J. Nonlinear Var. Anal. 7 (2023), No. 4, pp. 627-646
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.7.2023.4.10

DOUBLE INERTIAL PARAMETERS FORWARD-BACKWARD SPLITTING METHOD: APPLICATIONS TO COMPRESSED SENSING, IMAGE PROCESSING, AND SCAD PENALTY PROBLEMS

LATEEF OLAKUNLE JOLAOSO[1], YEKINI SHEHU[2,*], JEN-CHIH YAO[3,4], RENQI XU[2]

[1]*School of Mathematical Sciences, University of Southampton, S017 1BJ, United Kingdom*
[2]*School of Mathematical Sciences, Zhejiang Normal University, Jinhua 321004, China*
[3]*Research Center for Interneural Computing, China Medical University, Taichung 40402, Taiwan*
[4]*Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 804, Taiwan*

Abstract. In this paper, a forward-backward splitting algorithm with two inertial parameters (one non-negative and the other non-positive) extrapolation step is proposed for finding a zero point of the sum of maximal monotone and co-coercive operators in real Hilbert spaces. One of the interesting features of our proposed algorithm is that no online rule on the inertial parameters with the iterates is needed. The weak convergence result of the proposed algorithm is established under some standard assumptions. Numerical results arising from LASSO problems in compressed sensing, image processing, and SCAD penalty problems are provided to illustrate the behavior of our proposed algorithm.

Keywords. Compressed sensing; Forward-backward splitting; Image processing; Two-step inertial; Weak convergence.

1. INTRODUCTION

Let H be a real Hilbert space with norm $\|\cdot\|$ and inner product $\langle\cdot,\cdot\rangle$. The problem of finding a zero of the sum of two monotone operators, i.e.,

$$\text{Find} \quad x \in H \quad \text{such that} \quad 0 \in (Ax + Bx), \tag{1.1}$$

where $A : H \to H$ is the co-coercive operator and $B : H \to 2^H$ is a maximal monotone operator, plays a vital role in the study of optimization theory. It has found various real applications in some concrete problems, such as image processing, machine learning, linear inverse problem. Due to its importance, several authors investigated appropriate algorithms for seeking its solutions. One of the well-known algorithms for solving Problem (1.1) is the *forward-backward splitting algorithm* which was introduced by Passty [23] as follows: for $x^0 \in H$,

$$x^{k+1} = (I + \lambda B)^{-1}(I - \lambda A)x^k, \quad k \geq 1, \tag{1.2}$$

where $\lambda > 0$. In each step of the iterates, the forward step involves A and the backward step involves B. This algorithm includes in special cases, other important iterative processes for

*Corresponding author.
E-mail addresses: l.o.jolaoso@soton.ac.uk (L.O. Jolaoso), yekini.shehu@zjnu.edu.cn (Y. Shehu), yaojc@mail.cmu.edu.tw (J.C. Yao), xurenqi@zjnu.edu.cn (R. Xu).
Received January 2, 2023; Accepted March 30, 2023.

solving optimization problems, such as the gradient descent algorithm and the proximal point algorithm. It is well known that the sequence generated by (1.2) converges weakly to some solution of Problem (1.1). Moreover, several modifications of the forward-backward splitting algorithm have been introduced for solving Problem (1.1); see, e.g., [11, 12, 13, 20, 29, 30, 31].

On the other hand, an inertial proximal point algorithm for finding the zero point of a maximal monotone operator was introduced by Alvarez and Attouch [1] in 2001 as follows: for any $x^{k-1}, x^k \in H$ and parameter $\theta_k \in [0,1)$, $\lambda_k > 0$,

$$\text{find} \quad x^{k+1} \in H : 0 \in \lambda_k B(x^{k-1}) + x^{k-1} - x^k - \theta_k(x^k - x^{k-1}), \quad \forall k \geq 1,$$

which is equivalent to

$$x^{k+1} = J^B_{\lambda_k}(x^k + \theta_k(x^k - x^{k-1})), \quad k \geq 1,$$

where $J^B_{\lambda_k} = (I + \lambda_k B)^{-1}$, B is a maximal monotone operator, and the inertial is induced by the term $\theta_k(x^k - x^{k-1})$. The inertial extrapolation term was first introduced by Polyak [25] as a heavy ball method and was later employed as an inertial process by Nesterov [22] for solving minimization problems. Recently, numerous authors incorporated the inertial extrapolation step to accelerate the convergence properties of various algorithms; see, e.g., [4, 24, 33] to mention but a few. It turns out that the introduction of term θ_k and two iterates x^{k-1}, x^k considerably improves the speed of convergence for the inertial proximal point algorithm. This can be explained from the fact that vector $x^k - x^{k-1}$ acts as an impulsion term and θ_k acts as a speed regulator. Thus, the inertial extrapolation step can be regarded as a procedure for speeding up the convergence properties of other associated algorithms in the literature.

Motivation and Innovation. Poon and Liang [27, 28] pointed out some limitations of some optimization algorithms with one-step inertial extrapolation step $x^k + \theta_k(x^k - x^{k-1})$, like the Douglas-Rachford splitting algorithm and ADMM with one-step inertial extrapolation step, using this example on feasibility problem.

Example 1.1. Let $T_1, T_2 \subset \mathbb{R}^2$ be two subspaces such that $T_1 \cap T_2 \neq \emptyset$. Find $x \in \mathbb{R}^2$ such that $x \in T_1 \cap T_2$.

It was demonstrated in [28, Section 4] that two-step inertial Douglas-Rachford splitting algorithm

$$x^{k+1} = F_{DR}(x^k + \theta(x^k - x^{k-1}) + \delta(x^{k-1} - x^{k-2}))$$

has faster convergence than the one-step inertial Douglas-Rachford splitting algorithm

$$x^{k+1} = F_{DR}(x^k + \theta(x + -x^{k-1}))$$

for Example 1.1. It was also revealed from Example 1.1 that the one-step inertial Douglas-Rachford splitting algorithm

$$x^{k+1} = F_{DR}(x^k + \theta(x^k - x^{k-1}))$$

converges slower than the Douglas-Rachford splitting algorithm

$$x^{k+1} = F_{DR}(x^k),$$

where

$$F_{DR} := \frac{1}{2}\Big(I + (2P_{T_1} - I)(2P_{T_2} - I)\Big)$$

is the Douglas-Rachford splitting operator. We can deduct from this example that one-step inertial Douglas-Rachford splitting algorithm may fail to provide acceleration and in certain cases, the use of inertia of more than two points x^k and x^{k-1} could speed up convergence (see, for example, [19, Chapter 4]). For example, the following two-step inertial extrapolation

$$w^k = x^k + \theta(x^k - x^{k-1}) + \delta(x^{k-1} - x^{k-2}) \tag{1.3}$$

with $\theta \geq 0$ and $\delta \leq 0$ can provide acceleration as discussed in [19, Chapter 4]. The failure of one-step inertial acceleration of ADMM was also discussed in [27, Section 3] and [21]. Polyak [26] also pointed out that the multi-step inertial algorithms can boost the speed of optimization algorithms even though neither the convergence nor the rate result of such multi-step inertial algorithms is established in [26].

In most of the forward-backward splitting algorithms with more than one inertial parameter extrapolation step existing in the literature, an online rule on the inertial parameters with the iterates (which is a summability condition of the inertial parameters and the sequence of iterates) has always been applied during numerical implementations of such algorithms (see, for example, [19, 15, 39] and the references therein).

It is based on this observation above that we propose a forward-backward splitting algorithm with two-step inertial extrapolation as given in (1.3) above with no online rule on the inertial parameters and the iterative sequence. Consequently, weak convergence result is obtained. Our results in this paper also serve as extensions of the forward-backward splitting algorithm with one-step inertial extrapolation considered in [1, 2, 3, 4, 24, 33, 12, 13, 29, 34, 36].

Summarily, our contributions in this paper are highlighted as follows:

- We introduce a splitting algorithm which is a forward-backward splitting algorithm with two-step inertial extrapolation and give a weak convergence analysis.
- We carefully design some computational experiments which demonstrate that our proposed algorithm is efficient and performs better than some related algorithms in the literature.

Organization. The paper is organized as follows: We first recall some basic definitions and results in Section 2. We present our main results in Section 3, which involves weak convergence results of our proposed Algorithm 1 with an application of our results to the proximal gradient algorithm. Numerical experiments are found in Section 4 and we conclude with some final remarks in Section 5, the last section.

2. Preliminaries

In this section, we give some definitions and basic results that are used in our subsequent analysis. The weak and the strong convergence of $\{x^k\} \subset H$ to $x \in H$ is denoted by $x^k \rightharpoonup x$ and $x^k \to x$ as $n \to \infty$ respectively. For any $x, y \in H$, it is known that

$$\|x+y\|^2 = \|x\|^2 + 2\langle x, y\rangle + \|y\|^2, \tag{2.1}$$

$$\|x+y\|^2 \leq \|x\|^2 + 2\langle y, x+y\rangle, \tag{2.2}$$

and

$$\|\alpha x + (1-\alpha)y\|^2 = \alpha\|x\|^2 + (1-\alpha)\|y\|^2 - \alpha(1-\alpha)\|x-y\|^2, \forall \alpha \in \mathbb{R}. \tag{2.3}$$

A mapping $T : H \to H$ is called

(i) nonexpansive if $\|Tx - Ty\| \leq \|x - y\|$ for all $x, y \in H$;

(ii) firmly nonexpansive if $\|Tx-Ty\|^2 \leq \|x-y\|^2 - \|(I-T)x-(I-T)y\|$ for all $x,y \in H$. Equivalently $\|Tx-Ty\|^2 \leq \langle x-y, Tx-Ty\rangle$ for all $x,y \in H$;

(iii) averaged if T can be expressed as the averaged of the identity mapping I and a nonexpansive mapping S, i.e., $T = (1-\beta)I + \beta S$ with $\beta \in (0,1)$. Alternatively, T is β-averaged if

$$\|Tx-Ty\|^2 \leq \|x-y\|^2 - \frac{1-\beta}{\beta}\|(I-T)x-(I-T)y\|^2, \forall x,y \in H, \beta \in (0,1).$$

Next, we state some classes of functions that play an essential role in our convergence analysis.

Definition 2.1. A mapping $A : H \to H$ is said to be

(i) L-Lipschitz continuous with Lipschitz constant $L > 0$ if

$$\|Ax - Ay\| \leq L\|x-y\| \quad \forall x,y \in H;$$

(ii) monotone if

$$\langle Ax - Ay, x-y\rangle \geq 0 \quad \forall x,y \in H;$$

(iii) η-strongly monotone if there exists a constant $\eta > 0$ such that

$$\langle Ax - Ay, x-y \geq \eta\|x-y\|^2 \quad \forall x,y \in H,$$

(iv) α co-coercive (or α-inverse strongly monotone) if there exists a constant $\alpha > 0$ such that

$$\langle Ax - Ay, x-y\rangle \geq \alpha\|Ax-Ay\|^2 \quad \forall x,y \in H.$$

Definition 2.2. A multivalued mapping $B : H \to 2^H$ is said to be monotone if, for any $x,y \in H$, $\langle x-y, f-g\rangle \geq 0$, where $f \in Bx$ and $g \in By$. The Graph of A is defined by

$$Gr(B) := \{(x,f) \in H \times H : f \in Ax\}.$$

When $Gr(B)$ is not properly contained in the graph of any other monotone mapping, we say that B is maximal. It is well-known that, for each $x \in H$, and $\lambda > 0$, there is a unique $z \in H$ such that $x \in (I + \lambda B)z$. The single-valued operator $J_\lambda^B(x)$ is called the resolvent of B (see [6]).

3. Main Results

In this section, we introduce a forward-backward splitting algorithm with two-step inertial extrapolation to find the zero point of the sum of maximal monotone and co-coercive operators in real Hilbert spaces. Consequently, we give weak convergence results for the sequence of iterates generated by the algorithm below.

In the sequel, we assume that the following conditions are satisfied:

Assumption 3.1. (i) $A : H \to H$ is α co-coercive operator and $B : H \to 2^H$ is maximal monotone operator;

(ii) The solution set $(A+B)^{-1}(0)$ of Problem (1.1) is nonempty;

Algorithm 1 Double Inertial Parameters Forward-Backward Splitting Method (DIPFBSM)

1: Choose $\theta \in [0,1), \delta \leq 0, \lambda \in (0,2\alpha), x^{-1}, x^0, x^1 \in H$ arbitrarily and set $k = 1$.
2: Compute
$$\begin{cases} w^k = x^k + \theta(x^k - x^{k-1}) + \delta(x^{k-1} - x^{k-2}) \\ x^{k+1} = J_\lambda^B(w^k - \lambda A w^k) \end{cases}$$
3: Set $k \leftarrow k+1$ and **goto 2**.

Let us assume the following conditions on the inertial factors θ and δ.

Assumption 3.2. (i) $0 \leq \theta < \min\{\frac{1}{3}, \frac{1-\beta}{1+\beta}\},\ \beta := \frac{2\alpha}{4\alpha - \lambda}$;
(ii) $\delta \leq 0$ such that
$$\max\Big\{ -\frac{(1-\beta-\theta-\beta\theta)}{1-\beta}, \frac{\beta\theta(1+\theta)-(1-\beta)(1-\theta)^2}{1+\theta}\Big\} < \delta;$$
and
$$\beta\theta(1+\theta) - (1-\beta)(1-\theta)^2 < (2\theta - \beta + 2)\delta + (1-2\beta)\delta^2.$$

Lemma 3.1. *The sequence $\{x^k\}$ generated by Algorithm 1 is bounded when both Assumption 3.1 and Assumption 3.2 are satisfied with $\lambda \in (0,2\alpha)$.*

Proof. Observe that by [8, Theorem 7], we have that $J_\lambda^B(I-\lambda A)$ is $\frac{2\alpha}{4\alpha-\lambda}-$ averaged. Thus, our proposed Algorithm 1 can be converted to a fixed point iteration of the form:
$$\begin{cases} w^k = x^k + \theta(x^k - x^{k-1}) + \delta(x^{k-1} - x^{k-2}) \\ x^{k+1} = Tw^k, \end{cases} \tag{3.1}$$
where $T := J_\lambda^B(I-\lambda A)$. Let $x^* \in F(T) = (A+B)^{-1}(0)$. Then
$$\begin{aligned} w^k &= x^k + \theta(x^k - x^{k-1}) + \delta(x^{k-1} - x_{k-2}) - x^* \\ &= (1+\theta)(x^k - x^*) - (\theta - \delta)(x^{k-1} - x^*) - \delta(x^{k-2} - x^*). \end{aligned}$$
Consequently, we have
$$\begin{aligned} \|w^k - x^*\|^2 &= \|(1+\theta)(x^k - x^*) - (\theta-\delta)(x^{k-1} - x^*) - \delta(x^{k-2} - x^*)\|^2 \\ &= (1+\theta)\|x^k - x^*\|^2 - (\theta-\delta)\|x^{k-1} - x^*\|^2 - \delta\|x^{k-2} - x^*\|^2 \\ &\quad +(1+\theta)(\theta-\delta)\|x^k - x^{k-1}\|^2 + \delta(1-\theta)\|x^k - x^{k-2}\|^2 \\ &\quad -\delta(\theta-\delta)\|x^{k-1} - x^{k-2}\|^2. \end{aligned} \tag{3.2}$$
Observe that
$$\begin{aligned} 2\theta\langle x^{k+1} - x^k, x^k - x^{k-1}\rangle &= 2\langle \theta(x^{k+1} - x^k), x^k - x^{k-1}\rangle \\ &\leq 2|\theta|\|x^{k+1} - x^k\|\|x^k - x^{k-1}\| \\ &= 2\theta\|x^{k+1} - x^k\|\|x^k - x^{k-1}\| \end{aligned}$$
and so
$$-2\theta\langle x^{k+1} - x^k, x^k - x^{k-1}\rangle \geq -2\theta\|x^{k+1} - x^k\|\|x^k - x^{k-1}\|. \tag{3.3}$$

Also,

$$\begin{aligned} 2\delta\langle x^{k+1}-x^k, x^{k-1}-x^{k-2}\rangle &= 2\langle \delta(x^{k+1}-x^k), x^{k-1}-x^{k-2}\rangle \\ &\leq 2|\delta|\|x^{k+1}-x^k\|\|x^{k-1}-x^{k-2}\| \end{aligned}$$

which implies that

$$-2\delta\langle x^{k+1}-x^k, x^{k-1}-x^{k-2}\rangle \geq -2|\delta|\|x^{k+1}-x^k\|\|x^{k-1}-x^{k-2}\|. \tag{3.4}$$

Similarly, we note that

$$\begin{aligned} 2\delta\theta\langle x^{k-1}-x^k, x^{k-1}-x^{k-2}\rangle &= 2\langle \delta\theta(x^{k-1}-x^k), x^{k-1}-x^{k-2}\rangle \\ &\leq 2|\delta|\theta\|x^{k-1}-x^k\|\|x^{k-1}-x^{k-2}\| \\ &= 2|\delta|\theta\|x^k-x^{k-1}\|\|x^{k-1}-x^{k-2}\| \end{aligned}$$

and thus,

$$\begin{aligned} 2\delta\theta\langle x^k-x^{k-1}, x^{k-1}-x^{k-2}\rangle &= -2\delta\theta\langle x^{k-1}-x^k, x^{k-1}-x^{k-2}\rangle \\ &\geq -2|\delta|\theta\|x^k-x^{k-1}\|\|x^{k-1}-x^{k-2}\|. \end{aligned} \tag{3.5}$$

By (3.3), (3.4), and (3.5), we obtain

$$\begin{aligned} \|x^{k+1}-w^k\|^2 &= \|x^{k+1}-(x^k+\theta(x^k-x^{k-1})+\delta(x^{k-1}-x^{k-2}))\|^2 \\ &= \|x^{k+1}-x^k-\theta(x^k-x^{k-1})-\delta(x^{k-1}-x^{k-2})\|^2 \\ &= \|x^{k+1}-x^k\|^2-2\theta\langle x^{k+1}-x^k, x^k-x^{k-1}\rangle \\ &\quad -2\delta\langle x^{k+1}-x^k, x^{k-1}-x^{k-2}\rangle+\theta^2\|x^k-x^{k-1}\|^2 \\ &\quad +2\delta\theta\langle x^k-x^{k-1}, x^{k-1}-x^{k-2}\rangle+\delta^2\|x^{k-1}-x^{k-2}\|^2 \\ &\geq \|x^{k+1}-x^k\|^2-2\theta\|x^{k+1}-x^k\|\|x^k-x^{k-1}\| \\ &\quad -2|\delta|\|x^{k+1}-x^k\|\|x^{k-1}-x^{k-2}\|+\theta^2\|x^k-x^{k-1}\|^2 \\ &\quad -2|\delta|\theta\|x^k-x^{k-1}\|\|x^{k-1}-x^{k-2}\|+\delta^2\|x^{k-1}-x^{k-2}\|^2 \\ &\geq \|x^{k+1}-x^k\|^2-\theta\|x^{k+1}-x^k\|^2-\theta\|x^k-x^{k-1}\|^2 \\ &\quad -|\delta|\|x^{k+1}-x^k\|^2-|\delta|\|x^{k-1}-x^{k-2}\|^2+\theta^2\|x^k-x^{k-1}\|^2 \\ &\quad -|\delta|\theta\|x^k-x^{k-1}\|^2-|\delta|\theta\|x^{k-1}-x^{k-2}\|^2+\delta^2\|x^{k-1}-x^{k-2}\|^2 \\ &= (1-|\delta|-\theta)\|x^{k+1}-x^k\|^2+(\theta^2-\theta-|\delta|\theta)\|x^k-x^{k-1}\|^2 \\ &\quad +(\delta^2-|\delta|-|\delta|\theta)\|x^{k-1}-x^{k-2}\|^2. \end{aligned} \tag{3.6}$$

Since T is β-averaged quasi-nonexpansive, we obtain

$$\begin{aligned} \|x^{k+1}-x^*\|^2 &= \|Tw^k-x^*\|^2 \\ &\leq \|w^k-x^*\|^2-\frac{(1-\beta)}{\beta}\|w^k-Tw^k\|^2. \end{aligned} \tag{3.7}$$

Using (3.2) and (3.6) in (3.7), we obtain

$$\begin{aligned}
\|x^{k+1}-x^*\|^2 &\leq (1+\theta)\|x^k-x^*\|^2-(\theta-\delta)\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2\\
&\quad +(1+\theta)(\theta-\delta)\|x^k-x^{k-1}\|^2+\delta(1+\theta)\|x^k-x^{k-2}\|^2\\
&\quad -\delta(\theta-\delta)\|x^{k-1}-x^{k-2}\|^2-\frac{(1-\beta)}{\beta}(1-|\delta|-\theta)\|x^{k+1}-x^k\|^2\\
&\quad -\frac{(1-\beta)}{\beta}(\theta^2-\theta-|\delta|\theta)\|x^k-x^{k-1}\|^2\\
&\quad -\frac{(1-\beta)}{\beta}(\delta^2-|\delta|-|\delta|\theta)\|x^{k-1}-x^{k-2}\|^2\\
&= (1+\theta)\|x^k-x^*\|^2-(\theta-\delta)\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2\\
&\quad +\left[(1+\theta)(\theta-\delta)-\left(\frac{1-\beta}{\beta}\right)(\theta^2-\theta-|\delta|\theta)\right]\|x^k-x^{k-1}\|^2\\
&\quad +\delta(1+\theta)\|x^k-x^{k-2}\|^2-\left(\frac{1-\beta}{\beta}\right)(1-|\delta|-\theta)\|x^{k+1}-x^k\|^2\\
&\quad -\left[\delta(\theta-\delta)+\left(\frac{1-\beta}{\beta}\right)(\delta^2-|\delta|-|\delta|\theta)\right]\|x^{k-1}-x^{k-2}\|^2\\
&\leq (1+\theta)\|x^k-x^*\|^2-(\theta-\delta)\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2\\
&\quad +\left[(1+\theta)(\theta-\delta)-\left(\frac{1-\beta}{\beta}\right)(\theta^2-\theta-|\delta|\theta)\right]\|x^k-x^{k-1}\|^2\\
&\quad -\left(\frac{1-\beta}{\beta}\right)(1-|\delta|-\theta)\|x^{k+1}-x^k\|^2\\
&\quad -\left[\delta(\theta-\delta)+\left(\frac{1-\beta}{\beta}\right)(\delta^2-|\delta|-|\delta|\theta)\right]\|x^{k-1}-x^{k-2}\|^2
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\|x^{k+1}-x^*\|^2-\theta\|x^k-x^*\|^2-\delta\|x^{k-1}-x^*\|^2\\
&+\left(\frac{1-\beta}{\beta}\right)(1-|\delta|-\theta)\|x^{k+1}-x^k\|^2\\
&\leq \|x^k-x^*\|^2-\theta\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2\\
&+\left(\frac{1-\beta}{\beta}\right)(1-|\delta|-\theta)\|x^k-x^{k-1}\|^2\\
&+\left((\theta-\delta)(1+\theta)-\left(\frac{1-\beta}{\beta}\right)(\theta^2-2\theta-|\delta|\theta-|\delta|+1)\right)\|x^k-x^{k-1}\|^2\\
&-\left[\delta(\theta-\delta)+\left(\frac{1-\beta}{\beta}\right)(\delta^2-|\delta|-|\delta|\theta)\right]\|x^{k-1}-x^{k-2}\|^2. \qquad (3.8)
\end{aligned}$$

For each $k\geq 1$, define

$$\begin{aligned}
\Gamma_k := &\|x^k-x^*\|^2-\theta\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2\\
&+\left(\frac{1-\beta}{\beta}\right)(1-|\delta|-\theta)\|x^k-x^{k-1}\|^2.
\end{aligned}$$

We first show that $\Gamma_k \geq 0, \ \forall k \geq 1$. Note that

$$\|x^{k-1}-x^*\|^2 \leq 2\|x^k-x^{k-1}\|^2+2\|x^k-x^*\|^2.$$

Hence,

$$\begin{aligned}
\Gamma_k &= \|x^k-x^*\|^2-\theta\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2 \\
&\quad +\left(\frac{1-\beta}{\beta}\right)(1-|\delta|-\theta)\|x^k-x^{k-1}\|^2 \\
&\geq \|x^k-x^*\|^2-2\theta\|x^k-x^{k-1}\|^2-2\theta\|x^k-x^*\|^2 \\
&\quad -\delta\|x^{k-2}-x^*\|^2+\left(\frac{1-\beta}{\beta}\right)(1-|\delta|-\theta)\|x^k-x^{k-1}\|^2 \\
&= (1-2\theta)\|x^k-x^*\|^2+\left[\left(\frac{1-\beta}{\beta}\right)(1-|\delta|-\theta)-2\theta\right]\|x^k-x^{k-1}\|^2 \\
&\quad -\delta\|x^{k-2}-x^*\|^2. \qquad (3.9)
\end{aligned}$$

By Assumption 3.2 (i) and (ii), we obtain

$$|\delta| < 1-\theta-\frac{2\theta}{\left(\frac{1-\beta}{\beta}\right)} = \frac{1-\beta-\theta-\beta\theta}{1-\beta}.$$

We then obtain from (3.9) that $\Gamma_k \geq 0$ for all $k \geq 0$ since

$$-\frac{(1-\beta-\theta-\beta\theta)}{1-\beta} < \delta$$

and $\theta < \frac{1}{3}$ from Assumption 3.2 (i) and (ii). Consequently, we obtain from (3.8) that

$$\begin{aligned}
\Gamma_{k+1}-\Gamma_k &\leq \left((\theta-\delta)(1+\theta)-\left(\frac{1-\beta}{\beta}\right)(\theta^2-2\theta-|\delta|\theta-|\delta|+1)\right)\|x^k-x^{k-1}\|^2 \\
&\quad -\left[\delta(\theta-\delta)+\left(\frac{1-\beta}{\beta}\right)(\delta^2-|\delta|-|\delta|\theta)\right]\|x^{k-1}-x^{k-2}\|^2 \\
&\quad -\left((\theta-\delta)(1+\theta)-\left(\frac{1-\beta}{\beta}\right)(\theta^2-2\theta-|\delta|\theta-|\delta|+1)\right)(\|x^{k-1}-x^{k-2}\|^2 \\
&\quad -\|x^k-x^{k-1}\|^2)+((\theta-\delta)(1+\theta)-\left(\frac{1-\beta}{\beta}\right)(\theta^2-2\theta-|\delta|\theta-|\delta|+1) \\
&\quad -\delta(\theta-\delta)-\left(\frac{1-\beta}{\beta}\right)(\delta^2-|\delta|-|\delta|\theta))\|x^{k-1}-x^{k-2}\|^2 \\
&= c_1\left(\|x^{k-1}-x^{k-2}\|^2-\|x^k-x^{k-1}\|^2\right)-c_2\|x^{k-1}-x^{k-2}\|^2, \qquad (3.10)
\end{aligned}$$

where

$$c_1 := -\left((\theta-\delta)(1+\theta)-\left(\frac{1-\beta}{\beta}\right)(\theta^2-2\theta-|\delta|\theta-|\delta|+1)\right)$$

and

$$c_2 := -\Big((\theta-\delta)(1+\theta) - \left(\frac{1-\beta}{\beta}\right)(\theta^2 - 2\theta - |\delta|\theta - |\delta| + 1)$$
$$-\delta(\theta-\delta) - \left(\frac{1-\beta}{\beta}\right)(\delta^2 - |\delta| - |\delta|\theta)\Big).$$

In view of $|\delta| = -\delta$, we have that

$$c_1 = -\left((\theta-\delta)(1+\theta) - \left(\frac{1-\beta}{\beta}\right)(\theta^2 - 2\theta - |\delta|\theta - |\delta| + 1)\right) > 0$$

which is equivalent to

$$\frac{\theta(1+\theta) - \left(\frac{1-\beta}{\beta}\right)(1-\theta)^2}{(1+\theta)\left(1+\frac{1-\beta}{\beta}\right)} < \delta. \tag{3.11}$$

By Assumption 3.2 (ii), we see that (3.11) holds and thus $c_1 > 0$. Also,

$$c_2 := -\Big((\theta-\delta)(1+\theta) - \left(\frac{1-\beta}{\beta}\right)(\theta^2 - 2\theta - |\delta|\theta - |\delta| + 1)$$
$$-\delta(\theta-\delta) - \left(\frac{1-\beta}{\beta}\right)(\delta^2 - |\delta| - |\delta|\theta)\Big) > 0 \tag{3.12}$$

implies that

$$\theta(1+\theta) - \left(\frac{1-\beta}{\beta}\right)(1-\theta)^2 < \left(1+\frac{1-\beta}{\beta}\right)\delta(1+\theta)$$
$$+\delta(\theta-\delta) + \left(\frac{1-\beta}{\beta}\right)(\delta^2 + \delta(1+\theta)). \tag{3.13}$$

By Assumption 3.2 (ii), we have that inequality (3.13) is satisfied. Therefore, $c_2 > 0$ from (3.12). From (3.10), we obtain

$$\begin{aligned}\Gamma_{k+1} + c_1\|x^k - x^{k-1}\|^2 \;\; \leq \;\; & \Gamma_k + c_1\|x^{k-1} - x^{k-2}\|^2 \\ & -c_2\|x^{k-1} - x^{k-2}\|^2.\end{aligned} \tag{3.14}$$

Letting $\bar{\Gamma}_k := \Gamma_k + c_1\|x^{k-1} - x^{k-2}\|^2$, we obtain from (3.14) that $\bar{\Gamma}_{k+1} \leq \bar{\Gamma}_k$, which implies that sequence $\{\bar{\Gamma}_k\}$ is decreasing and thus $\lim\limits_{k\to\infty}\bar{\Gamma}_k$ exists. Consequently, we have from (3.14) that $\lim\limits_{k\to\infty} c_2\|x^{k-1} - x^{k-2}\|^2 = 0$. Hence,

$$\lim_{k\to\infty}\|x^{k-1} - x^{k-2}\| = 0. \tag{3.15}$$

Using (3.15) and existence of limit of $\{\bar{\Gamma}_k\}$, we have that

$$\begin{aligned}\lim_{k\to\infty}\Gamma_k \;\; := \;\; & \lim_{k\to\infty}\Big[\|x^k - x^*\|^2 - \theta\|x^{k-1} - x^*\|^2 - \delta\|x^{k-2} - x^*\|^2 \\ & + \left(\frac{1-\beta}{\beta}\right)(1 - |\delta| - \theta)\|x^k - x^{k-1}\|^2\Big]\end{aligned} \tag{3.16}$$

exists. Also,

$$\begin{aligned}\|x^{k+1}-w^k\| &= \|x^{k+1}-x^k-\theta(x^k-x^{k-1})-\delta(x^{k-1}-x^{k-2})\| \\ &\leq \|x^{k+1}-x^k\|+\theta\|x^k-x^{k-1}\|+|\delta|\|x^{k-1}-x^{k-2}\| \to 0,\ n\to\infty.\end{aligned}$$

So, we obtain $\lim_{k\to\infty}\|w^k-Tw^k\|=0$. Again, Note that

$$\|w^k-x^k\| \leq \theta\|x^k-x^{k-1}\|+|\delta|\|x^{k-1}-x^{k-2}\| \to 0,\ n\to\infty.$$

Since $\lim_{k\to\infty}\Gamma_k$ exists and $\lim_{n\to\infty}\|x^k-x^{k-1}\|=0$, we have from (3.9) that $\{x^k\}$ is bounded. □

Theorem 3.1. *Let $\lambda\in(0,2\alpha)$ with both Assumption 3.1 and Assumption 3.2 fulfilled. Then $\{x^k\}$ generated by Algorithm 1 converges weakly to a point in $(A+B)^{-1}(0)$.*

Proof. (i) Using (3.15) in (3.16), we have that

$$\lim_{k\to\infty}\left[\|x^k-x^*\|^2-\theta\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2\right]$$

exists. By Lemma 3.1, we have that $\{x^k\}$ is bounded. We first show that any weak cluster point of $\{x^k\}$ is in $F(T)$, where T is as defined in (3.1). Suppose $\{x_{n_k}\}\subset\{x^k\}$ such that $x^{n_k}\rightharpoonup v^*\in H$. Since $\|w^k-x^k\|\to 0,\ k\to\infty$, we have $y_{n_k}\rightharpoonup v^*\in H$. By the result that $\|w^k-Tw^k\|\to 0,\ n\to\infty$, and by the demiclosedness of T, we have that $v^*\in F(T)=(A+B)^{-1}(0)$.

We now prove that $x^k\rightharpoonup x^*\in F(T)$. Let us assume that there exist $\{x^{k_n}\}\subset\{x^k\}$ and $\{x^{k_j}\}\subset\{x^k\}$ such that $x^{k_n}\rightharpoonup v^*, n\to\infty$ and $x^{k_j}\rightharpoonup x^*, j\to\infty$. We show that $v^*=x^*$. Observe that

$$2\langle x^k, x^*-v^*\rangle = \|x^k-v^*\|^2-\|x^k-x^*\|^2-\|v^*\|^2+\|x^*\|^2, \tag{3.17}$$

$$\begin{aligned}2\langle -\theta x^{k-1}, x^*-v^*\rangle &= -\theta\|x^{k-1}-v^*\|^2+\theta\|x^{k-1}-x^*\|^2 \\ &\quad +\theta\|v^*\|^2-\theta\|x^*\|^2\end{aligned} \tag{3.18}$$

and

$$\begin{aligned}2\langle -\delta x^{k-2}, x^*-v^*\rangle &= -\delta\|x^{k-2}-v^*\|^2+\delta\|x^{k-2}-x^*\|^2 \\ &\quad +\delta\|v^*\|^2-\delta\|x^*\|^2.\end{aligned} \tag{3.19}$$

Addition of (3.17), (3.18), and (3.19) gives

$$\begin{aligned}2\langle x^k-\theta x^{k-1}-\delta x^{k-2}, x^*-v^*\rangle &= \left(\|x^k-v^*\|^2-\theta\|x^{k-1}-v^*\|^2-\delta\|x^{k-2}-v^*\|^2\right) \\ &\quad -\left(\|x^k-x^*\|^2-\theta\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2\right) \\ &\quad +(1-\theta-\delta)(\|x^*\|^2-\|v^*\|^2).\end{aligned}$$

According to (3.16), we have

$$\lim_{k\to\infty}\left[\|x^k-x^*\|^2-\theta\|x^{k-1}-x^*\|^2-\delta\|x^{k-2}-x^*\|^2\right]$$

exists and

$$\lim_{k\to\infty}\left[\|x^k-v^*\|^2-\theta\|x^{k-1}-v^*\|^2-\delta\|x^{k-2}-v^*\|^2\right]$$

exists. This implies that

$$\lim_{k\to\infty}\langle x^k-\theta x^{k-1}-\delta x^{k-2}, x^*-v^*\rangle$$

exists. Now,

$$\begin{aligned}\langle v^* - \theta v^* - \delta v^*, x^* - v^* \rangle &= \lim_{n\to\infty} \langle x^{k_n} - \theta x^{k_n-1} - \delta x^{k_n-2}, x^* - v^* \rangle \\ &= \lim_{k\to\infty} \langle x^k - \theta x^{k-1} - \delta x^{k-2}, x^* - v^* \rangle \\ &= \lim_{j\to\infty} \langle x^{k_j} - \theta x^{k_j-1} - \delta x^{k_j-2}, x^* - v^* \rangle \\ &= \langle x^* - \theta x^* - \delta x^*, x^* - v^* \rangle,\end{aligned}$$

which yields

$$(1-\theta-\delta)\|x^* - v^*\|^2 = 0.$$

Since $\delta \le 0 < 1-\theta$, we obtain that $x^* = v^*$. Hence, $\{x^k\}$ converges weakly to a point in $F(T) = (A+B)^{-1}(0)$. □

Remark 3.1. (i) When $\theta = 0 = \delta$, Algorithm 1 reduces to the algorithms in [12, 13, 36]. When $\delta = 0$, Algorithm 1 reduces to the algorithms in [1, 2, 3, 4, 24, 33, 29].

(ii) The summability conditions of the inertial parameters and the sequence of iterates imposed in [15, Theorem 4.2 (35)], and [19, Chapter 4] are dispensed with in the convergence analysis of Algorithm 1.

(iii) Algorithm 1 is one of the few available splitting algorithms with two-step inertial extrapolations to solve inclusion problem (1.1).

(iv) [18, Algorithm 1] is a special case of Algorithm 1 when $A \equiv 0$.

Consider the convex minimization problem

$$\min_{x\in H} F(x) := f(x) + g(x), \tag{3.20}$$

where

(A) $f : H \to \mathbb{R}$ is convex, L-smooth function (for $L > 0$), i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x-y\| \quad \forall x, y \in H,$$

(B) $g : H \to \mathbb{R} \cup \{+\infty\}$ is convex and proper lower semicontinuous,

(C) the optimal points $x^* = \text{argmin}_x F(x) \neq \emptyset$.

A commonly used approach for solving Problem (3.20) is the proximal gradient algorithm which can be described as: for $x_0 \in H$, $x^{k+1} = \text{prox}_{\lambda g}(x^k - \lambda \nabla f(x^k))$ forall $k \ge 0$, where $\text{prox}_{\lambda g}(x) = \text{argmin}_{u\in C}\left\{g(x) + \frac{1}{\lambda}\|x-u\|^2\right\}$ and $\lambda > 0$ is a stepsize. It is well known that $prox_{\lambda g}$ is firmly nonexpansive and the set of fixed points of $\text{prox}_{\lambda g}$ coincides with the set of minimizer of g (see [6]). Moreover, since ∇f is L-Lipschitz continuous, it is $\frac{1}{L}$ co-coercive. Also, the subdifferential of g which is defined by

$$\partial g(x) = \{u \in H : \langle y-x, u\rangle + f(x) \le f(u), \quad \forall y \in H\}$$

is maximally monotone (see [6, Theorem 21.2]).

Algorithm 1 reduces to the following algorithm for solving convex minimization problem (3.20).

Algorithm 2 Double Inertial Parameters Forward-Backward Splitting Method (DIPFBSM)

1: Choose $\theta \in [0,1), \delta \leq 0, \lambda \in (0, \frac{2}{L}), x^{-1}, x^0, x^1 \in H$ arbitrarily and set $k = 1$.
2: Compute

$$\begin{cases} w^k = x^k + \theta(x^k - x^{k-1}) + \delta(x^{k-1} - x^{k-2}) \\ x^{k+1} = \text{prox}_{\lambda g}(w^k - \lambda \nabla f(w^k)) \end{cases}$$

3: Set $k \leftarrow k+1$ and **goto 2**.

Corollary 3.1. *Let $f : H \to \mathbb{R}$ be a convex and L-smooth function with $L > 0$ and $g : H \to \mathbb{R} \cup \{+\infty\}$ be a convex and proper lower semicontinuous. Suppose the solution set* $\text{Argmin}(F) \neq \emptyset$*. Let $\{x^k\}$ be a sequence generated by Algorithm 2. Assume that Assumption 3.2 is satisfied. Then sequence $\{x^k\}$ converges weakly to a point in* $\text{Argmin}(F)$.

4. Numerical Simulations

In this section, we implement Algorithm 1 on MATLAB and run some numerical experiments using Least Absolute Selection and Shrinkage Operator (LASSO) [35] and Smoothly Clipped Absolute Deviation Problem (SCAD) [16] penalty problem. We compare the performance of the proposed Algorithm 1 with other related algorithms which include FISTA in [7], FISTA-CD in [10], cGIGPM in [38], and PGM in [37] . All codes were run on a PC with specifications: Intel(R) Core (TM) i7-9700 CPU @ 3.00 GHz 3.00 GHz, 8.0 GB installed RAM with MATLAB R2019b 9.7.0.1190202.

4.1. **Application to LASSO Problem in Compressed Sensing.**

Example 4.1. Given matrix $A \in \mathbb{R}^{M \times N}$, a vector $b \in \mathbb{R}^M$, λ a positive scalar, the l_1-norm regularized least squares model can be expressed as

$$\min_{x \in \mathbb{R}^N} \left\{ \frac{1}{2} ||Ax - b||_2^2 + \lambda ||x||_1 \right\}. \tag{4.1}$$

The l_1 regularization is a popular concept and has gained a lot of popularization in different areas. For example, when the least-squares problem is posed with l_1 penalty, this is called LASSO, and Basis Pursuit Denoising [14].

Compressed sensing is very important when it comes to the problem of efficiently acquiring and reconstructing a signal. This signal processing technique has to do with solving underdetermined linear systems $Ax = b$ (linear equations where $N \gg M$). In this case, where the number of unknowns is greater than the number of equations, the linear system generates many solutions or could result in no solution. The approach to solving such a system is known as the linear least-squares algorithm (finding the minimum l_2-norm solution). The above l_1-norm regularized least squares model (4.1) can be computed to recover x when x is sparse which is the case in most applications. The model given in (4.1) is most often referred to as LASSO. Standard general algorithms such as an Interior Point Method (IPM), [20], can be used to solve the LASSO problem by reformulating the problem as a second-order cone programming. However, the computational complexity of such traditional methods is too high to handle large-scale data encountered in many real-life applications.

The LASSO problem is a special case of (3.20) where

$$f(x) = \frac{1}{2}\|Ax - b\|^2, \;\; g(x) = \lambda\|x\|_1.$$

Its gradient $\nabla f = A^*Ax - A^*b$ is Lipschitz continuous with Lipschitz constant $L(f) = \|A^*A\|$. The proximal map with $g(x) = \lambda||x||_1$ is given as

$$\text{prox}_g(x) = arg\min_u \lambda||x||_1 + \frac{1}{2}||u-x||_2^2,$$

which is separable in indices. Thus, for $x \in \mathbb{R}^N$,

$$\begin{aligned}\text{prox}_g(x) &= \text{prox}_{\lambda||.||_1}(x) \\ &= (\alpha_1, \cdots, \alpha_N),\end{aligned}$$

where

$$\alpha_k = \text{sgn}(x_k)\max\{|x_k| - \lambda, 0\} \;\text{ for }\; k = 1, 2, \ldots, N.$$

We begin by testing our proposed Algorithm 1 with a real signal that has been compressed with the ultimate goal of reconstructing a length-N sparse signal from M observation, with $M \ll N$. We cut the real signal given in [32] to the same length of the simulated one and use data from the Case Western Reserve University in [32], where a comprehensive study of the signals is done.

We test the Algorithm 1 (DIPFBSM) with FISTA in [7], FISTA-CD in [10] and cGIGPM in [38]. We choose $N = 2^{12}$ and $M = 2^{10}$ for the experiment. The vector b is randomly generated by the normal distribution with $SNR = 40$, A is generated via the normal distribution with mean zero and variance one, and $x \in \mathbb{R}^N$ is generated by a uniform distribution in [-2,2]. We set $L = \|A^TA\|$, and use the following parameters for the algorithms: we take $\theta = 0.49, \delta = -0.5, \lambda = 1.5L$ for Algorithm 1 (DIPFBSM); λ = 1/L for FISTA; $\lambda = 1/L, a = 3$ for FISTA-CD; and $\alpha = 0.35, \beta = 0.5L, \lambda_0 = 0.005, \lambda = \min\{\lambda_0, \min\{\alpha/\beta, (2-2\alpha-\varepsilon)/(1-\beta)\}/L\}$ for cGIGPM. We studied the efficiency of the algorithms using the following stopping criterion

$$MSE = \frac{1}{N}\|x^k - x^*\| < 10^{-3},$$

where x^k is an estimated signal of x^*. Figure 4.1 demonstrates the graphs of objective function value against the number of iterations and CPU time by each algorithm and Figure 4.2 contains the reconstructed signal by each algorithm.

From the numerical results in Figure 4.1 and Figure 4.2, we see that all the tested algorithms are able to reconstruct the noise signal efficiently. Moreover, Algorithm 1 (DIPFBSM) possesses the lowest execution time of other tested algorithms in this case. Also, Algorithm 1 (DIPFBSM) has lower MSE values than FISTA, FISTA-CD, and cGIGPM. Thus Algorithm 1 (DIPFBSM) is more efficient than FISTA, FISTA-CD, and cGIGPM in reconstructing the signal.

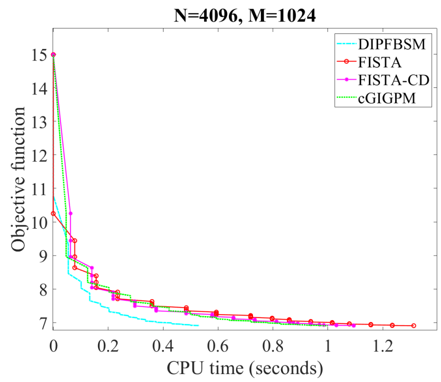

FIGURE 4.1. Example 4.1, graphs of objective function values against CPU time in seconds (left) for the signal reconstruction.

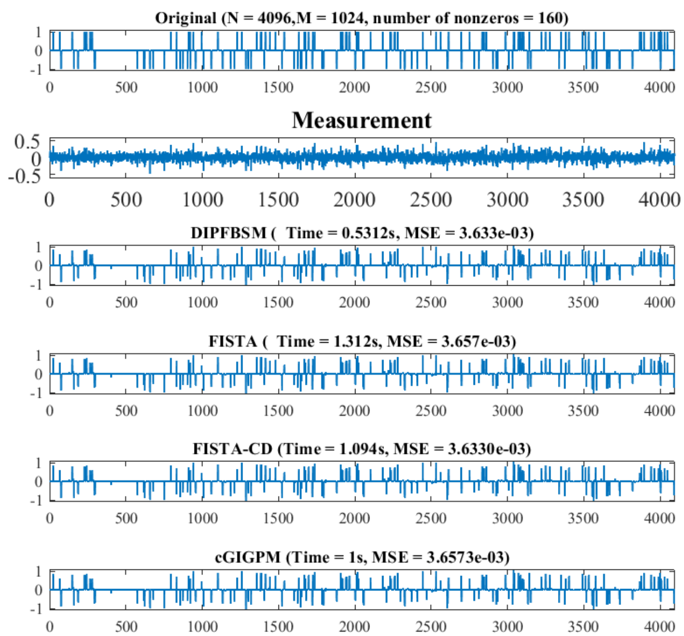

FIGURE 4.2. Example 4.1, From the top, Original signal, observed data, and recovered signal by DIPFBSM, FISTA, FISTA-CD, and cGIGPM when $N = 4096$ and $M = 1024$.

4.2. **Application to image processing.**

Example 4.2. Consider the application of Algorithm 1 to image restoration problem and compare its efficiency with FISTA in [7], FISTA-CD in [10], and cGIGPM in [38]. The image restoration problem is formulated by the following model $z = Ax + b$, where x is the original image, z is the degraded image, A is a blurring matrix, and b is the noise. One of the efficient algorithms for recovering the original image is the l_1-norm regularized least square model (4.1). In this case, A represents the blurring operator, x is the original image and b is the observed image. Our aim here is to recover the original image x given the data of the blurred image z. We consider the greyscale image of M pixels wide and N pixel height, each value is known to be in the range $[0,255]$. Let $D = M \times N$. The quality of the restored image is measured by the signal-to-noise ratio defined as

$$SNR = 20 \times \log_{10}\left(\frac{\|x\|_2}{\|x - x^*\|_2}\right),$$

where x is the original image and x^* is the restored image. Typically, the larger the SNR, the better the quality of the restored image and this leads to the curves in Figure 6. In our experiments, we used the grey test image Pout (291×240), Cameraman (256×256) and Tyre (205×232), while each test image is degraded by Gaussian 7×7 blur kernel with standard deviation 4. We also used similar parameters as in Example 4.1 for the test algorithms with the initial values taken as $x_0 = \mathbf{0} \in \mathbb{R}^D$ and $x_1 = \mathbf{1} \in \mathbb{R}^D$. Figure 4.3-4.5 show the restored images by the algorithms using the test images Pout.tif, cameraman.tif and tyre.tif. Figure 4.6 shows the graphs of SNR values against the number of iterations for each test image. In Table 1, we report the time (in seconds) and SNR values for each algorithm in the experiments.

From the computational results, we see that Algorithm 1 (DIPFBSM) performs more efficiently in terms of SNR values and CPU time for reconstructing the degraded images than other tested algorithms.

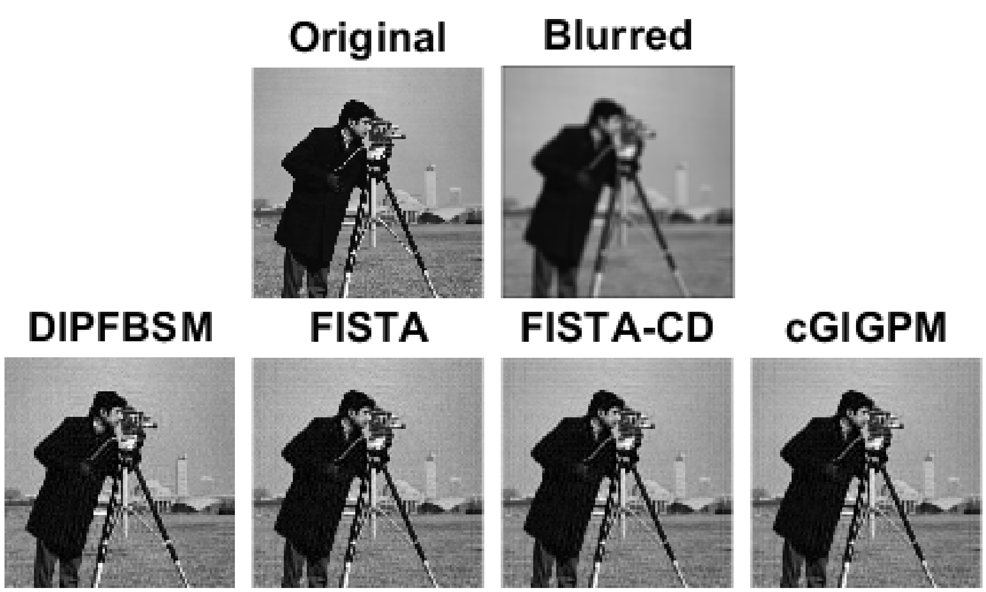

FIGURE 4.3. Example 4.2, Top demonstrates original image of Cameraman (left) and degraded image of Cameraman (right); Bottom shows recovered image by IRFBSM, FISTA, FISTA-CD and cGIGPM.

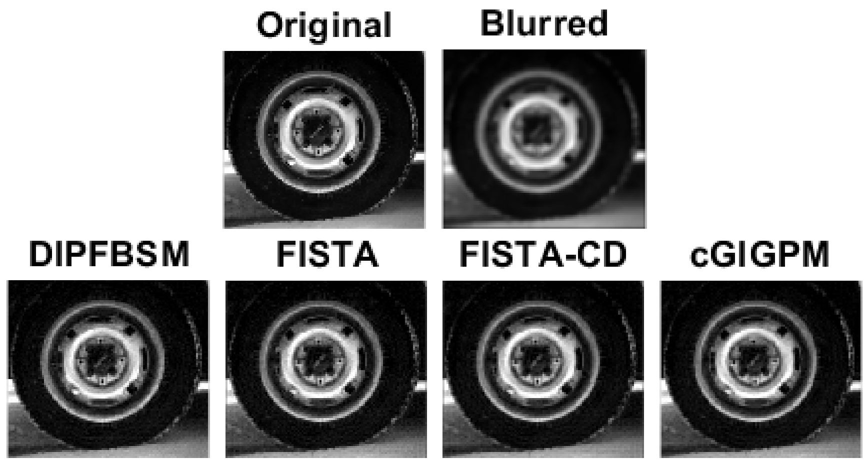

FIGURE 4.4. Example 4.2, Top shows original image of Tyre (left) and degraded image of Tyre (right); Bottom shows recovered image by DIPFBSM, FISTA, FISTA-CD and cGIGPM.

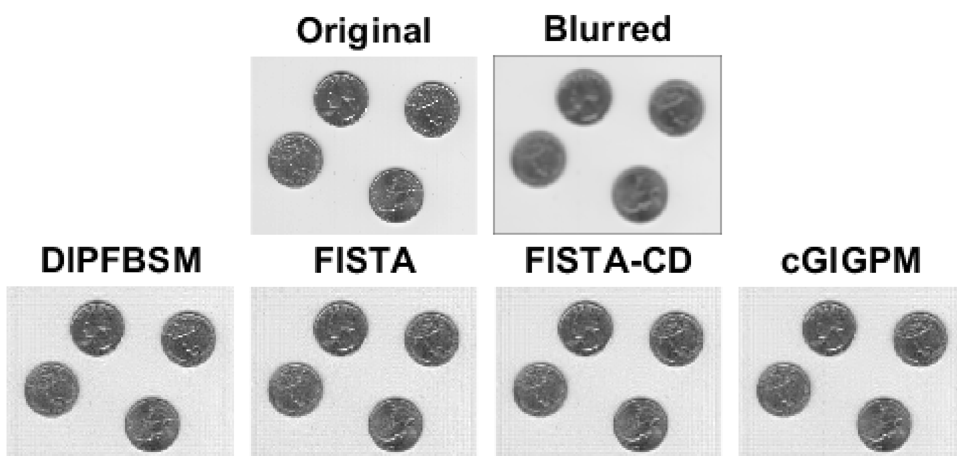

FIGURE 4.5. Example 4.2, Top shows original image of Coins (left) and degraded image of Coins (right); Bottom shows recovered image by DIPFBSM, FISTA, FISTA-CD and cGIGPM.

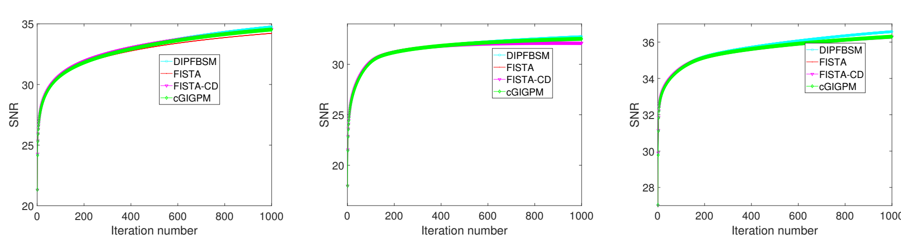

FIGURE 4.6. Example 4.2, Graphs of SNR value against number of iterations for Cameraman (left), Tyre (middle) and Coins (right) test images.

TABLE 1. Computational result for Example 4.2

Algorithms	Cameraman		Tyre		Coins	
	Time (secs)	SNR	Time (secs)	SNR	Time(secs)	SNR
DIPFBSM	20.6072	34.5685	9.2108	39.2236	11.9821	32.7439
FISTA	26.9805	34.2205	10.8231	38.1461	14.92222	32.1087
FISTA-CD	26.3214	34.5441	12.3832	37.5964	17.6809	32.2273
cGIGPM	26.1604	34.7533	12.6887	37.2201	17.7146	32.4618

4.3. **Application to SCAD penalty problem.** Next, we apply our result to solve the SCAD penalty problem arising in statistical learning.

Example 4.3. The SCAD penalty problem can be formulated as follows (see, e.g. [16]):

$$\min_{x\in\mathbb{R}^N} \frac{1}{2}\|Ax-b\|_2^2 + \sum_{i=1}^{N} g_\kappa(|x_i|), \tag{4.2}$$

where $A \in \mathbb{R}^{M\times N}$, $b \in \mathbb{R}^M$, and SCAD penalty function $g_\kappa(\cdot)$ is defined by

$$g_\kappa(\theta) = \begin{cases} \kappa|\theta| & \text{if} \quad |\theta| \le \kappa, \\ \dfrac{-\theta^2+2a\kappa|\theta|-\kappa^2}{2(a-1)}, & \text{if} \quad \kappa < |\theta| \le c\kappa, \\ \dfrac{(a+1)\kappa^2}{2}, & \text{if} \quad |\theta| > c\kappa, \end{cases} \tag{4.3}$$

where $a > 2$ and $\kappa > 0$ correspond to the knots of the quadratic spline functions.

Note that $g_\kappa(\cdot) + \frac{\omega}{2}|\cdot|^2$ is convex in $\mathbb{R}$ if $\omega \ge \frac{1}{a-1}$ (see, [17, Theorem 3.1]). Then, SCAD penalty problem (4.2) can be reformulated as the following:

$$\min_{x\in\mathbb{R}^N} F(x) := f(x) + g(x),$$

where

$$f(x) = \frac{1}{2}\|Ax-b\|_2^2 - \frac{1}{2(a-1)}\|x\|^2 \quad \text{and} \quad g(x) = \sum_{i=1}^{N} g_\kappa(|x_i|) + \frac{1}{2(a-1)}\|x\|^2.$$

In our experiment, we compare Algorithm 1 (DIPFBSM) with the FISTA in [7], FISTA-CD in [10], PGM in [37] and cGIPGM in [38]. The matrix A is normally distributed with zero mean and variance one, x is a random sparse vector with density 0.01 and b is the noise vector. Following [38], the parameters a and κ are chosen as $a = 3.7$ and $\kappa = 0.1$ in our experiments. We fixed the maximum iteration as 100 and test the algorithms for different values of M and N. We take $L = \max(eig(A^TA))$ and choose the following parameters for each algorithm: $\theta = 0.4, \delta = -0.28, \lambda = 1.3L$ for DIPFBSM; $\lambda = \frac{1}{1.5L}$ for FISTA and PGM; $a = 5, \lambda = \frac{1}{1.5L}$ for FISTA-CD; $\beta_k = 0.35\sqrt{L/(L+l)}$ and $\lambda_k = \frac{1}{L}$ for PGM, where $l = |\min(eig(A^TA))|$; $\beta_k = \beta = \frac{0.7L}{L+l}$, $\alpha_k = \frac{2\beta}{1+\beta}$ and $\lambda_k = \min\{\alpha/\beta, (1.99-2\alpha)/(1-\beta)\}/L$ for cGIGPM, where $l = |\min(eig(A^TA))|$. Each algorithm is initialized at $x^0 = x^1 = \mathbf{0} \in \mathbb{R}^N$ and we considered the values of M and N given as follows:

Case I: $M = 100, N = 300$;

Case II: $M = 200, N = 500$;

Case III: $M = 500, N = 1000$.
Figure 4.7 shows the graphs of objective function value against the number of iterations in each case. In Table 2, we record the objective function value at the last iteration and CPU time taken by each algorithm for each case. The plot trend of Algorithm 1 (DIPFBSM) in Figure 4.7 shows that DIPFBSM is stable which is a desirable property and in the application, it shows that DIPFBSM has few errors affecting the execution of DIPFBSM during numerical implementations.

The curves and fluctuations in Figure 7 are due to the fact that our proposed Algorithm 1 converge faster at the beginning of the iterations and then tail off as iterations progress. This scenario suggests that we consider the ”restarting version” of our proposed Algorithm 1 as part of our future project.

The computational results show that Algorithm 1 (DIPFBSM) is competitive with the algorithms in [7, 10, 37, 38] in the cases considered.

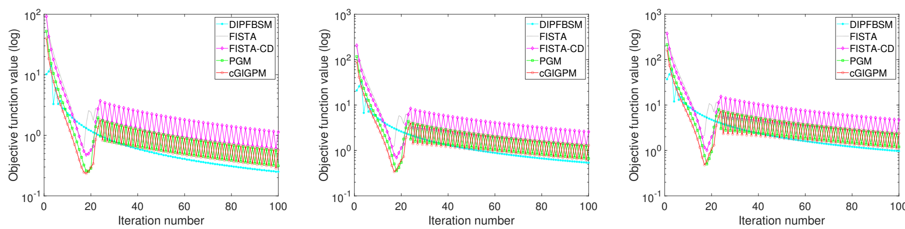

FIGURE 4.7. Example 4.3: Graphs of objective function values (in log) against number of iterations for Case I (left); Case II (middle) and Case III (right).

TABLE 2. Computational result for Example 4.3

		DIPFBSM	FISTA	FISTA-CD	PGM	cGIGPM
Case I	Obj.	0.3391	1.1347	0.3991	0.5705	0.5315
	Time (s)	0.5453	0.9685	0.5581	0.8076	0.7951
Case II	Obj.	0.6165	2.0955	0.6659	1.085	0.6670
	Time (s)	1.6735	1.9163	1.9415	1.7714	1.7465
Case III	Obj.	1.0941	3.7771	1.2754	1.9120	1.9688
	Time (s)	5.1452	5.9780	5.374	5.4499	5.3389

5. CONCLUSION

This paper introduces a forward-backward splitting algorithm with two inertial parameters (one non-negative and the other non-positive) extrapolation step and weak convergence results obtained. Several inertial type forward-backward algorithms in the literature serve as special cases of this algorithm. Numerical implementations arising from LASSO problems in compressed sensing, application to image processing and SCAD penalty problems are given and comparisons with popular algorithms in [7, 10, 37, 38] are also given. The numerical tests demonstrate that our proposed algorithms are promising and competitive. Part of our future considerations is to study the error estimates of our proposed algorithms.

Acknowledgements
The authors are grateful to the reviewers for useful suggestions which improved the contents of this paper. The third was supported by the Grant MOST 111-2115-M-039-001-MY2.

References

[1] F. Alvarez, H. Attouch, An inertial proximal method for monotone operators via discretization of a nonlinear oscillator with damping, Set-Valued Anal. 9 (2001), 3–11.
[2] H. Attouch, A. Cabot, Convergence of a relaxed inertial proximal algorithm for maximally monotone operators, Math. Program. 184 (2020), 243-287.
[3] H. Attouch, A. Cabot, Convergence rates of inertial forward-backward algorithms, SIAM J. Optim. 28 (2018), 849–874.
[4] H. Attouch, J. Peypouquet, P. Redont, A dynamical approach to an inertial forward–backward algorithm for convex minimization, SIAM J. Optim. 24 (2014), 232–256.
[5] J.-F. Aujol, C. Dossal, A. Rondepierre, Optimal convergence rates for Nesterov acceleration, SIAM J. Optim. 29 (2019), 3131-3153.
[6] H. H. Bauschke, P. L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011.
[7] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (2009), 183–202.
[8] R. I. Bot, E. R. Csetnek, D. Meier, Inducing strong convergent into the asymptotic behaviour of proximal splitting algorithms in Hilbert spaces, Optim. Methods Softw. 34 (2019), 489-514.
[9] A. Ben-Tal, A. Nemirovski, Lectures on modern convex optimization: Analysis, Algorithms and Engineering Applications, MPS-SIAM series on optimization, SIAM, Philadelphia, 2001.
[10] A. Chambolle, C. Dossal, On the convergence of the iterates of the ”fast iterative shrinkage/ thresholding algorithm”, J. Optim. Theory Appl. 66 (2015), 968–982.
[11] S.Y. Cho, B.A. Bin Dehaish, X. Qin, Weak convergence of a splitting algorithm in Hilbert spaces, J. Appl. Anal. Comput. 7 (2017), 427-438.
[12] P.L. Combettes, V.R. Wajs, Signal recovery by proximal forward-backward splitting, Multiscale Model Simul. 4 (2005), 1168-1200.
[13] P. Cholamjiak, A generalized forward–backward splitting method for solving quasi inclusion problems in Banach spaces, Numer. Algorithms 8 (2017), 221–239.
[14] S. Chen, D. L. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998), 33-61.
[15] Q. L. Dong, J.Z. Huang, X. H. Li, Y. J. Cho, Th. M. Rassias, MiKM: multi-step inertial Krasnosel'skii–Mann algorithm and its applications, J. Global Optim. 73 (2019), 801–824.
[16] J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, J. Amer. Stat. Assoc. 96 (2001), 1348–1360.
[17] K. Guo, X. Yuan, S. Zeng, Convergence analysis of ISTA and FISTA for “strongly + semi” convex programming, (2016) https://optimization-online.org/2016/06/5506.
[18] O. S. Iyiola, Y. Shehu, Convergence results of two-step inertial proximal point algorithm, Appl. Numer. Math. 182 (2022), 57-75.
[19] J. Liang, Convergence rates of first-order operator splitting methods, PhD thesis, Normandie Université; GREYC CNRS UMR 6072, (2016).
[20] L. Liu, B. Tan, S.Y. Cho, On the resolution of variational inequality problems with a double-hierarchical structure, J. Nonlinear Convex Anal. 21 (2020), 377-386.
[21] Y. Malitsky, T. Pock, A first-order primal-dual algorithm with linesearch, SIAM J.Optim. 28 (2018), 411–432.
[22] Y. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, Dokl. Akad. Nauk. SSSR. 269 (1983), 543–547.
[23] G.B. Passty, Ergodic convergence to a zero of the sum of monotone operators in Hilbert space, J. Math. Anal. Appl. 72 (1979), 383–390.

[24] P. Peeyada, W. Cholamjiak, D. Yambangwai, A hybrid inertial parallel subgradient extragradient-line algorithm for variational inequalities with an application to image recovery, J. Nonlinear Funct. Anal. 2022 (2022), 9.
[25] B. T. Polyak, Some methods of speeding up the convergence of iteration methods, U.S.S.R. Comput. Math. Math. Phys. 4 (1964), 1-17.
[26] B. T. Polyak, Introduction to optimization, Optimization Software, Publications Division, New York, (1987).
[27] C. Poon, J. Liang, Trajectory of Alternating Direction Method of Multipliers and Adaptive Acceleration, In Advances In Neural Information Processing Systems (2019).
[28] C. Poon, J. Liang, Geometry of First-Order Methods and Adaptive Acceleration, arXiv:2003.03910.
[29] X. Qin, S.Y. Cho, L. Wang, A regularization method for treating zero points of the sum of two monotone operators, Fixed Point Theory Appl. 2014 (2014), Article ID 75.
[30] X. Qin, S.Y. Cho, L. Wang, Strong convergence of an iterative algorithm involving nonlinear mappings of nonexpansive and accretive type, Optimization, 67 (2018), 1377-1388.
[31] X. Qin, A weakly convergent method for splitting problems with nonexnpansive mappings, J. Nonlinear Convex Anal. 24 (2023), 1033-1043.
[32] W. A. Smith, R. B. Randall, Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study, Mech. Systems Signal Process. 64–65 (2015), 100–131.
[33] A. Taiwo, O.T. Mewomo, Inertial viscosity with alternative regularization for certain optimization and fixed point problems, J. Appl. Numer. Optim. 4 (2022), 405-423.
[34] B. Tan, S.Y. Cho, Strong convergence of inertial forward–backward methods for solving monotone inclusions, Appl. Anal. 101 (2022), 5386-5414.
[35] R. Tibshirami, Regression shrinkage and selection via lasso, J. Roy. Statist. Soc. Ser. B 58 (1996), 267-288.
[36] Y. Wang, F. Wang, Strong convergence of forward-backward splitting method with multiple parameters in Hilbert spaces, Optimization, 67 (2018), 493–505.
[37] B. Wen, X. Chen, T.K. Pong, Linear convergence of proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems, SIAM J. Optim. 27 (2017), 124–145.
[38] Z. Wu, M. Li, General inertial proximal gradient method for a class of nonconvex nonsmooth optimization problems, Comput. Optim. Appl. 73 (2019), 129-158.
[39] C. Zong, G. Zhang, Y. Tang, Multi-step inertial forward-backward-half forward algorithm for solving monotone inclusion, Linear Multilinear Algebra, (2022) doi: 10.1080/03081087.2022.2040940.