EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2026, Vol. 19, Issue 2, Article Number 6828
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

On Accelerated Dai-Liao-Type Derivative-Free Projection Method for Solving Nonlinear Equations

Ibrahim Jabo Abubakar[1], Kazeem Olalekan Aremu[1,2,*], Ibrahim Muhammad[1], Sefina Bola Jaiyeola[3]

[1] *Department of Mathematics, Faculty of Physical and Computing Sciences, Usmanu Danfodiyo University Sokoto, PMB 2346 Sokoto, Sokoto State, Nigeria*
[2] *Department of Mathematics and Applied Mathematics, School of Science and Technology, Sefako Makgatho Health Sciences University, P.O. Box 94 Medunsa 0204, Ga-Rankuwa, Pretoria, South Africa*
[3] *Department of Physical Sciences, Faculty of Natural and Applied Sciences, Al-Hikmah University, Ilorin, Kwara State, Nigeria*

Abstract. Ou and Li [Journal of Applied Mathematics and Computing 56.1-2 (2018): 195-216] introduced an adaptive line search that reduces the computational cost of the line search method for solving nonlinear systems. Based on this line search, we propose an accelerated Dai-Liao-type derivative-free projection method for solving large-scale systems of nonlinear equations with convex constraints. Under some mild assumptions, we establish the global convergence of the proposed method. Numerical experiments show that the method is fast, and competitive with other related methods.

2020 Mathematics Subject Classifications: 65K15, 47J25, 65J15, 90C33

Key Words and Phrases: Derivative-free method, line-search, nonlinear equations

---

1. Introduction

In this work, we address the problem of finding the solution to a nonlinear equation

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

where F is a mapping from an n-dimensional Euclidean space $\mathbb{R}^n$ into itself, $x \in C$ and C is a closed convex subset of $\mathbb{R}^n$. The mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is monotone, if

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

---

*Corresponding Author.
DOI: https://doi.org/10.29020/nybg.ejpam.v19i2.6828

Email addresses: ibjabo@gmail.com (I. J. Abubakar), aremukazeemolalekan@gmail.com (K. O. Aremu), muhammad.ibrahim@udusok.edu.ng (I. Muhammad), sbjaiyeola@alhikmah.edu.ng (S. B. Jaiyeola

The problem (1) encompasses a wide range of models in various fields including chemical equilibrium systems, optimal flow equations [1] and financial forecasting problems [2, 3] and many others. Over the years, researchers have developed numerous methods to solve (1). Among the classical iterative methods for solving (1) are Newton's, inexact Newton's and the quasi-Newton's methods [4–6]. They have been widely used due to their rapid convergence from a suitable initial guess . However, these methods often require the computation of Jacobian matrix and its approximation at each iteration, making them unsuitable for handling large scale optimization problems.

To overcome the drawback, matrix-free approaches [7, 8], specifically conjugate gradient methods (CGM) have been developed to address the aforementioned problems (see [9–15] and other references therein). These methods eliminate the need for explicit computation of Jacobian matrix making them more efficient for large-scale optimizations. Due to this advantage, the CGM and its numerous variants have been adopted as efficient and standard methods of solving nonlinear equations. (see [16–19]). On the other hand, the derivative-free methods for solving systems of nonlinear equations aim to find solutions to nonlinear equations without relying on derivatives. As a result, they have gained significant attention by researchers. The projection method have received considerable attention with Solodov and Svaiter [20] as pioneers. In 2006, Zheng and Zhou [21] proposed a spectral gradient projection method for solving nonlinear monotone equations. Wang *et al.,* [22] in 2007, combined the projection technique of Solodov and Svaiter [20] with the CGM to solve large-scale nonlinear equations. By popularizing the idea of Zheng and Zhou [21], Yu *et al.,* [23] in 2009, proposed a constrained version of the spectral gradient algorithm for solving monotone nonlinear equations. In this algorithm, computing the sequence steps neither needs matrix storage nor solutions of linear systems of equations. Other variants of derivative-free projection methods were developed by researchers. These include an efficient three-term derivative-free method for solving monotone nonlinear equations with convex constraints by choosing a part of the Liu-Storey (LS) conjugate parameter as a new conjugate parameter by Gao and He [24]. Liu and Feng [25] proposed a derivative free iterative method for nonlinear monotone equations with convex contraints, in 2020 Dai and Zhu [26] introduced a modified Hestenes-Stiefel-type derivative free method for large scale non-linear monotone equations. Ibrahim *et al.,* [27] introduced a Dai-Liao type projection method for finding the solution to monotone nonlinear equations. The method can be seen as an extension of the modified mixed hybrid Dai-Liao method (MMDL) introduced by Stanimirovic *et al.,* [28].

The computational cost of the projection methods depends on determining the line search direction and step-size. Researchers have proposed two main line search scheme which are used in projection-type methods. The two used line searches in projection type methods are as follows;

Line Search A (**LSA**) Choose step-size $\alpha_k = \max\{\kappa\rho^i : i = 0, 1, 2, 3...\}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \gamma \alpha_k \|d_k\|^2, \tag{2}$$

Line Search B (**LSB**)

$$-F(x_k + \alpha_k d_k)^T d_k \geq \gamma \alpha_k \|F(x_k + \alpha_k d_k)\| \|d_k\|^2. \tag{3}$$

Results from numerical experiments have shown that these line searches exhibit different performance when solving a problem. It was noted that in **LSA**, the right hand side will be too large when x_k is near the solution of the problem 1 and also $\|d_k\|^2$ be too large. Similarly, if the second line search (**LSB**) is used , the right hand side will also be too large and $\|F(x_k + \alpha_k d_k)\|$ will also be too large if x_k is far from the solution to the problem 1. To overcome the drawbacks in **LSA** and **LSB**, Oui and Li [29] proposed a new line search (**NLS**) as follows:
Given a current approximation x_k and search direction d_k, compute a step-size $\alpha_k = \max\{\kappa\rho^i : i = 0, 1, 2, 3...\}$ such that;

$$-F(x_k + \alpha_k d_k)^T d_k \geq \gamma \theta_k \alpha_k \|d_k\|^2, \tag{4}$$

where $\kappa > 0$ is an initial step size, $\rho \in (0, 1)$ $\gamma > 0$ are constants and

$$\theta_k = \lambda_k + (1 - \lambda_k)\|F(x_k + \alpha_k d_k)\|. \tag{5}$$

The **NLS** combines the advantages of the two line searches **LSA** and **LSB** by using convex combination. Furthermore, **NLS** behaves like **LSA** as $\lambda_k \to 1$ and **LSB** as $\lambda_k \to 0$. Thus,the **NLS** has better optimal performance when compared to **LSA** and **LSB**.

One of the ways to test efficiency of an iterative method is the rate of convergence. In doing this, most researchers employ the inertial extrapolation technique. The inertial technique introduced by Polyak[30] is a two-step iterative method used to accelerate the convergence of iterative methods. It uses the previous two iterates to compute the next iterate, effectively improving the convergence rate. There is a substantial body of literature dedicated to this subject, with various fast iterative algorithms constructed using inertial extrapolation. Examples of such algorithms include inertial forward-backward splitting method [31, 32], inertial Mann method [33], inertial sub-gradient extra-gradient [34] and inertial contraction method [35]. These inertial-type methods generally require fewer iterations than their non-inertial counterparts. Thus, the effectiveness in improving convergence make the inertial technique valuable tool in solving optimization problems efficiently.(Ibrahim *et al.,* [36]). These inspired researchers to incorporate the inertial term into the derivative-free projection methods to achieve faster convergence.

It is observed that the algorithm proposed by Ibrahim *et al.,* [27] offers an efficient approach to finding the solution to the nonlinear equation (1). However, We observe that the line search adopted in the algorithm proposed by Ibrahim *et al.,* [27] is one of the two most commonly used line search for projection-type methods. It was noted that in **LSA** used by Ibrahim *et al.,* [27], the right hand side will be too large when x_k is near the solution of the problem 1 and also $\|d_k\|^2$ be too large. Similarly, if the second line search (**LSB**) is used , the right hand side will also be too large and $\|F(x_k + \alpha_k d_k)\|$ will also be too large if x_k is far from the solution to the problem 1. Inspired by recent researches in this direction, we propose a new iterative method by incorporating an inertial term into the Dai-liao type projection method with the New Line Search (NLS) to solve monotone nonlinear equation. This combination will help enhance the convergence speed and efficiency of our proposed algorithm. In this work, we present a theoretical analysis of the proposed method and conduct numerical experiment to validate its efficiency. The result will help to demonstrate the superiority of our method in terms of speed and accuracy. Also, by introducing this method with inertial term and New Line Search, this work contributes to the advancement of solving monotone nonlinear equations and offers a promising approach for handling large scale problems in various scientific and practical applications.

The rest of the paper is organised as follows. In section 2 we present the Accelerated Dai-Liao type projection method for solving monotone nonlinear equations. Section 3 is the convergence result of the proposed method. Section 4 presents the numerical experiments for solving nonlinear equations. Finally, section 5 is the conclusion.

2. Preliminaries and Algorithm

In this section, we present some definitions and lemmas that will used in establishing the convergence of the proposed algorithm. .

Definition 1. *[37] The projection operator $P_{[a,b]} : \mathbb{R} \to [a, b]$, which projects onto the closed and bounded interval $[a, b]$, as*

$$P_{[a,b]}(x) = \max\{a, \min\{b, x\}\}, \quad \forall x \in \mathbb{R}. \tag{6}$$

Lemma 1. *[10] Let $\mathbb{R}^n$ be an Euclidean space, for every $x, y \in \mathbb{R}^n$,*

$$\|x + y\|^2 \leq \|x\|^2 + 2y^T(x + y). \tag{7}$$

Lemma 2. *[10] Let $\mathcal{C} \subset \mathbb{R}^n$ be a nonempty closed and convex set. The following statements holds:*

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

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

(iii) $\|P_{\mathcal{C}}(x) - y\|^2 \leq \|x - y\|^2 - \|x - P_{\mathcal{C}}(x)\|^2$.

Lemma 3. *[38] Let $\{x_k\}$ and $\{w_k\}$ be sequences of non-negative real numbers such that $\sum_{k=1}^{\infty} w_k < \infty$ and $x_{k+1} \leq x_k + w_k$. Then $\{x_k\}$ is a convergent sequence.*

2.1. Proposed Search Direction

Based on the Dai-Liao derivative-free method proposed by Ibrahim *et.al* [27], in this paper, we proposed an inertial derivative-free projection method that generates a trial point w_k using the recursive formula,

$$w_k = t_k + s_k, \quad s_k = \alpha_K d_k. \quad k = 0, 1, 2, ...$$

where t_k is the inertial extrapolation step, defined as

$$t_k = x_k + \beta_k(x_k - x_{k-1}),$$

and

$$\beta_k = \begin{cases} \min\left\{\beta, \frac{\epsilon_k}{\|x_k - x_{k-1}\|}\right\}, & \text{if} \quad x_k \neq x_{k-1}, \\ \beta, & \text{otherwise}. \end{cases}$$

The parameter $\alpha_k > 0$ is a step length obtained using a line search procedure and d_k is a search direction defined as follows:

$$d_0 = -F(t_0), \quad d_k = -F(t_k) + \delta_k \left(I - \frac{F(t_k)F(t_k)^T}{\|F(t_k)\|^2}\right) s_{k-1} \forall k > 0, \tag{8}$$

Other parameters are defined as follows:

$$\begin{aligned} y_k &= -F(t_k) - F(t_{k-1}) + rs_{k-1}, r > 0, \\ v_{k-1} &= 1 + max\left\{0, -\frac{d_{k-1}^T y_{k-1}}{d_{k-1}^T d_{k-1}}\right\}, \quad z_{k-1} = y_{k-1} + v_{k-1}d_{k-1}. \end{aligned} \tag{9}$$

$$\begin{aligned} \delta_k^{(1)} &= \frac{\|F(t_k)\|^2 - \frac{\|F(t_k)\|}{\|F(t_{k-1})\|}|F(t_k)^T F(t_{k-1})|}{\mu|F(t_k)^T d_{k-1}| + d_{k-1}^T z_{k-1}} - \alpha_k \frac{F(t_k)^T s_{k-1}}{d_{k-1}^T z_{k-1}}. \\ \delta_k^{(2)} &= \frac{\|F(t_k)\|^2 - \frac{\|F(t_k)\|}{\|F(t_{k-1})\|}|F(t_k)^T F(t_{k-1})|}{\mu|F(t_k)^T d_{k-1}| - d_{k-1}^T F(t_{k-1})} - \alpha_k \frac{F(t_k)^T s_{k-1}}{d_{k-1}^T z_{k-1}}. \\ \delta_k &= \max\{0, \min\{\delta_k^1, \delta_k^2\}\}. \end{aligned} \tag{10}$$

Next, we present our algorithm as follows:

Algorithm 1. *Accelerated Dai-Liao type derivative-free projection algorithm*
Initialization. *Choose $x_0 \in \mathcal{C}$, $\epsilon_k \in (0,1)$ such that $\sum_{k=0}^{\infty} \epsilon_k < \infty$, $\rho \in (0,1)$, $\beta \in [0,1)$, $\epsilon \in (0,1)$, $\kappa \in (0,1]$, $\gamma \in (0,1)$, $\varrho \in (0,2)$, $0 < \eta_1 \leq \eta_2$. Set $x_1 = x_0$ and set $k = 1$.*

Step 1. *If $\|F(t_k)\| \leq \epsilon$, stop. Otherwise, choose an inertial extrapolation steplength*

$$\beta_k = \begin{cases} \min\left\{\beta, \frac{\epsilon_k}{\|x_k - x_{k-1}\|}\right\}, & \textit{if} \quad x_k \neq x_{k-1}, \\ \beta, & \textit{otherwise}, \end{cases} \tag{11}$$

and compute the inertial step

$$t_k = x_k + \beta_k(x_k - x_{k-1}). \tag{12}$$

Step 2. *If* $\|F(t_k)\| \leq \epsilon$*, stop. Otherwise compute the search direction* d_k *by* (8)
Step 3. *Set* $w_k = t_k + \alpha_k d_k$*, where the step-size* $\alpha_k = \kappa\rho^i$ *with* i *being the least non-negative integer such that*

$$-F(t_k + \alpha_k d_k)^T d_k \geq \gamma\theta_k\alpha_k\|d_k\|^2, \tag{13}$$

where

$$\theta_k = \lambda_k + (1 - \lambda_k)P_{[\eta_1,\eta_2]}\|F(t_k + \alpha_k d_k)\|, \tag{14}$$

$P_{[\eta_1,\eta_2]}$ *is defined by 1 and* $0 < \eta_1 \leq \eta_2$.

Step 4. *If* $w_k \in \mathcal{C}$ *and* $\|F(w_k)\| \leq \epsilon$*, stop. Otherwise, determine the next iterates by*

$$x_{k+1} = P_{\mathcal{C}}[t_k - \varrho\psi_k F(w_k)], \tag{15}$$

where

$$\psi_k = \frac{F(w_k)^T(t_k - w_k)}{\|F(w_k)\|^2}.$$

Step 5. *Set* $k = k + 1$ *and go to step 1.*

Remark 1. *From step 1 of Algorithm 1, it can be observed that* $\beta_k\|x_k - x_{k-1}\| \leq \epsilon_k$*, and we have*

$$\sum_{k=1}^{\infty}\beta_k\|x_k - x_{k-1}\| \leq \sum_{k=1}^{\infty}\epsilon_k < \infty, \tag{16}$$

which further implies that

$$\lim_{k\to\infty}\beta_k\|x_k - x_{k-1}\| = 0. \tag{17}$$

3. Convergence Analysis

In this section, we establish the convergence analysis of the proposed Algorithm 1. To do so, we need the following assumptions and lemmas as follows:

(A1) The solution set denoted by Γ of the considered problem is nonempty.

(A2) The mapping F is monotone on $\mathcal{C}$. That is, for all $x, y \in \mathcal{C}$

$$(F(x) - F(y))^T(x - y) \geq 0.$$

(A3) The mapping F is Lipschitz continuous on $\mathcal{C}$. That is, for all $x, y \in \mathcal{C}$ there exist a constant $L > 0$ such that

$$\|F(x) - F(y)\| \leq L\|x - y\|.$$

Lemma 4. *The search direction d_k generated by Algorithm 1 satisfies the sufficient decent condition. That is for all k,*

$$F(t_k)^T d_k = -\|F(t_k)\|^2. \tag{18}$$

Proof. For $k = 0$, multiplying both sides of (8) by $F_{t_0}^T$, we have

$$F(t_0)^T d_0 = -\|F_{t_0}\|^2.$$

For $k > 0$, multiplying both sides of (8) by $F(t_k)^T$, we obtain

$$\begin{aligned} F(t_k)^T d_k &= F(t_k)^T \left[-F(t_k) + \delta_k \left(I - \frac{F(t_k)F(t_k)^T}{\|F(t_k)\|^2} \right) s_{k-1} \right] \\ &= F(t_k)^T \left[-F(t_k) + \delta_k s_{k-1} - \delta_k \frac{\|F(t_k)\|^2}{\|F(t_k)\|^2} s_{k-1} \right] \\ &= -\|F(t_k)\|^2 + \delta_k F(t_k)^T s_{k-1} - \delta_k F(t_k)^T s_{k-1} \\ &= -\|F(t_k)\|^2. \end{aligned}$$

Lemma 5. *The line search condition* (13) *is well defined. That is for all $k \geq 0$, there exists a non-negative integer i satisfying* (13)*.*

Proof. We prove this claim by contradiction and assume there exists $k_0 \geq 0$, such that for any non-negative integer i, (13) does not hold. That is for all $i \geq 0$, it holds that

$$-F(t_{k_0} + \kappa\rho^i d_{k_0})^T d_{k_0} < \gamma\kappa\rho^i\theta_{k_0}\|d_k\|^2, \tag{19}$$

where θ_k is defined by (14) and obviously

$$\lambda_{\min} \leq \theta_{k_0} \leq \max\{1, P_{[\eta_1,\eta_2]}[\|F(t_0 + \kappa\rho^i d_{k_0}\|]\}.$$

By continuity of the mapping F and passing $i \to \infty$, we obtain

$$-F(t_{k_0})^T d_{k_0} \leq 0, \tag{20}$$

which contradicts (18) and that completes the proof.

Lemma 6. *Suppose that Assumption (A1)-(A3) hold, and the sequences $\{t_k\}$ and $\{w_k\}$ are generated by Algorithm (1). Then there exist*

$$\alpha_k \geq \frac{\rho}{L + \gamma\hat{\theta}_k} \frac{\|F(t_k)\|^2}{\|d_k\|^2}, \quad \forall k. \tag{21}$$

Proof. By Definition 1, it is clear that

$$P_{[\eta_1,\eta_2]}[\|F(t_k + \bar{\alpha}_k d_k)\|] \leq \eta_2, \quad \forall k. \tag{22}$$

If $\alpha_k \neq \kappa$, the line search procedure implies that $\bar{\alpha}_k = \frac{\alpha_k}{\rho}$ does not adhere to condition (13), namely

$$-F(t_k + \rho^{-1}\alpha_k d_k)^T d_k < \gamma\rho^{-1}\alpha_k \bar{\theta}_k \|d_k\|^2, \tag{23}$$

where

$$\bar{\theta}_k = \lambda_k + (1-\lambda_k)\|F(t_k + \bar{\alpha_k} d_k)\| \leq \max\{1, P_{[\eta_1,\eta_2]}[\|F(t_k + \bar{\alpha}_k d_k)\|]\}.$$

Following (22), $\bar{\theta}_k \leq \hat{\theta}$, where $\hat{\theta} = \max\{1, \eta_2\}$. By Lipschitz continuity and (23), we have

$$\begin{aligned} \|F(t_k)\|^2 \leq & (F(t_k + \rho^{-1}\alpha_k d_k) - F(t_k))^T d_k - F(t_k + \rho^{-1}\alpha_k d_k)^T d_k \\ \leq & \|F(t_k + \rho^{-1}\alpha_k d_k) - F(t_k)\|\|d_k\| - F(t_k + \rho^{-1}\alpha_k d_k)^T d_k \\ \leq & (L + \gamma\hat{\theta}_k)\rho^{-1}\alpha_k \|d_k\|^2. \end{aligned} \tag{24}$$

This inequality leads to the desired inequality.

Lemma 7. *Let the sequences $\{x_k\}$ and $\{w_k\}$ be generated by Algorithm (1). For any $x^* \in \Gamma$, it holds that,*

$$\|x_{k+1} - x^*\|^2 \leq \|t_k - x^*\|^2 - \varrho(2-\varrho)\left(\frac{F(w_k)^T(t_k - w_k)}{\|F(w_k)\|}\right)^2, \quad \forall k.$$

Moreover, the sequence $\{x_k\}$ is bounded.

Proof. Using the nonexpansive property of projection operator (Lemma 2(iii)),

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|P_{\mathcal{C}}(t_k - \varrho\psi_k F(w_k)) - P_{\mathcal{C}}(x^*)\|^2 \\ &\leq \|t_k - \varrho\psi_k F(w_k) - x^*\|^2 \\ &= \|t_k - x^*\|^2 - 2\varrho\psi_k F(w_k)^T(t_k - x^*) + \varrho^2\psi_k^2\|F(w_k)\|^2. \end{aligned} \tag{25}$$

Observe that by Assumption (A2), we have

$$\begin{aligned} F(w_k)^T(t_k - x^*) &= F(w_k)^T(t_k - w_k) + F(w_k)^T(w_k - x^*) \\ &\geq F(w_k)^T(t_k - w_k) + F(x^*)^T(w_k - x^*) \\ &= F(w_k)^T(t_k - w_k). \end{aligned} \tag{26}$$

Combining (26) with (25), we have

$$\leq \|t_k - x^*\|^2 - 2\varrho\psi_k F(w_k)^T(t_k - w_k) + \varrho^2\psi_k^2\|F(w_k)\|^2$$

$$\leq \|t_k - x^*\|^2 - \varrho(2-\varrho)\left(\frac{F(w_k)^T(t_k - w_k)}{\|F(w_k)\|}\right)^2.$$

Furthermore, from the above relation, we have

$$\begin{aligned} \|x_{k+1} - x^*\| &\leq \|t_k - x^*\| = \|x_k + \beta_k(x_k - x_{k-1}) - x^*\| \\ &\leq \|x_k - x^*\| + \beta_k\|x_k - x_{k-1}\|. \end{aligned} \tag{27}$$

Noting (16), it follows from by Lemma (3), that the sequence $\{\|x_k - x^*\|\}$ is convergent. Therefore, $\{\|x_k - x^*\|\}$ is bounded by a positive number, say $\bar{m}_0$.

Lemma 8. *Suppose that Assumption (A1)-(A3) holds, and the sequence* $\{t_k\}$ *and* $\{w_k\}$ *are generated by Algorithm 1. Then*

$$\lim_{k\to\infty} \|w_k - t_k\| = 0. \tag{28}$$

Proof. Recall from Lemma 7 that

$$\|x_{k+1} - x^*\|^2 \leq \|t_k - x^*\|^2 - \varrho(2-\varrho)\left(\frac{F(w_k)^T(t_k - w_k)}{\|F(w_k)\|}\right)^2. \tag{29}$$

Noting the definition of the inertial extrapolation term (12), Lemma (1), Lemma 7 and Remark 1, we have

$$\begin{aligned} \|t_k - x^*\|^2 &= \|x_k - x^* + \beta_k(x_k - x_{k-1})\|^2 \\ &\leq \|x_k - x^*\|^2 + 2\beta_k(x_k - x_{k-1})^T(x_k - x^* + \beta_k(x_k - x_{k-1})) \\ &\leq \|x_k - x^*\|^2 + 2\beta_k\|x_k - x_{k-1}\|\|(x_k - x^*) + \beta_k(x_k - x_{k-1})\| \\ &\leq \|x_k - x^*\|^2 + 2\beta_k\|x_k - x_{k-1}\|\left(\|x_k - x^*\| + \|\beta_k(x_k - x_{k-1})\|\right) \\ &\leq \|x_k - x^*\|^2 + 2\epsilon_k(\bar{m}_0 + \epsilon_k). \end{aligned} \tag{30}$$

Furthermore, from (13), it follows that

$$F(w_k)^T(t_k - w_k) = -\alpha_k F(w_k)^T d_k \geq \gamma\alpha_k^2\theta_k\|d_k\|^2, \tag{31}$$

where θ_k is defined by (14). By utilizing relation (31) and noting that $\theta_k \geq \lambda_{\min} > 0$ holds true for all k, we have

$$F(w_k)^T(t_k - w_k) \geq \gamma\alpha_k^2\lambda_{\min}\|d_k\|^2 = \gamma\lambda_{\min}\|w_k - t_k\|^2. \tag{32}$$

Integrating (30) and (32) with (29), it follows that

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 + 2\epsilon_k(\bar{m}_0 + \epsilon_k) - \varrho(2-\varrho)\gamma^2\lambda_{\min}^2\frac{\|w_k - t_k\|^4}{\|F(w_k)\|^2}. \tag{33}$$

Thus, we have

$$\varrho(2-\varrho)\gamma^2\lambda_{\min}^2\frac{\|w_k-t_k\|^4}{\|F(w_k)\|^2}\le\|x_k-x^*\|^2-\|x_{k+1}-x^*\|^2+2\epsilon_k(\bar{m}_0+\epsilon_k).$$

From Lemma 7 we can infer that $\{x_k\}$ is bounded, therefore, it is easy to see that $\{t_k\}$ is also bounded. Building upon the assumption of Lipschitz continuity for the underlying mapping F, we can conclude the existence of a positive constant, denoted $\bar{m}_2$ that serves as an upper bound for $\{F(t_k)\}$. That is

$$\|F(t_k)\|\le\bar{m}_2,\quad\forall k. \tag{34}$$

Using (34) and Assumption (A2) yields,

$$\begin{aligned}F(w_k)^T(t_k-w_k)&=(F(w_k)-F(t_k))^T(t_k-w_k)+F(t_k)^T(t_k-w_k)\\&\le\|F(t_k)\|\|t_k-w_k\|\\&\le\bar{m}_2\|t_k-w_k\|.\end{aligned}\tag{35}$$

Combined with (32), it is easy to deduce that $\|w_k-t_k\|\le\frac{\bar{m}_2}{\gamma\lambda_{\min}}$. Then we can obtain

$$\|w_k\|\le\frac{\bar{m}_2}{\gamma\lambda_{\min}}+\|t_k\|.$$

Therefore, the boundedness of $\{t_k\}$ implies that the sequence $\{w_k\}$ is also bounded.
Given the Lipschitz continuity of the mapping F and the boundedness of the sequence $\{w_k\}$, it follows that the sequence $\{F(w_k)\}$ is also bounded. Consequently, there exists a positive constant $\bar{m}_1>0$, such that $\|F(w_k)\|\le\bar{m}_1$. By integrating this information with relation (33) and summing over $k=1,2,3\cdots$, we have

$$\varrho(2-\varrho)\gamma^2\lambda_{\min}^2\|w_k-t_k\|^4\le\bar{m}_1^2\sum_{k=1}^{\infty}\big(\|x_k-x^*\|^2-\|x_{k+1}-x^*\|^2)+2\epsilon_k(\bar{m}_0+\epsilon_k)\big)<\infty.$$

Hence the desired result (28) follows.

Lemma 9. *Let $\epsilon>0$, such that for all k, $\|F(t_k)\|\ge\epsilon$. Then the search direction d_k generated by Algorithm (1) is bounded. That is,*

$$\|d_k\|\le\tilde{m},\quad\forall k \tag{36}$$

Proof. Recall from (18)

$$\|F(t_k)\|^2=-F(t_k)^Td_k\le\|F(t_k)\|\|d_k\|$$

which implies that

$$\|d_k\|\ge\|F(t_k)\|\ge\epsilon,\quad\forall k. \tag{37}$$

On the other hand, from the definition of y_{k-1} and z_{k-1} from (9), we have

$$d_{k-1}^T z_{k-1} \geq d_{k-1}^T y_{k-1} + \|d_{k-1}\|^2 - d_{k-1}^T y_{k-1} = \|d_{k-1}\|^2 > 0. \tag{38}$$

Now, to establish that the search direction d_k is bonded. We have the two cases (i) when $\delta_k = 0$ and when $\delta_k = \min\{\delta_k^{(I)}, \delta_k^{(II)}\}$. If $\delta_k = 0$, it is easy to see that (36) holds. On the other hand, if $\delta_k = \min\{\delta_k^{(I)}, \delta_k^{(II)}\}$, we have the following sub-cases:

Sub-case (i) If $\delta_k = \delta_k^{(I)}$, then

$$\begin{aligned}
\|d_k\| &\leq \|F(t_k)\| + 2|\delta_k^{(I)}|\|s_{k-1}\| \\
&\leq \|F(t_k)\| + 2\left|\frac{\|F(t_k)\|^2 - \frac{\|F(t_k)\|}{\|F(t_{k-1})\|}|F(t_k)^T F(t_{k-1})|}{\mu|F(t_k)^T d_{k-1}| + d_{k-1}^T z_{k-1}} - \alpha_k \frac{F(t_k)^T s_{k-1}}{d_{k-1}^T z_{k-1}}\right| \|s_{k-1}\| \\
&\leq \|F(t_k)\| + 2\left(2\frac{\|F(t_k)\|^2}{\|d_{k-1}\|^2} + \alpha_k \frac{\|F(t_k)\|}{\|d_{k-1}\|^2}\alpha_{k-1}\|d_{k-1}\|\right)\|s_{k-1}\| \\
&\leq \|F(t_k)\| + 2\left(2\frac{\|F(t_k)\|^2}{\|d_{k-1}\|^2} + \frac{\|F(t_k)\|}{\|d_{k-1}\|}\right)\|s_{k-1}\| \\
&\leq \|F(t_k)\| + 2\alpha_{k-1}\|d_{k-1}\|\left(2\frac{\|F(t_k)\|^2}{\|d_{k-1}\|^2} + \frac{\|F(t_k)\|}{\|d_{k-1}\|}\right) \\
&\leq \|F(t_k)\| + 2\left(2\frac{\|F(t_k)\|^2}{\|d_{k-1}\|^2}\alpha_{k-1}\|d_{k-1}\| + \frac{\|F(t_k)\|}{\|d_{k-1}\|}\alpha_{k-1}\|d_{k-1}\|\right) \\
&\leq \|F(t_k)\| + 2\left(2\frac{\|F(t_k)\|^2}{\|d_{k-1}\|} + \|F(t_k)\|\right) \\
&\leq \bar{m}_2 + 2\left(2\frac{\bar{m}_2^2}{\epsilon} + \bar{m}_2\right) \triangleq b_1.
\end{aligned} \tag{39}$$

Sub-case (ii) If $\delta_k = \delta_k^{(II)}$, then

$$\begin{aligned}
\|d_k\| &\leq \|F(t_k)\| + 2|\delta_k^{(II)}|\|s_{k-1}\| \\
&\leq \|F(t_k)\| + 2\left|\frac{\|F(t_k)\|^2 - \frac{\|F(t_k)\|}{\|F(t_{k-1})\|}|F(t_k)^T F(t_{k-1})|}{\mu|F(t_k)^T d_{k-1}| - d_{k-1}^T F(t_{k-1})} - \alpha_k \frac{F(t_k)^T s_{k-1}}{d_{k-1}^T z_{k-1}}\right| \|s_{k-1}\| \\
&\leq \|F(t_k)\| + 2\left(2\frac{\|F(t_k)\|^2}{\|F(t_{k-1)}\|^2} + \alpha_k \frac{\|F(t_k)\|}{\|d_{k-1}\|^2}\alpha_{k-1}\|d_{k-1}\|\right)\|s_{k-1}\| \\
&\leq \|F(t_k)\| + 2\alpha_{k-1}\|d_{k-1}\|\left(2\frac{\|F(t_k)\|^2}{\|F(t_{k-1})\|^2} + \alpha_k \frac{\|F(t_k)\|}{\|d_{k-1}\|^2}\alpha_{k-1}\|d_{k-1}\|\right) \\
&\leq \|F(t_k)\| + 2\left(2\frac{\|F(t_k)\|^2}{\|F(t_{k-1})\|^2}\alpha_{k-1}\|d_{k-1}\| + \frac{\|F(t_k)\|}{\|d_{k-1}\|}\alpha_{k-1}^2\|d_{k-1}\|\right)
\end{aligned}$$

$$\begin{aligned}
&\leq \|F(t_k)\| + 2\left(2\frac{\|F(t_k)\|^2}{\|F(t_{k-1})\|^2}\alpha_{k-1}\|d_{k-1}\| + \|F(t_k)\|\right) \\
&\leq 3\|F(t_k)\| + 4\frac{\|F(t_k)\|^2}{\|F(t_{k-1})\|^2}\alpha_{k-1}\|d_{k-1}\| \\
&\leq 3\bar{m}_2 + 4\frac{\bar{m}_2^2}{\epsilon^2}\alpha_{k-1}\|d_{k-1}\| \triangleq b_2. \qquad (40)
\end{aligned}$$

For all $k \in \mathbb{N}$, (28) implies for every $\epsilon_1 > 0$, there exists an index $k_0 \in \mathbb{N}$ such that $\alpha_{k-1}\|d_{k-1}\| < \epsilon_1 \forall k > k_0$. If we choose $\epsilon_1 = \epsilon_0^2$ and $b_2 = \max\{\|d_0\|, \|d_1\|, ..., \|d_k\|, b_2^*\}$, where $b_2^* = \bar{m}_2(3 + 4\bar{m}_2)$, it holds $\|d_k\| \leq b_2$ for every $k > k_0 \in \mathbb{N}$.

Let $\tilde{m} = \max\{\bar{m}_2, b_1, b_2\}$, then the inequality (36) holds.

Theorem 1. *Suppose that (A1)-(A3) hold. If $\{x_k\}$ is the sequence generated by Algorithm 1. Then $\{x_k\}$ converges to solution of (1).*

Proof. Considering (21), (34), and (36), it follows that $0 \leq \alpha\|d_k\| \leq \alpha_k\|d_k\| \to 0$, which implies $\lim \|d_k\| = 0$. This, coupled with relation (37), indicates that

$$\lim_{k\to\infty} \|F(t_k)\| = 0. \qquad (41)$$

In view of the inertial extrapolation term, and (17), we obtain

$$\|x_k - t_k\| = \|x_k - (x_k + \beta_k(x_k - x_{k-1})\| = \lambda_k\|x_k - x_{k-1}\| \to 0. \qquad (42)$$

Meanwhile, by Assumption (A3), one gets,

$$\begin{aligned}
\|F(x_k)\| &= \|F(x_k) - F(t_k) + F(t_k)\| \\
&\leq \|F(x_k) - F(t_k)\| + \|F(t_k)\| \\
&\leq L\|x_k - t_k\| + \|F(t_k)\|.
\end{aligned}$$

Therefore, taking into account (41) and (42), we can conclude that $\lim_{k\to\infty} \|F(x_k)\| = 0$. Recall that $\{x_k\}$ is bounded, so it guarantees at least one accumulation point of $\{x_k\}$. Let assume $\bar{x} \in \mathcal{C}$ is an accumulation point of $\{x_k\}$ and $\mathcal{K}$ be an infinite index set such that $\lim_{k\in\mathcal{K},k\to\infty} x_k = \bar{x} \in \mathcal{C}$. Then

$$\lim_{k\in\mathcal{K},k\to\infty} \|F(x_k)\| = \lim_{k\to\infty} \|F(x_k)\| = \|F(\bar{x})\| = 0, \qquad (43)$$

which indicates that $\bar{x}$ is in the solution set of (1). Setting $x^* = \bar{x} \in \Gamma$ together with the convergence of $\{\|x_k - x^*\|\}$, we obtain

$$\lim_{k\to\infty} \|x_k - \bar{x}\| = \lim_{k\in\mathcal{K},k\to\infty} \|x_k - \bar{x}\| = 0. \qquad (44)$$

Therefore, the entire sequence $\{x_k\}$ converges to $\bar{x} \in \Gamma$.

4. Numerical Experiment

In this section, we present an experiment to show the efficiency of Algorithm (1) under assumptions A1-A3. Also, an efficiency comparison is made between the proposed method named IDLPA with other existing two algorithms named CPDY [25] denoted by NPDY in the graphs and FIDFP [39] designed to solve monotone nonlinear equation. The IDPLA with New Line Search (NLS) is a modification of the Dai-Liao type projection method by incorporating an inertial term and a New Line Search. The Number of Iteration denoted by (NIT), Number of Functions Evaluations denoted by (NFV) and CPU time denoted by (CPUT) are the metrics considered for comparison. The method with the least value of the metric is considered to be better or more efficient. The following were also considered during the experiments;

* Dimensions: 1000, 5000, 10000, 50000.
* Parameters: For Algorithm (1), we choose $\epsilon_k = \frac{1}{k^2}$, $\rho = 0.6$, $\beta = 0.08$, $\gamma = 0.01$, $\kappa = 1$, $\varrho = 1.2$, $\mu = 1.2$, $\eta_1 = 0.1$, $\eta_2 = 0.5$, $r = 10^{-3}$, $\epsilon = 10^{-6}$.
* Terminating criterion: When $f(t_k) < \epsilon$, or maximum iteration (max it)=1,000.
* Implementation software: All methods were coded using MATLAB 2023B and run on a PC of 32GB RAM and CPU 3.70GHz.

Below are the test problems used for the experiments with $F = (F_1, F_2, ..., F_n)^T$.

Problem 1. *: Modified exponential function [7]*

$$\begin{aligned} F_1(x) &= e^{x1} - 1, \\ F_i(x) &= e^{xi} + x_i - 1, \quad i = 1, 2, ..., n-1, \\ \mathcal{C} &= R^n_+. \end{aligned}$$

Problem 2. *: Modified logarithmic function [7]*

$$\begin{aligned} F_i(x) &= log(x_i + 1) - \frac{x_i}{n}, \quad i = 1, 2, ..., n, \\ \mathcal{C} &= \left\{ x \in \mathbf{R^n_+} : \mathbf{x} \geq \mathbf{-1}, \quad \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i} \leq \mathbf{n} \right\}. \end{aligned}$$

Problem 3. *: Nonsmooth function [40]*

$$F_i(x) = 2x_i - sin(|x_i|) \quad for \quad i = 1, 2, ..., n,$$

$$\mathcal{C} = \left\{x \in \mathbf{R}^{\mathbf{n}}_{+} : \mathbf{x} \geq \mathbf{0}, \quad \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i} \leq \mathbf{n}\right\}.$$

Problem 4. *[41]*

$$F_i(x) = \min\left(\min(|x_i|, x_i^2), \max(|x_i|, x_i^3)\right) \text{ for } i = 2, 3, ..., n, \quad \text{and } \mathcal{C} = \mathbb{R}^n_+.$$

Problem 5. *: Strictly convex function I [7]*

$$\begin{aligned} F_i(x) &= e^{xi} - 1 \quad i = 1, 2, ..., n, \\ \mathcal{C} &= R^n_+. \end{aligned}$$

Problem 6. *: Strictly convex function II [7]*

$$\begin{aligned} F_i(x) &= \left(\frac{i}{n}\right) e^{xi} - 1 \quad i = 1, 2, ..., n, \\ \mathcal{C} &= R^n_+. \end{aligned}$$

Problem 7. *: Tridiagonal exponential function [7]*

$$\begin{aligned} F_1(x) &= x_1 - e^{cos(h(x_1+x_2))} \\ F_i(x) &= x_i - e^{cos(h(x_i-1+x_i+x_i+1))}, \quad i = 2, ..., n-1, \\ F_n(x) &= x_n - e^{cos(h(x_n-1+x_n))}, \\ h &= \frac{1}{n+1} \quad and \quad \mathcal{C} = R^n_+. \end{aligned}$$

Problem 8. *: Nonsmooth function II [23]*

$$\begin{aligned} F_i(x) &= x_i - sin(|x_i - 1|) \quad for \quad i = 1, 2, ..., n, \\ \mathcal{C} &= \left\{x \in \mathbf{R}^{\mathbf{n}}_{+} : \mathbf{x} \geq -\mathbf{1}, \quad \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{x_i} \leq \mathbf{n}\right\}. \end{aligned}$$

Problem 9. *: Trig exp function [42] with constraint set $\mathcal{C} = R^n_+$, that is,*

$F_1(x) = 3x_1^3 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2)$
$F_i(x) = 3x_i^3 + 2x_{i+1} - 5 + \sin(x_i - x_{i+1})\sin(x_i + x_{i+1}) + 4x_i - x_{i-1}e^{x_{i-1}-x_i} - 3 \;\; for\, i = 2, 3, ..., n-1$
$F_n(x) = x_{n-1}e^{x_{n-1}-x_n} - 4x_n - 3, \;\; where\ h = \dfrac{1}{n+1}.$

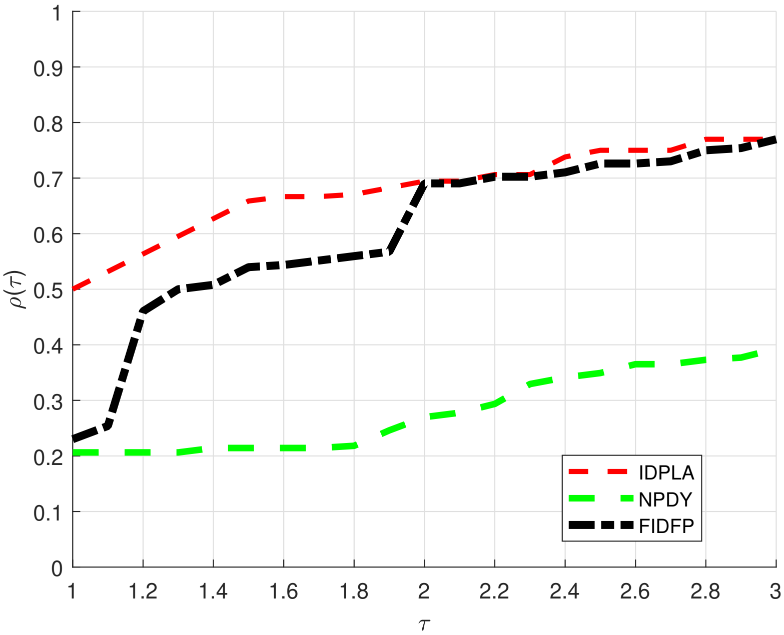

Figure 1: Performance profile of IDPLA, NPDY and FIDFP methods in terms of number of iterations

The result of the experiments can be found in Tables 1-9. The Dolan and More performance profile [43] is employed in order to have a clear visualization of the efficiency of the algorithms. The performance profile is considered as a tool for evaluating and comparing the performances of iterative methods. The performance profile by Dolan and More is defined as ;

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

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 (NF or CPU) for $t_p \in T_p$ and $q \in Q$.

By using Dolan and More [43] performance profile tool, from figures $1-3$ obtained, in terms of number of iteration (NIT) in figure (1) the performances of the considered methods are competitive, but the IDPLA performs better than the other two methods with solving about 50% of the test problems with least number of iterations. Also, considering the number of function evaluation (NFV) and CPU time (CPUT) in figures $(2, 3)$ respectively, the IDPLA has the least number of function evaluation (NFV) of 42% and CPU time (CPUT) of 48% of the tests problems compared to the other two methods. The good numerical performance of the method (IDPLA) compared to the other two, shows that the purpose of introducing the inertial effect and the new line search (NLS) is achieved as the convergence of 1 is faster in most of the problems. The detailed results for each of the nine test problems, comparing the three algorithms in terms of number of iteration

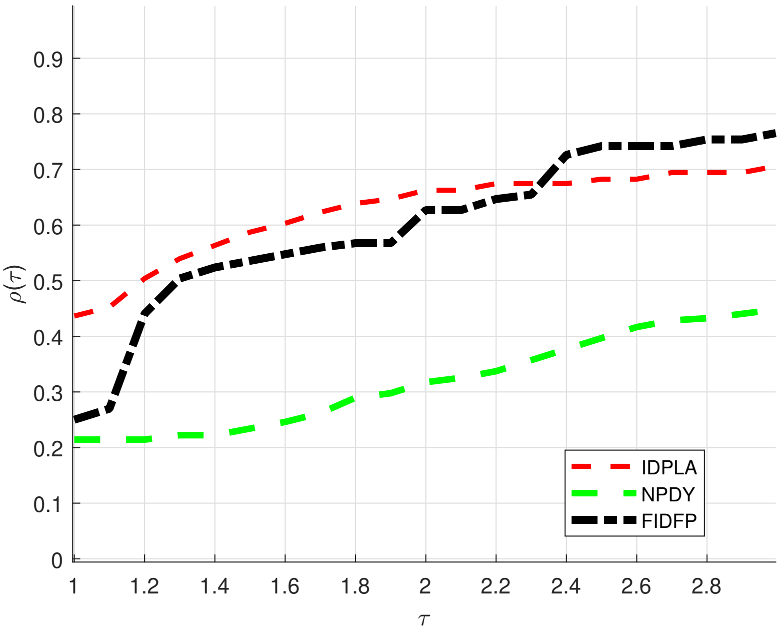

Figure 2: Performance profile of IDPLA, NPDY and FIDFP methods in terms of number of function evaluation

(NI), number of function evaluations(NFV) and CPU time, are provided in Tables 1-9 (See Appendix). These results form the basis for the performance profiles shown in Figures 1-3

5. Conclusion

In this work, a derivative-free projection method for solving monotone nonlinear equations is proposed. This method is a combination of the Dai-Liao type projection method of Ibrahim *et.al,* [27] with an inertial term and a new line search to speed up the convergence of the method as well as ensuring the efficiency of the method. Under standard assumptions, the convergence of the method is obtained. Numerical experiments were performed to support the results obtained in theoretical convergence result. These numerical results confirms the fast convergence and efficiency of our method over the existing ones.

Acknowledgements

The authors wish to express their profound appreciation to Dr. A.H. Ibrahim (Faculty of Mathematics and Data Science, Emirates Aviation University, Dubai, UAE) for his expert guidance and valuable assistance in the numerical implementation of this study.

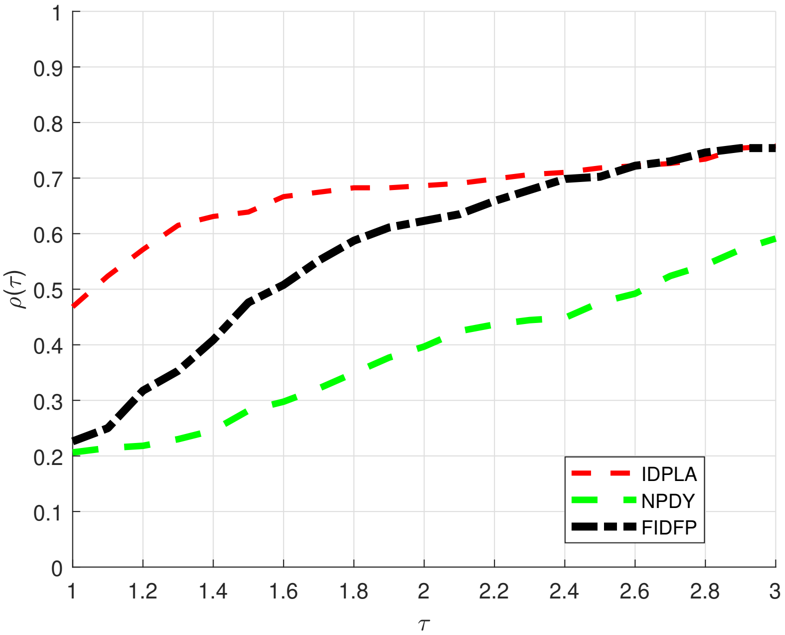

Figure 3: Performance profile of IDPLA, NPDY and FIDFP methods in terms of CPU time

Declarations

Funding: The project is not funded.

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of data and materials: Not applicable.

Code availability: Not applicable

Authors' contributions: All authors contributed equally to the manuscript, read and approve the final manuscript.

References

[1] Bissan Ghaddar, Jakub Marecek, and Martin Mevissen. Optimal power flow as a polynomial optimization problem. *IEEE Transactions on Power Systems*, 31(1):539–546, 2015.

[2] Zhifeng Dai and Jie Kang. Some new efficient mean–variance portfolio selection models. *International Journal of Finance & Economics*, 27(4):4784–4796, 2022.

[3] Zhifeng Dai, Xiaodi Dong, Jie Kang, and Lianying Hong. Forecasting stock market returns: New technical indicators and two-step economic constraint method. *The North American Journal of Economics and Finance*, 53(C), 2020.

[4] Wenyu Sun and Ya-Xiang Yuan. *Optimization theory and methods: nonlinear programming*, volume 1. Springer Science & Business Media, 2006.

[5] N Alfredo Iusem and V Michael Solodov. Newton-type methods with generalized distances for constrained optimization. *Optimization*, 41(3):257–278, 1997.

[6] Nocedal Jorge and J Wright Stephen. *Numerical optimization*. Spinger, 2006.

[7] William La Cruz, José Martínez, and Marcos Raydan. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Mathematics of computation*, 75(255):1429–1448, 2006.

[8] William La Cruz and Marcos Raydan. Nonmonotone spectral methods for large-scale nonlinear systems. *Optimization Methods and software*, 18(15):583–599, 2003.

[9] Abdulkarim Hassan Ibrahim, Poom Kumam, Ahmad Kamandi, and Auwal Bala Abubakar. An efficient hybrid conjugate gradient method for unconstrained optimization. *Optimization Methods and Software*, 37(4):1370–1383, 2022.

[10] Abdulkarim Hassan Ibrahim, Poom Kumam, Auwal Bala Abubakar, and Jamilu Abubakar. A method with inertial extrapolation step for convex constrained monotone equations. *Journal of Inequalities and Applications*, 2021(1):1–25, 2021.

[11] Abdulkarim Hassan Ibrahim, Morteza Kimiaei, and Poom Kumam. A new black box method for monotone nonlinear equations. *Optimization*, 72(5):1119–1137, 2023.

[12] Abdulkarim Hassan Ibrahim, Jitsupa Deepho, Auwal Bala Abubakar, and Abubakar Adamu. A three-term polak-ribière-polyak derivative-free method and its application to image restoration. *Scientific African*, 13:e00880, 2021.

[13] Abdulkarim Hassan Ibrahim, Poom Kumam, Basim A Hassan, Auwal Bala Abubakar, and Jamilu Abubakar. A derivative-free three-term hestenes–stiefel type method for constrained nonlinear equations and image restoration. *International Journal of Computer Mathematics*, 99(5):1041–1065, 2022.

[14] Auwal Bala Abubakar, Poom Kumam, Abdulkarim Hassan Ibrahim, Parin Chaipunya, and Sadiya Ali Rano. New hybrid three-term spectral-conjugate gradient method for finding solutions of nonlinear monotone operator equations with applications. *Mathematics and Computers in Simulation*, 201:670–683, 2022.

[15] Yutao Zheng and Bing Zheng. Two new dai–liao-type conjugate gradient methods for unconstrained optimization problems. *Journal of Optimization Theory and Applications*, 175:502–509, 2017.

[16] William W Hager and Hongchao Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, 16(1):170–192, 2005.

[17] William W Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006.

[18] Wei-Jun Zhou and Dong-Hui Li. A globally convergent bfgs method for nonlin-

ear monotone equations without any merit functions. *Mathematics of Computation*, 77(264):2231–2240, 2008.

[19] Yunhai Xiao and Hong Zhu. A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing. *Journal of Mathematical Analysis and Applications*, 405(1):310–319, 2013.

[20] Michael V Solodov and Benav F Svaiter. A globally convergent inexact newton method for systems of monotone equations. *Reformulation: Nonsmooth, piecewise smooth, semismooth and smoothing methods*, pages 355–369, 1999.

[21] Li Zhang and Weijun Zhou. Spectral gradient projection method for solving nonlinear monotone equations. *Journal of Computational and Applied Mathematics*, 196(2):478–484, 2006.

[22] Chuanwei Wang, Yiju Wang, and Chuanliang Xu. A projection method for a system of nonlinear monotone equations with convex constraints. *Mathematical Methods of Operations Research*, 66:33–46, 2007.

[23] Zhensheng Yu, Ji Lin, Jing Sun, Yunhai Xiao, Liying Liu, and Zhanhui Li. Spectral gradient projection method for monotone nonlinear equations with convex constraints. *Applied numerical mathematics*, 59(10):2416–2423, 2009.

[24] Peiting Gao and Chuanjiang He. An efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints. *Calcolo*, 55:1–17, 2018.

[25] Jinkui Liu and Yuming Feng. A derivative-free iterative method for nonlinear monotone equations with convex constraints. *Numerical Algorithms*, 82:245–262, 2019.

[26] Zhifeng Dai and Huan Zhu. A modified hestenes-stiefel-type derivative-free method for large-scale nonlinear monotone equations. *Mathematics*, 8(2):168, 2020.

[27] Abdulkarim Hassan Ibrahim, Poom Kumam, Auwal Bala Abubakar, Muhammad Sirajo Abdullahi, and Hassan Mohammad. A dai-liao-type projection method for monotone nonlinear equations and signal processing. *Demonstratio Mathematica*, 55(1):978–1013, 2022.

[28] Predrag S Stanimirović, Branislav Ivanov, Snežana Djordjević, and Ivona Brajević. New hybrid conjugate gradient and broyden–fletcher–goldfarb–shanno conjugate gradient methods. *Journal of Optimization Theory and Applications*, 178:860–884, 2018.

[29] Yigui Ou and Jingya Li. A new derivative-free scg-type projection method for nonlinear monotone equations with convex constraints. *Journal of Applied Mathematics and Computing*, 56:195–216, 2018.

[30] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

[31] Hédy Attouch, Juan Peypouquet, and Patrick Redont. A dynamical approach to an inertial forward-backward algorithm for convex minimization. *SIAM Journal on Optimization*, 24(1):232–256, 2014.

[32] Dirk A Lorenz and Thomas Pock. An inertial forward-backward algorithm for monotone inclusions. *Journal of Mathematical Imaging and Vision*, 51:311–325, 2015.

[33] Duong Viet Thong and Dang Van Hieu. An inertial method for solving split common fixed point problems. *Journal of Fixed Point Theory and Applications*, 19:3029–3051, 2017.

[34] Duong Viet Thong and Dang Van Hieu. Modified subgradient extragradient method for variational inequality problems. *Numerical Algorithms*, 79:597–610, 2018.

[35] Qiao-Li Dong, YJ Cho, LL Zhong, and Th M Rassias. Inertial projection and contraction algorithms for variational inequalities. *Journal of Global Optimization*, 70:687–704, 2018.

[36] Abdulkarim Hassan Ibrahim, Poom Kumam, Min Sun, Parin Chaipunya, and Auwal Bala Abubakar. Projection method with inertial step for nonlinear equations: application to signal recovery. *Journal of Industrial and Management Optimization*, 19(1):30–55, 2022.

[37] Abdulkarim Hassan Ibrahim, Mohammed Alshahrani, and Suliman Al-Homidan. Two classes of spectral three-term derivative-free method for solving nonlinear equations with application. *Numerical Algorithms*, pages 1–21, 2023.

[38] Alfred Auslender, Marc Teboulle, and Sami Ben-Tiba. A logarithmic-quadratic proximal method for variational inequalities. *Computational Optimization: A Tribute to Olvi Mangasarian Volume I*, pages 31–40, 1999.

[39] Jinbao Jian, Jianghua Yin, Chunming Tang, and Daolan Han. A family of inertial derivative-free projection methods for constrained nonlinear pseudo-monotone equations with applications. *Computational and Applied Mathematics*, 41(7):309, 2022.

[40] Qingna Li and Dong-Hui Li. A class of derivative-free methods for large-scale nonlinear monotone equations. *IMA Journal of Numerical Analysis*, 31(4):1625–1635, 2011.

[41] William La Cruz. A spectral algorithm for large-scale systems of nonlinear monotone equations. *Numerical Algorithms*, 76:1109–1130, 2017.

[42] Abdulkarim Hassan Ibrahim, Poom Kumam, Auwal Bala Abubakar, Wachirapong Jirakitpuwapat, and Jamilu Abubakar. A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing. *Heliyon*, 6(3), 2020.

[43] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91:201–213, 2002.

Appendix A

Table 1: Test Results of the Three Algorithms on Problem 1

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
1000	x_1	2	18	0.001435	0	14	98	0.011526	6.53E-07	11	79	0.0118	8.66E-07
	x_2	5	43	0.002053	7.26E-08	14	98	0.006486	5.08E-07	21	171	0.010406	8.45E-07
	x_3	3	27	0.001795	0	14	98	0.003625	9.13E-07	21	171	0.006287	6.49E-07
	x_4	8	66	0.002649	5.88E-07	15	107	0.003826	9.06E-07	12	86	0.003891	5.06E-07
	x_5	5	45	0.002348	0	15	107	0.006903	7.77E-07	12	86	0.00578	6.13E-07
	x_6	2	20	0.001694	0	15	107	0.003269	6.62E-07	22	182	0.005906	6.30E-07
	x_7	8	72	0.002692	4.25E-08	17	119	0.003574	3.22E-07	20	164	0.004795	5.35E-07
5000	x_1	2	18	0.003133	0	15	105	0.008487	3.41E-07	15	141	0.015155	4.19E-07
	x_2	5	43	0.003893	7.65E-08	15	105	0.009702	3.40E-07	21	171	0.018683	9.23E-07
	x_3	3	27	0.003531	0	15	105	0.009108	6.10E-07	22	180	0.017236	7.11E-07
	x_4	7	59	0.005093	9.47E-07	16	114	0.010099	5.94E-07	22	192	0.021624	6.19E-07
	x_5	5	45	0.004852	0	16	114	0.009959	4.91E-07	11	103	0.011328	8.81E-11
	x_6	2	20	0.003069	0	16	114	0.010684	3.81E-07	25	203	0.01936	1.50E-08
	x_7	8	72	0.006377	6.96E-08	17	119	0.011262	7.41E-07	22	180	0.022138	5.77E-07
10000	x_1	2	18	0.00427	0	15	105	0.016725	4.64E-07	12	118	0.01902	5.70E-08
	x_2	5	43	0.007976	7.63E-08	15	105	0.013976	4.81E-07	22	180	0.02805	6.45E-07
	x_3	3	27	0.007421	0	15	105	0.01354	8.62E-07	22	180	0.032226	4.95E-07
	x_4	7	59	0.011223	9.19E-07	16	114	0.018528	8.38E-07	13	127	0.025261	1.17E-07
	x_5	5	45	0.007432	0	16	114	0.015694	6.89E-07	15	141	0.023112	5.32E-08
	x_6	2	20	0.004143	0	16	114	0.014507	5.27E-07	23	191	0.031047	5.00E-07
	x_7	8	72	0.009943	1.22E-07	18	126	0.01872	3.19E-07	22	180	0.034824	8.21E-07
50000	x_1	2	18	0.013645	0	16	112	0.08303	3.02E-07	4	28	0.023993	0
	x_2	5	43	0.02616	7.59E-08	16	112	0.074776	3.22E-07	22	180	0.15395	7.08E-07
	x_3	3	27	0.01853	0	16	112	0.071748	5.78E-07	23	189	0.15211	5.51E-07
	x_4	7	59	0.036648	8.96E-07	17	121	0.07959	5.61E-07	4	28	0.025347	0
	x_5	5	45	0.029737	0	17	121	0.076425	4.59E-07	4	28	0.027026	0
	x_6	2	20	0.013455	0	17	121	0.093836	3.48E-07	26	212	0.17137	1.19E-08
	x_7	8	72	0.046767	2.48E-07	18	126	0.083654	6.78E-07	21	173	0.13517	8.99E-07

Table 2: Test Results of the Three Algorithms on Problem 2

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
1000	x_1	11	73	0.0252	6.25E-07	5	25	0.002844	3.60E-08	8	40	0.003874	6.11E-07
	x_2	10	68	0.002917	7.49E-07	4	20	0.001956	6.04E-09	10	50	0.003394	4.58E-07
	x_3	10	68	0.002811	5.09E-07	4	20	0.001642	4.37E-07	12	60	0.003814	2.59E-07
	x_4	11	73	0.00295	4.96E-07	5	25	0.00192	1.52E-07	5	25	0.002357	4.36E-07
	x_5	11	73	0.003789	6.22E-07	6	30	0.002144	1.10E-09	10	50	0.003157	6.35E-08
	x_6	11	73	0.003018	5.76E-07	6	30	0.001977	1.74E-08	12	60	0.00427	4.61E-07
	x_7	11	73	0.003779	6.84E-07	61	341	0.015467	7.69E-07	16	98	0.005736	1.06E-07
5000	x_1	11	73	0.007849	7.74E-07	5	25	0.003229	6.26E-09	9	45	0.007255	8.36E-07
	x_2	10	68	0.007294	9.32E-07	4	20	0.00314	6.27E-10	11	55	0.008675	8.62E-08
	x_3	11	75	0.009256	3.37E-07	4	20	0.003428	1.42E-07	12	60	0.008666	5.97E-07
	x_4	12	80	0.008951	3.24E-07	5	25	0.003153	3.94E-08	5	25	0.004648	7.99E-07
	x_5	12	80	0.010034	4.09E-07	5	25	0.003802	4.05E-07	10	50	0.007683	1.72E-07
	x_6	12	80	0.008438	3.84E-07	6	30	0.003825	2.36E-09	13	65	0.009858	4.65E-07
	x_7	12	80	0.010763	4.62E-07	59	337	0.053852	4.38E-08	12	72	0.011678	5.09E-07
10000	x_1	12	80	0.017561	3.32E-07	5	25	0.007108	3.62E-09	10	50	0.012953	5.01E-08
	x_2	11	75	0.013277	4.00E-07	4	20	0.004263	2.79E-10	11	55	0.016766	1.22E-07
	x_3	11	75	0.014165	4.75E-07	4	20	0.004038	9.73E-08	12	60	0.014677	8.68E-07
	x_4	12	80	0.017221	4.56E-07	5	25	0.005628	2.56E-08	6	30	0.007951	7.29E-07
	x_5	12	80	0.014829	5.75E-07	5	25	0.00519	2.93E-07	10	50	0.0146	3.13E-07
	x_6	12	80	0.016273	5.42E-07	6	30	0.006311	1.24E-09	13	65	0.019176	6.48E-07
	x_7	12	80	0.018426	6.58E-07	41	231	0.051577	5.64E-07	13	79	0.022572	4.05E-09
50000	x_1	12	80	0.071067	7.38E-07	5	25	0.02449	1.34E-09	10	50	0.064891	1.58E-07
	x_2	11	75	0.064615	5.12E-07	4	20	0.019727	6.75E-11	11	55	0.068908	1.87E-07
	x_3	11	75	0.060008	6.09E-07	4	20	0.01973	4.87E-08	13	65	0.072372	4.48E-07
	x_4	12	80	0.064539	5.84E-07	5	25	0.022906	1.11E-08	7	35	0.037274	2.94E-08
	x_5	12	80	0.074048	7.36E-07	5	25	0.024092	1.84E-07	10	50	0.06534	4.45E-07
	x_6	12	80	0.065033	6.95E-07	6	30	0.029659	4.01E-10	14	70	0.080274	1.52E-07
	x_7	12	80	0.078366	8.32E-07	63	357	0.37446	9.64E-07	11	69	0.09023	7.91E-08

Table 3: Test Results of the Three Algorithms on Problem 3

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
1000	x_1	10	70	0.002993	6.71E-07	40	280	0.012903	9.01E-07	12	84	0.004024	6.44E-07
	x_2	10	70	0.002538	5.25E-07	37	257	0.009911	0	12	84	0.004705	5.15E-07
	x_3	10	70	0.002658	6.21E-07	39	271	0.00882	0	12	84	0.003127	5.58E-07
	x_4	16	116	0.005526	5.66E-07	41	285	0.008331	0	12	84	0.003338	9.48E-07
	x_5	12	88	0.003118	5.74E-07	41	287	0.008522	7.56E-07	13	91	0.003459	6.87E-07
	x_6	11	77	0.003188	7.09E-07	41	285	0.00802	0	13	91	0.003567	6.90E-07
	x_7	13	91	0.003728	5.69E-07	39	273	0.00959	8.38E-07	12	84	0.00358	5.27E-07
5000	x_1	10	70	0.007038	8.56E-07	40	278	0.028096	0	13	91	0.011957	7.12E-07
	x_2	10	70	0.00724	6.74E-07	37	257	0.025409	9.36E-22	12	84	0.010476	5.64E-07
	x_3	11	77	0.008045	4.22E-07	39	271	0.030729	4.68E-22	13	91	0.010775	6.17E-07
	x_4	16	116	0.01196	7.21E-07	42	292	0.028362	0	13	91	0.0109	5.13E-07
	x_5	12	88	0.008338	7.31E-07	43	299	0.028073	4.68E-22	14	98	0.013464	7.78E-07
	x_6	11	77	0.009682	9.05E-07	42	292	0.029543	0	14	98	0.014128	7.81E-07
	x_7	13	91	0.011875	7.25E-07	41	287	0.035506	8.02E-07	13	91	0.013935	5.83E-07
10000	x_1	11	77	0.014573	3.66E-07	40	278	0.048821	1.32E-21	13	91	0.020386	4.94E-07
	x_2	10	70	0.011591	9.53E-07	38	264	0.037225	6.62E-22	12	84	0.020874	7.97E-07
	x_3	11	77	0.013663	5.96E-07	39	271	0.038254	1.32E-21	13	91	0.024147	8.73E-07
	x_4	17	123	0.02043	3.07E-07	42	292	0.050179	1.32E-21	13	91	0.021712	7.26E-07
	x_5	13	95	0.017582	3.12E-07	43	299	0.054612	6.62E-22	14	98	0.020205	5.44E-07
	x_6	12	84	0.017885	3.86E-07	41	285	0.049473	1.32E-21	14	98	0.020597	5.45E-07
	x_7	14	98	0.021706	3.09E-07	42	294	0.062758	7.33E-07	13	91	0.025439	8.27E-07
50000	x_1	11	77	0.053668	4.66E-07	41	285	0.2192	6.66E-21	14	98	0.10327	5.62E-07
	x_2	11	77	0.054663	6.47E-07	36	250	0.18894	8.88E-21	13	91	0.095231	8.82E-07
	x_3	11	77	0.061685	7.64E-07	37	257	0.19775	0	13	91	0.08938	9.79E-07
	x_4	17	123	0.090153	6.87E-07	39	271	0.20756	5.92E-21	13	91	0.087534	8.14E-07
	x_5	13	95	0.064845	6.97E-07	43	299	0.23296	2.96E-21	15	105	0.1063	6.18E-07
	x_6	12	84	0.065308	4.93E-07	44	306	0.22795	2.22E-21	15	105	0.10357	6.20E-07
	x_7	14	98	0.090112	6.91E-07	261	1817	1.6337	7.41E-07	13	91	0.098636	9.24E-07

Table 4: Test Results of the Three Algorithms on Problem 4

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
1000	x_1	11	77	0.011498	3.99E-07	1	5	0.003935	0	2	12	0.002634	0
	x_2	10	68	0.003421	4.54E-07	1	5	0.001591	0	2	10	0.001513	0
	x_3	10	68	0.003377	6.54E-07	1	5	0.000789	0	2	10	0.001344	0
	x_4	11	77	0.003699	4.79E-07	1	5	0.000843	0	2	12	0.001417	0
	x_5	11	77	0.00409	5.99E-07	1	5	0.001926	0	2	12	0.001428	0
	x_6	11	77	0.003625	4.59E-07	1	5	0.000796	0	2	12	0.001236	0
	x_7	11	73	0.004133	6.59E-07	1	5	0.001269	0	14	88	0.005137	9.65E-07
5000	x_1	11	77	0.010418	5.13E-07	1	5	0.001186	0	2	12	0.003133	0
	x_2	10	68	0.009364	5.85E-07	1	5	0.001333	0	2	10	0.002751	0
	x_3	11	75	0.010835	4.46E-07	1	5	0.001331	0	2	10	0.002389	0
	x_4	11	77	0.012966	6.16E-07	1	5	0.001502	0	2	12	0.002704	0
	x_5	11	77	0.013265	7.69E-07	1	5	0.001328	0	2	12	0.002911	0
	x_6	12	84	0.013832	3.12E-07	1	5	0.001721	0	2	12	0.002904	0
	x_7	11	73	0.011808	8.72E-07	1	5	0.001333	0	13	87	0.017411	6.20E-07
10000	x_1	11	77	0.026357	7.25E-07	1	5	0.001799	0	2	12	0.004168	0
	x_2	10	68	0.017871	8.28E-07	1	5	0.001998	0	2	10	0.00452	0
	x_3	11	75	0.020493	6.31E-07	1	5	0.002185	0	2	10	0.00455	0
	x_4	11	77	0.019367	8.71E-07	1	5	0.001967	0	2	12	0.004675	0
	x_5	12	84	0.020937	3.31E-07	1	5	0.001912	0	2	12	0.004162	0
	x_6	12	84	0.020607	4.41E-07	1	5	0.001984	0	2	12	0.004544	0
	x_7	13	85	0.022692	3.14E-07	1	5	0.00203	0	15	95	0.032278	7.26E-07
50000	x_1	12	84	0.10469	4.93E-07	1	5	0.005672	0	2	12	0.017055	0
	x_2	11	75	0.097604	5.64E-07	1	5	0.008993	0	2	10	0.019951	0
	x_3	11	75	0.080606	8.11E-07	1	5	0.007343	0	2	10	0.015905	0
	x_4	12	84	0.097711	5.92E-07	1	5	0.009436	0	2	12	0.019977	0
	x_5	12	84	0.099127	7.40E-07	1	5	0.006321	0	2	12	0.020134	0
	x_6	12	84	0.097686	5.67E-07	1	5	0.006738	0	2	12	0.017065	0
	x_7	13	85	0.10358	4.04E-07	1	5	0.008954	0	16	102	0.14568	8.16E-07

Table 5: Test Results of the Three Algorithms on Problem 5

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
1000	x_1	10	72	0.006148	6.57E-07	39	273	0.008472	7.22E-07	7	37	0.002229	3.60E-08
	x_2	9	63	0.002042	7.74E-07	36	252	0.010298	9.98E-07	11	77	0.003174	6.61E-07
	x_3	3	21	0.001231	0	38	266	0.006297	8.38E-07	10	70	0.00309	7.47E-07
	x_4	9	65	0.002153	7.73E-07	39	273	0.006844	7.08E-07	7	37	0.001696	4.32E-08
	x_5	10	70	0.002574	8.70E-07	38	266	0.006516	8.58E-07	12	86	0.003032	6.29E-07
	x_6	8	60	0.002151	0	40	282	0.006348	7.05E-07	7	39	0.001754	7.16E-08
	x_7	13	93	0.002765	4.64E-07	38	266	0.006697	7.86E-07	12	84	0.003124	5.14E-07
5000	x_1	10	72	0.005634	8.33E-07	41	287	0.019285	6.82E-07	7	37	0.004352	8.06E-08
	x_2	10	70	0.005451	5.24E-07	38	266	0.019369	9.43E-07	12	84	0.00771	7.32E-07
	x_3	3	21	0.002644	0	40	280	0.023729	7.91E-07	11	77	0.007188	8.26E-07
	x_4	10	72	0.005332	5.04E-07	41	287	0.019257	6.69E-07	7	37	0.004244	9.66E-08
	x_5	11	77	0.006214	5.93E-07	40	280	0.018893	8.11E-07	13	93	0.009027	7.05E-07
	x_6	8	60	0.005287	0	42	296	0.019694	6.66E-07	7	39	0.004486	1.60E-07
	x_7	13	93	0.006984	8.68E-07	40	280	0.020248	7.37E-07	12	84	0.009387	5.65E-07
10000	x_1	11	79	0.010659	3.53E-07	41	287	0.037426	9.65E-07	7	37	0.008127	1.14E-07
	x_2	10	70	0.009829	7.41E-07	39	271	0.032708	0	12	84	0.013132	5.07E-07
	x_3	3	21	0.003956	0	41	287	0.029114	7.27E-07	11	77	0.013907	5.72E-07
	x_4	10	72	0.008637	7.12E-07	41	287	0.040725	9.47E-07	7	37	0.008459	1.37E-07
	x_5	11	77	0.009203	4.82E-07	41	287	0.031563	7.45E-07	13	93	0.018001	4.97E-07
	x_6	8	60	0.00786	0	42	296	0.036997	9.41E-07	7	39	0.007552	2.26E-07
	x_7	15	109	0.016411	7.03E-07	41	287	0.037478	6.77E-07	12	84	0.01322	8.27E-07
50000	x_1	11	79	0.042499	4.48E-07	43	299	0.14652	0	7	37	0.033158	2.55E-07
	x_2	10	70	0.034678	9.48E-07	40	278	0.1332	0	13	91	0.064437	5.60E-07
	x_3	3	21	0.017826	0	42	292	0.14194	0	12	84	0.061562	6.33E-07
	x_4	10	72	0.03808	8.85E-07					7	37	0.032163	3.05E-07
	x_5	12	84	0.04334	3.28E-07	43	301	0.13346	7.04E-07	14	100	0.071571	5.62E-07
	x_6	8	60	0.033015	0					7	39	0.037219	5.06E-07
	x_7	17	123	0.056924	1.87E-07	42	294	0.14465	9.88E-07	13	91	0.082438	5.15E-07

Table 6: Test Results of the Three Algorithms on Problem 6

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
1000	x_1	15	107	0.016514	5.66E-07	485	2505	0.087155	5.70E-12	18	116	0.005842	7.03E-07
	x_2	13	85	0.007322	6.90E-07	40	268	0.006962	6.61E-07	14	92	0.003514	8.02E-07
	x_3	14	96	0.004573	3.70E-07					16	106	0.003682	9.23E-07
	x_4	13	93	0.003851	9.24E-07					9	59	0.002413	5.95E-07
	x_5	14	102	0.006453	3.39E-07					13	89	0.003377	2.92E-07
	x_6	17	155	0.004528	3.47E-07	55	347	0.00857	2.36E-11	16	112	0.004754	7.41E-07
	x_7	24	220	0.005919	5.38E-07	69	417	0.010567	3.34E-12	19	147	0.005769	9.01E-07
5000	x_1	16	114	0.009787	4.32E-07	377	1971	0.15192	7.39E-07	24	176	0.015648	5.03E-07
	x_2	14	92	0.007664	5.06E-07	47	295	0.025466	9.67E-07	14	92	0.011676	7.87E-07
	x_3	14	96	0.008502	7.37E-07	71	431	0.035291	1.38E-13	21	143	0.016167	8.17E-07
	x_4	14	100	0.007893	5.85E-07								
	x_5	14	102	0.008852	8.67E-07					17	117	0.014653	3.38E-08
	x_6	18	174	0.013299	7.82E-07					13	87	0.009481	3.53E-08
	x_7	29	285	0.021842	4.43E-07	64	388	0.030603	7.35E-12	29	247	0.020922	6.22E-07
10000	x_1	16	114	0.018196	6.52E-07	328	1710	0.21767	3.06E-12	21	149	0.024102	8.26E-07
	x_2	14	92	0.011533	7.49E-07	65	369	0.044415	2.79E-07	16	116	0.021773	5.78E-07
	x_3	15	103	0.012868	3.79E-07	68	408	0.053798	5.03E-12	17	119	0.023581	5.02E-07
	x_4	14	100	0.013012	7.42E-07	589	3033	0.41491	6.93E-07	21	149	0.028558	7.15E-07
	x_5	15	109	0.015043	6.15E-07					23	167	0.025823	9.81E-07
	x_6	22	210	0.028053	6.66E-07	81	493	0.065575	1.94E-12	18	126	0.022475	6.00E-07
	x_7	30	326	0.043706	9.73E-07	73	437	0.054568	2.03E-12				
50000	x_1	17	121	0.067292	6.23E-07	250	1320	0.74801	5.41E-11				
	x_2	15	99	0.054411	3.73E-07	54	306	0.16761	3.17E-07				
	x_3	15	103	0.054458	9.37E-07	69	415	0.2174	1.73E-11				
	x_4	15	107	0.066553	5.10E-07	428	2224	1.2362	6.90E-12				
	x_5	15	109	0.059716	4.71E-07								
	x_6	25	271	0.14099	8.16E-07	249	1339	0.73237	6.67E-07				
	x_7	45	545	0.27796	3.99E-07	72	434	0.2354	2.02E-11				

Table 7: Test Results of the Three Algorithms on Problem 7

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
	x_1	11	77	0.014163	4.64E-07	42	294	0.014339	7.53E-07	13	91	0.006442	5.36E-07
	x_2	11	77	0.003674	6.80E-07	43	301	0.010593	7.17E-07	13	91	0.006032	7.99E-07
	x_3	11	77	0.003872	5.99E-07	42	294	0.010353	9.72E-07	13	91	0.00501	7.01E-07
1000	x_4	11	77	0.005187	7.15E-07	42	294	0.010367	6.65E-07	13	91	0.004267	9.66E-07
	x_5	11	77	0.005043	5.74E-07	41	287	0.011151	8.21E-07	13	91	0.004266	7.75E-07
	x_6	11	77	0.003745	3.38E-07	40	280	0.009782	7.45E-07	12	84	0.003803	9.23E-07
	x_7	11	77	0.003676	6.01E-07	42	294	0.015132	9.88E-07	13	91	0.004051	7.09E-07
	x_1	12	84	0.011835	3.16E-07	44	308	0.03463	7.12E-07	14	98	0.015753	6.13E-07
	x_2	12	84	0.009957	4.63E-07	45	315	0.037909	6.79E-07	14	98	0.015979	9.14E-07
	x_3	12	84	0.010933	4.08E-07	44	308	0.040685	9.20E-07	14	98	0.015836	8.02E-07
5000	x_4	11	77	0.012008	9.21E-07	43	301	0.037998	9.68E-07	14	98	0.013734	5.38E-07
	x_5	11	77	0.010436	7.39E-07	43	301	0.038048	7.77E-07	13	91	0.01827	8.68E-07
	x_6	11	77	0.008746	4.36E-07	42	294	0.036724	7.05E-07	13	91	0.01445	5.01E-07
	x_7	12	84	0.011299	4.10E-07	44	308	0.039252	9.28E-07	14	98	0.01385	8.07E-07
	x_1	12	84	0.017545	4.47E-07	45	315	0.070029	6.55E-07	14	98	0.023324	8.81E-07
	x_2	12	84	0.019701	6.55E-07	45	315	0.061577	9.60E-07	14	98	0.023678	6.38E-07
	x_3	12	84	0.018054	5.77E-07	45	315	0.058706	8.46E-07	14	98	0.025116	5.60E-07
10000	x_4	12	84	0.022946	3.95E-07	44	308	0.05901	8.90E-07	14	98	0.022606	7.75E-07
	x_5	12	84	0.017263	3.17E-07	44	308	0.059381	7.14E-07	14	98	0.023754	6.15E-07
	x_6	11	77	0.016911	6.16E-07	42	294	0.068939	9.97E-07	13	91	0.021147	7.20E-07
	x_7	12	84	0.020687	5.82E-07	45	315	0.058742	8.52E-07	14	98	0.023318	5.64E-07
	x_1	12	84	0.097756	5.73E-07	45	313	0.32478	0	14	98	0.12325	9.90E-07
	x_2	12	84	0.094636	8.39E-07	45	313	0.33369	0	15	105	0.13634	7.26E-07
	x_3	12	84	0.087172	7.39E-07	45	313	0.32289	0	15	105	0.12414	6.38E-07
50000	x_4	12	84	0.083262	5.06E-07	44	306	0.30159	0	14	98	0.11655	8.70E-07
	x_5	12	84	0.090462	7.09E-07	44	306	0.32032	0	14	98	0.11594	6.90E-07
	x_6	12	84	0.089634	4.18E-07	43	299	0.31157	0	14	98	0.12765	8.22E-07
	x_7	12	84	0.088747	7.45E-07	46	320	0.3351	0	15	105	0.1317	6.44E-07

Table 8: Test Results of the Three Algorithms on Problem 8

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
1000	x_1	8	72	0.013269	9.64E-08	17	119	0.005727	3.58E-07	6	42	0.002851	3.62E-07
	x_2	8	72	0.003109	7.01E-07	16	112	0.005234	7.69E-07	9	63	0.003082	1.09E-07
	x_3	7	63	0.002826	6.00E-07	13	91	0.00292	6.28E-07	5	35	0.001701	4.56E-07
	x_4	9	79	0.004187	3.23E-07	17	119	0.004392	7.57E-07	10	70	0.002768	2.54E-07
	x_5	9	79	0.003413	2.67E-07	18	126	0.00566	3.58E-07	8	56	0.00257	1.15E-07
	x_6	9	77	0.00287	7.86E-07	17	117	0.004641	7.42E-07	7	47	0.002457	9.03E-08
	x_7	8	72	0.002965	6.80E-07	16	112	0.00421	7.07E-07	13	95	0.004213	1.99E-07
5000	x_1	8	72	0.007738	2.16E-07	17	119	0.011569	8.00E-07	8	56	0.00692	4.48E-07
	x_2	9	81	0.007682	2.26E-07	17	119	0.011658	5.93E-07	9	63	0.007491	2.44E-07
	x_3	7	63	0.006689	5.49E-07	14	98	0.010785	4.84E-07	6	42	0.005109	1.53E-08
	x_4	9	79	0.010404	7.23E-07	18	126	0.012728	5.84E-07	10	70	0.009277	5.68E-07
	x_5	9	79	0.00863	1.63E-07	18	126	0.010443	8.00E-07	8	56	0.007101	2.53E-07
	x_6	9	77	0.007731	9.24E-07	18	124	0.011701	5.72E-07	7	47	0.006656	3.66E-07
	x_7	9	81	0.008617	2.59E-07	17	119	0.012214	5.36E-07	13	99	0.013587	8.03E-07
10000	x_1	8	72	0.010842	3.05E-07	18	126	0.017559	3.90E-07	8	56	0.011068	5.77E-07
	x_2	9	81	0.015059	3.20E-07	17	119	0.018149	8.38E-07	9	63	0.012734	3.31E-07
	x_3	7	63	0.011419	7.76E-07	14	98	0.014294	6.84E-07	6	42	0.008788	2.16E-08
	x_4	10	88	0.015191	2.09E-07	18	126	0.020685	8.25E-07	10	70	0.013833	8.04E-07
	x_5	9	79	0.012595	2.31E-07	19	133	0.021481	3.90E-07	8	56	0.011252	4.25E-07
	x_6	10	86	0.014451	3.36E-07	18	124	0.01883	8.08E-07	7	47	0.009595	6.79E-07
	x_7	9	81	0.01619	3.60E-07	17	119	0.020666	7.57E-07	22	174	0.041435	6.34E-08
50000	x_1	8	72	0.050495	6.82E-07	18	126	0.094878	8.72E-07	9	63	0.057807	1.20E-07
	x_2	9	81	0.062791	5.78E-09	18	126	0.094306	6.46E-07	9	63	0.058119	3.29E-07
	x_3	8	72	0.052236	3.48E-07	15	105	0.079557	5.27E-07	6	42	0.038082	4.84E-08
	x_4	10	88	0.057097	4.67E-07	19	133	0.090992	6.36E-07	11	77	0.075323	4.02E-07
	x_5	9	79	0.060498	5.16E-07	19	133	0.087837	8.72E-07	9	61	0.056936	5.34E-07
	x_6	10	86	0.065512	7.52E-07	19	131	0.10573	6.23E-07	7	47	0.043321	2.21E-07
	x_7	9	81	0.066168	3.06E-07	18	126	0.11584	5.84E-07	26	190	0.21195	5.12E-07

Table 9: Test Results of the Three Algorithms on Problem 9

		IDPLA				CPDY				FIDFP			
DIM	IP	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$	NIT	NFV	CPUT	$\|F(x^*)\|$
1000	x_1	0	0	0.005349	0	0	0	0.001356	0	0	0	0.001615	0
	x_2	34	544	0.047304	7.85E-07	82	812	0.055977	8.91E-07	32	466	0.030071	1.50E-07
	x_3	33	531	0.043246	9.47E-07	72	718	0.045436	9.70E-07	23	319	0.022449	2.11E-07
	x_4	12	204	0.014906	3.11E-07	52	528	0.035805	9.27E-07	8	254	0.01866	NaN
	x_5	37	595	0.037444	9.16E-07	56	566	0.036542	9.27E-07	27	393	0.024904	2.03E-07
	x_6	26	426	0.038468	7.17E-07	107	1045	0.077601	9.24E-07	48	728	0.04569	3.16E-07
	x_7	16	268	0.016758	4.66E-07	53	543	0.044623	8.12E-07	218	2850	0.19436	9.89E-07
5000	x_1	0	0	0.000837	0	0	0	0.001135	0	0	0	0.00082	0
	x_2	33	529	0.11353	6.37E-07	57	585	0.13451	8.14E-07				
	x_3	33	529	0.12682	4.60E-07	64	644	0.15649	8.47E-07	49	745	0.19021	2.11E-07
	x_4	12	204	0.046581	3.28E-07	43	445	0.1224	7.99E-07	30	434	0.10899	9.05E-07
	x_5	34	548	0.12036	6.92E-07	47	481	0.1231	9.79E-07	38	564	0.13866	6.69E-07
	x_6	23	377	0.079967	4.61E-07	97	953	0.23218	8.30E-07	74	1180	0.28688	2.67E-07
	x_7	17	285	0.078292	3.03E-07	54	554	0.13372	8.13E-07	248	3232	0.89299	9.55E-07
10000	x_1	0	0	0.001168	0	0	0	0.001045	0	0	0	0.001428	0
	x_2	33	529	0.25076	6.10E-07	60	610	0.2748	9.55E-07				
	x_3	30	484	0.2497	8.28E-07	62	626	0.30386	8.76E-07	53	797	0.39372	8.20E-07
	x_4	12	204	0.099238	2.97E-07	40	418	0.20217	8.43E-07	54	840	0.44177	8.42E-07
	x_5	33	531	0.25589	4.65E-07	44	454	0.22736	8.69E-07	48	766	0.38045	3.06E-07
	x_6	22	362	0.17403	8.15E-07	90	888	0.42968	9.77E-07	6	72	0.044187	NaN
	x_7	17	285	0.13244	4.30E-07	70	700	0.35722	9.87E-07				
50000	x_1	0	0	0.003275	0	0	0	0.003316	0	0	0	0.002865	0
	x_2	33	529	1.1151	8.37E-07	52	536	1.2117	8.36E-07				
	x_3	30	484	1.0389	5.80E-07	42	442	0.99242	9.14E-07				
	x_4	11	187	0.40756	9.41E-07	30	330	0.71105	7.85E-07	6	142	0.33785	NaN
	x_5	31	501	1.0699	6.49E-07	30	330	0.77068	9.27E-07				
	x_6	21	347	0.72852	5.71E-07	84	832	1.8821	8.59E-07				
	x_7	18	302	0.64947	2.66E-07	35	381	0.85757	9.84E-07				

Appendix B

The initial points used for this experiments are
$x_1 = (1, 1, ..., 1)^T$, $x_2 = (0.2, 0.2, ..., 0.2)^T$, $x_3 = (0.5, 0.5, ..., 0.5)^T$, $x_4 = (1.2, 1.2, ..., 1.2)^T$, $x_5 = (1.5, 1.5, ..., 1.5)^T$, $x_6 = (2, 2, ..., 2)^T$, $x_7 = rand(n, 1)$.