k \|x_k - x_{k-1}\|.
\end{aligned}$$

Therefore, we gets the desired result. □

Theorem 3.5. *Let $\{x_k\}$ be a sequence generated by Algorithm A. Then, under Assumption* 2.5 *and* 2.6*, we have that $\{x_k\}$ converges to an element of $\mathfrak{C}^*$.*

Proof. By the boundedness of $\{x_k\}$, it implies that there exists a subsequence $\{x_{k_j}\}$ of $\{x_k\}$ such that $\{x_{k_j}\}$ converges to some point $\bar{q}$. Also, we have that

$$\|c_{k_j} - x_{k_j}\| = \theta_{k_j} \|x_{k_j} - x_{k_j - 1}\| \to 0, \text{ as } k \to \infty. \qquad (3.16)$$

Claim: $\bar{q} \in \mathfrak{C}^*$. Suppose by contradiction, $\bar{q} \notin \mathfrak{C}^*$. Then by Lemma 3.4 and (3.16)

$$\lim_{j\to\infty} x_{k_j+1} = \lim_{j\to\infty} P_{\mathfrak{C}} \left(c_{k_j} - \lambda_{k_j} g(v_{k_j}) \right) = \bar{q}. \qquad (3.17)$$

Without loss of generality, suppose $\lambda_{k_j} \to \lambda^*$ and $g(v_{k_j}) \to g(z^*)$. Then since g is continuous, we have $g(z^*) = g(\bar{q})$. Therefore, from (3.17)

$$P_{\mathfrak{C}} (\bar{q} - \lambda^* g(z^*)) = \bar{q}.$$

It then follows from Lemma 2.4 that $\bar{q} \in \mathfrak{C}^*$, which is a contraction. Hence, our claim holds. Replacing q with $\bar{q}$ in (3.6), it is easy to see that $\lim_{k\to\infty} \|x_k - \bar{q}\|$ exists by Lemma 2.3. Since $\bar{q}$ is an accumulation point of $\{x_k\}$, we obtain that $\{x_k\}$ converges to $\bar{q}$. □

4. Numerical experiment

In this section, we evaluate the efficiency of the proposed algorithm, which we refer to as ITTMDY on some test problems, taking into account the number of iterations (NOI), the number of function evaluations (NFE), and the time it takes to achieve convergence (CPU TIME). We also compare ITTMDY with its non-inertial counterpart, the TTMDY algorithm, which was proposed in [12].

Matlab R2019b was used to code all of the algorithms. For the experiments, we consider the following factors:

- Number of problems: Ten test problems.
- Five dimensions (DIM): $1000, \ 5000, \ 10000, \ 50000, \ 100000$.
- Seven Initial points (INP):

Table 1. Initial points for the implementation of the algorithms

Starting Points		
ITTMDY		TTMDY
$x^1_{-1} = (0.1, 0.1, \cdots, 0.1)^T$	$x^1_0 = x^1_{-1}$	x^1_0
$x^2_{-1} = (0.2, 0.2, \cdots, 0.2)^T$	$x^2_0 = x^2_{-1}$	x^2_0
$x^3_{-1} = (0.5, 0.5, \cdots, 0.5)^T$	$x^3_0 = x^3_{-1}$	x^3_0
$x^4_{-1} = (1.2, 1.2, \cdots, 1.2)^T$	$x^4_0 = x^4_{-1}$	x^4_0
$x^5_{-1} = (1.5, 1.5, \cdots, 1.5)^T$	$x^5_0 = x^5_{-1}$	x^5_0
$x^6_{-1} = (2, 2, \cdots, 2)^T$	$x^6_0 = x^6_{-1}$	x^6_0
$x^7_{-1} = \mathrm{rand}(n, 1)$	$x^7_0 = x^7_{-1}$	x^7_0

- Algorithm implementation parameters: we select $\zeta = 1, \ \rho = 0.7, \ r = 1.8, \ \sigma = 10^{-3}, \ \psi = 0.9$. As for TTMDY, the parameters are selected as in [12].
- Stopping condition: Iterations are stopped when $\|g(c_k)\| \leq 10^{-6}$ and/or the number of iterations exceed 1000 without reaching a solution.

The test problems considered can be found in [17]

Problem 1: Modified exponential function:

$$\begin{aligned} g_1(x) &= e^{x_1} - 1 \\ g_i(x) &= e^{x_i} + x_i - 1, \ i = 1, 2, \ldots, n-1, \\ \mathfrak{C} &= \Re^n_+. \end{aligned}$$

Problem 2: Logarithmic function:

$$g_i(x_i) = \log(x_i + 1) - \frac{x_i}{n}, \ i = 1, 2, \dots, n,$$
$$\mathfrak{C} = \mathfrak{R}^n_+.$$

Problem 3: Nonsmooth function:

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

Problem 4:

$$g_i(x) = \min\left(\min(|x_i|, x_i^2), \max(|x_i|, x_i^3)\right) \ \ i = 1, 2, \dots, n,$$
$$\mathfrak{C} = \mathfrak{R}^n_+.$$

Problem 5: Strictly convex function I:

$$g_i(x) = e^{x_i} - 1, \ i = 1, 2, \dots, n,$$
$$\mathfrak{C} = \mathfrak{R}^n_+.$$

Problem 6: Strictly convex function II;

$$g_i(x) = \left(\frac{i}{n}\right) e^{x_i} - 1, \ i = 1, 2, \dots, n,$$
$$\mathfrak{C} = \mathfrak{R}^n_+.$$

Problem 7: Tridiagonal exponential function:

$$g_1(x) = x_1 - e^{\cos(l(x_1 + x_2))}$$
$$g_i(x) = x_i - e^{\cos(l(x_{i-1} + x_i + x_{i+1}))}, \ i = 2, \dots, n-1,$$
$$g_n(x) = x_n - e^{\cos(l(x_{n-1} + x_n))},$$
$$l = \frac{1}{n+1} \text{ and } \mathfrak{C} = \mathfrak{R}^n_+.$$

Problem 8: Nonsmooth function II:

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

Problem 9: Trig-Exp Function:

$$
\begin{aligned}
g_1(x) &= 3x_1^3 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2), \\
g_i(x) &= 3x_i^3 + 2x_{i+1} - 5 + \sin(x_i - x_{i+1})\sin(x_i + x_{i+1}) \\
&\qquad + 4x_i - x_{i-1}e^{(x_{i-1}-x_i)} - 3, \\
&\quad \text{for } 1 < i < n, \\
g_n(x) &= 4x_n - x_{n-1}e^{(x_{n-1}-x_n)} - 3, \\
\mathfrak{C} &= \mathfrak{R}^n_+.
\end{aligned}
$$

Problem 10: Penalty function I:

$$
\begin{aligned}
\xi_i &= \sum_{i=1}^{n} x_i^2, \ c = 10^{-5}, \\
g_i(x) &= 2c(x_i - 1) + 4(\xi_i - 0.25)x_i, \ i = 1, 2, \ldots, n, \\
\mathfrak{C} &= \mathfrak{R}^n_+.
\end{aligned}
$$

To depict the efficiency of the ITTMDY algorithm, we plot three graphs each corresponding to NOI, NFE and CPU TIME, respectively, with the aid of the Dolan and Morè [14] performance profiles. From the profile curve, the algorithm that top the curve has the best performance. From Figures 1, 2 and 3, we can notice that the inertial algorithm has lesser NOI and NFEin 85%, less CPU time in 75% of the problems, respectively. It can also be observed that the best performing algorithm is ITTMDY as it top all the curves. Hence, we can say that the ITTMDY algorithm is more efficient than TTMDY based on NOI, NFE and CPU TIME.

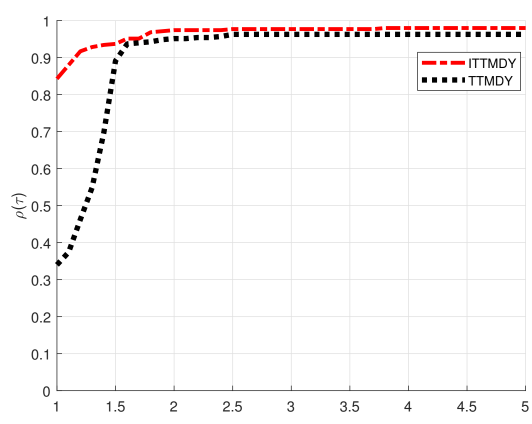

Figure 1. Performance profiles for the number of iterations

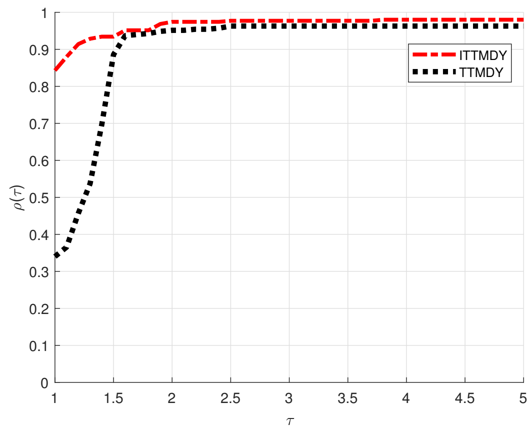

Figure 2. Performance profiles for the number of function evaluations

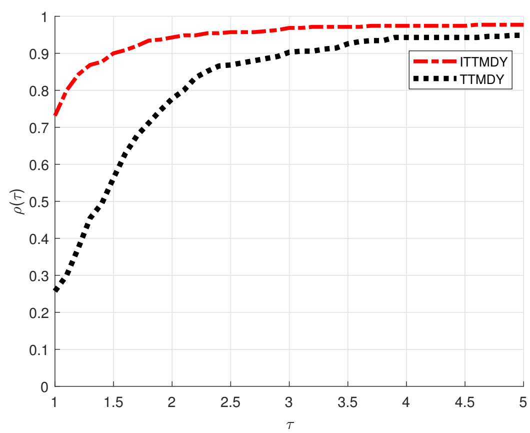

Figure 3. Performance profiles for the CPU time

5. Conclusions

In this article, a new derivative-free iterative algorithm for solving nonlinear monotone operator equations with convex constraints is proposed. The proposed method combines the inertial extrapolation term with a derivative-free method. Independent of the line search, the search direction of the new method is descent. Under some standard assumptions, the sequence generated by the new method converges globally. Finally, the efficiency of the new method was given through numerical experiments on some benchmark test problems. The results revealed that the inertial algorithm is more efficient than the existing non inertial.

Acknowledgments: The first and fourth authors would like to thank Phetchabun Rajabhat University. The third author acknowledge with thanks, the Department of Mathematics and Applied Mathematics at the Sefako Makgatho Health Sciences University.

References

[1] A.B. Abubakar, P. Kumam, M. Malik, P. Chaipunya, and A.H. Ibrahim, *A hybrid fr-dy conjugate gradient algorithm for unconstrained optimization with application in portfolio selection,* AIMS Mathematics, **6**(6) (2021), 6506–6527.

[2] A.B. Abubakar, P. Kumam, M. Malik, and A.H. Ibrahim, *A hybrid conjugate gradient based approach for solving unconstrained optimization and motion control problems,* Math. Comput. Simul., (2021), doi.org/10.1016/j.matcom.2021.05.038.

[3] A.B. Abubakar, P. Kumam, A.H. Ibrahim, and J. Rilwan, *Derivative-free hs-dy-type method for solving nonlinear equations and image restoration,* Heliyon, **6**(11):e05400, (2020), doi.org/10.1016/j.heliyon.2020.e05400.

[4] A.B. Abubakar, K. Muangchoo, A.H. Ibrahim, A.B. Muhammad, L.O. Jolaoso, and K.O. Aremu, *A new three-term hestenes-stiefel type method for nonlinear monotone operator equations and image restoration,* IEEE Access, **9** (2021), 18262–18277.

[5] A.B. Abubakar, K. Muangchoo, A.H. Ibrahim, J. Abubakar, and S.A. Rano, *Fr-type algorithm for finding approximate solutions to nonlinear monotone operator equations,* Arabian J. Math., **10** (2021), 261-270.

[6] A.B. Abubakar, P. Kumam, H. Mohammad, and A.H. Ibrahim, *Prp-like algorithm for monotone operator equations,* Japan J. Indus. Appl. Math., **38** (2021), 805-822.

[7] A.B. Abubakar, K. Muangchoo, A.H. Ibrahim, S.E. Fadugba, K.O. Aremu, and L.O. Jolaoso, *A modified scaled spectral- conjugate gradient-based algorithm for solving monotone operator equations,* J. Math., **2021** (2021), doi.org/10.1155/2021/5549878.

[8] A.B. Abubakar, P. Kumam, and A.H. Ibrahim, *Inertial derivative-free projection method for nonlinear monotone operator equations with convex constraints,* IEEE Access, **9** (2021), 92157-92167.

[9] A.B. Abubakar, P. Kumam, A.H. Ibrahim, P. Chaipunya, and S.A. Rano, *New hybrid three-term spectral-conjugate gradient method for finding solutions of nonlinear monotone operator equations with applications,* Math. Comput. Simul., (2021). doi.org/10.1016/j.matcom.2021.07.005.

[10] J. Abubakar, P. Kumam, A.H. Ibrahim, and A. Padcharoen, *Relaxed inertial tseng's type method for solving the inclusion problem with application to image restoration,* MDPI Mathematics, **8**(5) (2020),818. doi.org/10.1016/j.matcom.2021.07.005.

[11] J. Abubakar, P. Kumam, H.ur Rehman and A.H. Ibrahim, *Inertial iterative schemes with variable step sizes for variational inequality problem involving pseudomonotone operator,* MDPI Mathematics, **8**(4) (2020), 609. doi.org/10.3390/math8040609.

[12] H. Cao, *A three-term derivative-free projection method for convex constrained monotone equations,* ScienceAsia, **47**(2) (2021), 235-240.

[13] S.P. Dirkse and M.C. Ferris, *Mcplib: A collection of nonlinear mixed complementarity problems,* Optimi. Meth. Software, **5**(4) (1995), 319–345.

[14] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles.* Math. Programming, **91**(2) (2002), 201–213.

[15] P. Gao and C. He, *An efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints*, Calcolo, **55** (2018), Article number: 53. doi.org/10.1007/s10092-018-0291-2

[16] A.H. Ibrahim, A.I. Garba, H. Usman, J. Abubakar and A.B. Abubakar, *Derivative-free rmil conjugate gradient method for convex constrained equations,* Thai J. Math., **18**(1) (2020), 211–231.

[17] A.H. Ibrahim, P. Kumam, A.B. Abubakar, W. Jirakitpuwapat and J. Abubakar, *A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing,* Heliyon, **6**(3) (2020), e03466. doi: 10.1016/j.heliyon.2020.e03466.

[18] A.H. Ibrahim, P. Kumam, A.B. Abubakar, J. Abubakar and A.B. Muhammad, *Least-square-based three-term conjugate gradient projection method for ℓ_1-norm problems with application to compressed sensing,* MDPI Mathematics, **8**(4) (2020):602. doi.org/10.3390/math8040602.

[19] A.H. Ibrahim, P. Kumam, A.B. Abubakar, U.B. Yusuf and J. Rilwan. *Derivative-free conjugate residual algorithms for convex constraints nonlinear monotone equations and signal recovery,* J. Nonlinear Convex Anal., **21** (2020), 1959–1972.

[20] A.H. Ibrahim, P. Kumam and W. Kumam, *A family of derivative-free conjugate gradient methods for constrained nonlinear equations and image restoration,* IEEE Access, **8** (2020), 162714–162729.

[21] A.H. Ibrahim, P. Kumam, A.B. Abubakar, U.B. Yusuf, S.E. Yimer and K.O. Aremu,*An efficient gradient-free projection algorithm for constrained nonlinear equations and image restoration,* AIMS Mathematics, **6**(1) (2020), 235-260. doi: 10.3934/math.2021016.

[22] A.H. Ibrahim, K. Muangchoob, N.S. Mohamed and A.B. Abubakar, *Derivative-free smr conjugate gradient method for constraint nonlinear equations,* J. Math. Comput. Sci., **24**(2) (2022),147–164.

[23] A.H. Ibrahim, K. Muangchoo, A.B. Abubakar, A.D. Adedokun and H. Mohammed, *Spectral conjugate gradient like method for signal reconstruction,* Thai J. Math., **18**(4) (2020), 2013–2022.

[24] A.H. Ibrahim and P. Kumam, *Re-modified derivative-free iterative method for nonlinear monotone equations with convex constraints,* Ain Shams Eng. J., (2021). doi.org/10.1016/j.asej.2020.11.009.

[25] A.H. Ibrahim, J. Deepho, A.B. Abubakar and K.O. Aremu, *A modified liu-storey-conjugate descent hybrid projection method for convex constrained nonlinear equations and image restoration,* Num. Algebra Control Opti., (2021). doi: 10.3934/naco.2021022

[26] A.H. Ibrahim, P. Kumam, B.A. Hassan, A.B. Abubakar and J. Abubakar, *A derivative-free three-term hestenes-stiefel type method for constrained nonlinear equations and image restoration,* Inter. J.Comput. Math., (2021), 1–22. doi.org/10.1080/00207160.2021.1946043

[27] A.H. Ibrahim, J. Deepho, Auwal Bala Abubakar and A. Adamu, *A three-term polak-ribire-polyak derivative-free method and its application to image restoration,* Scientific African, **13** (2021), e00880. doi.org/10.1016/j.sciaf.2021.e00880.

[28] A.H. Ibrahim, P. Kumam, M. Sun, P. Chaipunya and A.B. Abubakar, $\hat{\text{P}}$rojection method with inertial step for nonlinear equations: Application to signal recovery, J. Indus.Manag. Optimi., (2021). doi: 10.3934/jimo.2021173

[29] C. Izuchukwu and Y. Shehu, *New inertial projection methods for solving multivalued variational inequality problems beyond monotonicity,* Networks Spatial Econo., **22**(2) (2021), 291-323. doi: 10.1007/s11067-021-09517-w

[30] C. Izuchukwu, S. Reich, and Y. Shehu, *Relaxed inertial methods for solving the split monotone variational inclusion problem beyond co-coerciveness,* Optimization., (2021), 1–40. doi.org/10.1080/02331934.2021.1981895

[31] K. Meintjes and A.P. Morgan, *A methodology for solving chemical equilibrium systems,* Appl. Math. Comput., **22**(4) (1987), 333–361.

[32] B.T. Polyak, *Some methods of speeding up the convergence of iteration methods,* Ussr Comput. Math. Math. Physics, **4**(5) (1964), 1–17.
[33] M.V. Solodov and B.F. Svaiter, *A new projection method for variational inequality problems,* SIAM J. Cont. Optimi., **37**(3) (1999), 765–776.
[34] A.J. Wood and B.F. Wollenberg, *Power generation, operation and control,* New York: John Wiley & Sons, 1996.
[35] Y. Xiao and H. Zhu, *A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing,* J. Math. Anal. Appl., **405**(1) (2013), 310–319.