A modified inertial proximal gradient method for minimization problems and applications

Suparat Kesornprom and Prasit Cholamjiak∗

School of Science, University of Phayao, Phayao, Thailand

* Correspondence: Email: prasit.ch@up.ac.th; Tel: +66 5446 6666 Ext. 1713.

Abstract: In this paper, the aim is to design a new proximal gradient algorithm by using the inertial technique with adaptive stepsize for solving convex minimization problems and prove convergence of the iterates under some suitable assumptions. Some numerical implementations of image deblurring are performed to show the efficiency of the proposed methods.

Keywords: convex minimization problem; forward-backward method; adaptive stepsize; inertial method; weak convergence

Mathematics Subject Classification: 65K05, 90C25, 90C30

1. Introduction

In this paper, we investigate the following convex minimization problem

\[
\min_{x \in H}(f(x) + g(x)),
\]

where \( H \) is a real Hilbert space, \( g : H \to (-\infty, +\infty] \) is proper, lower semicontinuous and convex and \( f : H \to \mathbb{R} \) is convex and differentiable with the Lipschitz continuous gradient denoted by \( \nabla f \). It is known that \( x^* \) is a minimizer of \( f + g \) if and only if

\[
0 \in (\partial g + \nabla f)(x^*),
\]

where \( \partial g \) denotes the subdifferential of \( g \).

The convex minimization problem is an important mathematical models which unify numerous issues in applied mathematics for example, signal processing, image reconstruction, machine learning and so on. See [1, 3, 8, 9, 11, 22, 31].

The most popular algorithm for solving the convex minimization problem is the so-called forward-backward algorithm (FB), which generates by a starting point \( x_1 \in H \) and

\[
x_{n+1} = \text{prox}_{\lambda g}(x_n - \lambda \nabla f(x_n)), \ n \geq 1
\]
where $\text{prox}_g$ is the proximal operator of $g$ and the stepsize $\lambda \in (0, 2/L)$, $L$ is the Lipschitz constant of $\nabla f$.

Polyak [21] first proposed the inertial idea to improve the convergence speed of the method. In recent years, many authors introduced various fast iterative methods via inertial technique, for example, [7, 8, 10, 15, 16, 18, 23, 25, 26, 32].

In 2009, Beck and Teboulle [4] introduced the fast iterative shrinkage-thresholding algorithm for linear inverse problem (FISTA). Let $t_0 = 1$ and $x_0 = x_1 \in H$. Compute

$$
\begin{align*}
t_{n+1} &= 1 + \sqrt{1 + 4t_{n-1}^2} - 2t_{n-1}, \\
\theta_n &= t_{n-1} - 1, \\
y_n &= x_n + \theta_n(x_n - x_{n-1}), \\
x_{n+1} &= \text{prox}_{\frac{1}{\lambda}(y_n - \frac{1}{L}\nabla f(y_n))} n \geq 1.
\end{align*}
$$

(1.4)

This improves the convergence speed for $O(1/n^2)$. However, the stepsize is established under the condition of the Lipschitz constant which is not known in general.

In 2000, Tseng [29] proposed a modified forward-backward algorithm (MFB) via the stepsize with linesearch technique as follows. Given $\sigma > 0, \rho \in (0, 1), \delta \in (0, 1)$ and $x_1 \in H$. Compute

$$
\begin{align*}
y_n &= \text{prox}_{\lambda \delta}(x_n - \lambda \nabla f(x_n)), \\
x_{n+1} &= \text{prox}_{\lambda \delta}(y_n - \lambda (\nabla f(y_n) - \nabla f(x_n))), n \geq 1
\end{align*}
$$

(1.5)

where $\lambda_n$ is the largest $\lambda \in \{\sigma, \sigma \rho, \sigma \rho^2, \ldots\}$ satisfying $\lambda \|\nabla f(y_n) - \nabla f(x_n)\| \leq \|y_n - x_n\|.$

In 2020, Padcharoen et al. [20] proposed the modified forward-backward splitting method based on inertial Tseng method (IMFB). Given $\{\lambda_n\} \subset (0, \frac{1}{\gamma}), \{\alpha_n\} \subset [0, \alpha] \subset (0, 1)$. Let $x_0, x_1 \in H$ and compute

$$
\begin{align*}
w_n &= x_n + \theta_n(x_n - x_{n-1}), \\
y_n &= \text{prox}_{\lambda_n \delta}(w_n - \lambda_n \nabla f(w_n)), \\
x_{n+1} &= y_n - \lambda_n (\nabla f(y_n) - \nabla f(w_n)), n \geq 1
\end{align*}
$$

(1.6)

They established weak convergence of the proposed method.

In 2015, Shehu et al. [24] introduced the modified split proximal method (MSP). Let $r : H \to H$ be a contraction mapping with constant $\alpha \in (0, 1)$. Set $\phi(x) = \sqrt{||h(x)||^2 + ||\ell(x)||^2}$ with $h(x) = \frac{1}{2}||I - \text{prox}_{\lambda \rho}(Ax)||^2, \ell(x) = \frac{1}{2}||(I - \text{prox}_{\lambda \rho})x||^2$. Given an initial point $x_1 \in H$ and construct

$$
\begin{align*}
y_n &= x_n - \mu_n A^*(I - \text{prox}_{\lambda \rho})Ax_n, \\
x_{n+1} &= \alpha_n r(x_n) + (1 - \alpha_n)\text{prox}_{\lambda \rho \delta}y_n, n \geq 1
\end{align*}
$$

(1.7)

where the stepsize $\mu_n = \psi_n h(x_n) + \ell(x_n) \phi(x_n) \delta \rho \lambda$ with $0 < \psi_n < 4$. They proved strong convergence theorem for proximal split feasibility problems.

In 2016, Cruz and Nghia [5] presented a fast multistep forward-backward method (FMFB) with a linesearch. Given $\sigma > 0, \mu \in (0, \frac{1}{2}), \rho \in (0, 1)$ and $t_0 = 1$. Choose $x_0, x_1 \in H$ and compute

$$
I_{n+1} = \frac{1 + \sqrt{1 + 4t_{n-1}^2}}{2},
$$
It is known that the proximal operator is single-valued. Moreover, we have
\[ y_n = x_n + \theta_n(x_n - x_{n-1}) \]
\[ x_{n+1} = \text{prox}_{\lambda_n g}(y_n - \lambda_n \nabla f(y_n)), \quad n \geq 1 \]
(1.8)
where \( \lambda_n = \sigma \rho^m n \) and \( m_n \) is the smallest nonnegative integer such that
\[ \lambda_n \| \nabla f(\text{prox}_{\lambda_n g}(y_n - \lambda_n \nabla f(y_n)) - \nabla f(y_n)) \| \leq \mu \| \text{prox}_{\lambda_n g}(y_n - \lambda_n f(y_n)) - y_n \|. \]
(1.9)

Very recently, Malitsky and Tam [17] introduced the forward-reflected-backward algorithm (FRB). Given \( \lambda_0 > 0, \delta \in (0, 1), \gamma \in \{1, \beta^{-1}\} \) and \( \beta \in (0, 1) \). Compute
\[ x_{n+1} = \text{prox}_{\lambda_n g}(x_n - \lambda_n \nabla f(x_n) - \lambda_{n-1} (\nabla f(x_n) - \nabla f(x_{n-1}))), \quad n \geq 1 \]
(1.10)
where the stepsize \( \lambda_n = \gamma \lambda_{n-1} \beta^i \) with \( i \) being the smallest nonnegative integer satisfying \( \lambda_n \| \nabla f(x_{n+1}) - \nabla f(x_n) \| \leq \frac{\delta}{\beta} \| x_{n+1} - x_n \| \).

Very recently, Hieu et al. [13] proposed the modified forward-reflected-backward method (MFRB) with adaptive stepsize. Given \( x_0, x_1 \in H, \lambda_0, \lambda_1 > 0, \mu \in (0, \frac{1}{2}) \):
\[ x_{n+1} = \text{prox}_{\lambda_n g}(x_n - \lambda_n \nabla f(x_n) - \lambda_{n-1} (\nabla f(x_n) - \nabla f(x_{n-1}))), \]
\[ \lambda_{n+1} = \min\{\lambda_n, \frac{\mu \| x_{n+1} - x_n \|}{\| \nabla f(x_{n+1}) - \nabla f(x_n) \|}\}, \quad n \geq 1. \]
(1.11)

This stepsize allows the proposed method without knowing the Lipschitz constant to solve the problem.

Inspired and motivated by previous works, we propose based on the adaptive stepsize, the inertial proximal gradient algorithm for convex minimization problems. This method requires more flexible conditions than the fixed stepsize does. We then establish weak convergence of our scheme under some assumptions. Moreover, we present some numerical experiments in image deblurring. It reveals that our algorithm outperforms other methods.

2. Basic definitions and lemmas

In this section, we provide some definitions and lemmas for proving our theorem.

Weak and strong convergence of a sequence \( \{x_n\} \subset \Omega \) to \( z \in \Omega \) are denoted by \( x_n \rightharpoonup z \) and \( x_n \to z \), respectively.

Let \( g : H \to (-\infty, +\infty] \) be a proper, lower semicontinuous and convex function. We denote the domain of \( g \) by \( \text{dom} g = \{x \in H | g(x) < +\infty\} \). For any \( x \in \text{dom} g \), the subdifferential of \( g \) at \( x \) is defined by
\[ \partial g(x) = \{v \in H | (v, y - x) \leq g(y) - g(x), \quad y \in H\}. \]

Recall that the proximal operator \( \text{prox}_g : \text{dom}(g) \to H \) is given by \( \text{prox}_g(x) = (I + \partial g)^{-1}(x), \quad z \in H \). It is known that the proximal operator is single-valued. Moreover, we have
\[ \frac{z - \text{prox}_{\lambda g}(z)}{\lambda} \in \partial g(\text{prox}_{\lambda g}(z)) \quad \text{for all} \quad z \in H, \quad \lambda > 0. \]
(2.1)
Definition 2.1. Let $S$ be a nonempty subset of $H$. A sequence $\{x_n\}$ in $H$ is said to be quasi-Fejér convergent to $S$ if and only if for all $x \in S$ there exists a positive sequence $\{\varepsilon_n\}$ such that $\sum_{n=1}^{\infty} \varepsilon_n < +\infty$ and $\|x_{n+1} - x\|^2 \leq \|x_n - x\|^2 + \varepsilon_n$ for all $n \geq 1$. If $\{\varepsilon_n\}$ is a null sequence, we say that $\{x_n\}$ is Fejér convergent to $S$.

Lemma 2.1. [6] The subdifferential operator $\partial g$ is maximal monotone. Moreover, the graph of $\partial g$, $\text{Gph}(\partial g) = \{(x, v) \in H \times H : v \in \partial g(x)\}$ is demiclosed, i.e., if the sequence $\{(x_n, v_n)\} \subset \text{Gph}(\partial g)$ satisfies that $\{x_n\}$ converges weakly to $x$ and $\{v_n\}$ converges strongly to $v$, then $(x, v) \in \text{Gph}(\partial g)$.

Lemma 2.2. [19] Let $\{a_n\}$, $\{b_n\}$ and $\{c_n\}$ be real positive sequences such that

$$a_{n+1} \leq (1 + c_n)a_n + b_n, \quad n \geq 1.$$

If $\sum_{n=1}^{\infty} c_n < +\infty$ and $\sum_{n=1}^{\infty} b_n < +\infty$, then $\lim_{n \to +\infty} a_n$ exists.

Lemma 2.3. [12] Let $\{a_n\}$ and $\{\theta_n\}$ be real positive sequences such that

$$a_{n+1} \leq (1 + \theta_n)a_n + \theta_n a_{n-1}, \quad n \geq 1.$$

Then, $a_{n+1} \leq K \cdot \prod_{i=1}^{n} (1 + \theta_i)$ where $K = \max\{a_1, a_2\}$. Moreover, if $\sum_{n=1}^{\infty} \theta_n < +\infty$, then $\{a_n\}$ is bounded.

Lemma 2.4. [2, 14] If $\{x_n\}$ is quasi-Fejér convergent to $S$, then we have:

(i) $\{x_n\}$ is bounded.

(ii) If all weak accumulation points of $\{x_n\}$ is in $S$, then $\{x_n\}$ weakly converges to a point in $S$.

3. Main result

In this section, we assume that the following conditions are satisfied for our convergence analysis:

(A1) The solution set of the convex minimization problem (1.1) is nonempty, i.e., $\Omega = \arg\min(f + g) \neq \emptyset$.

(A2) $f, g : H \to (-\infty, +\infty]$ are two proper, lower semicontinuous and convex functions.

(A3) $f$ is differentiable on $H$ and $\nabla f$ is Lipschitz continuous on $H$ with the Lipschitz constant $L > 0$.

We next introduce a new inertial forward-backward method for solving (1.1).

Algorithm 3.1. **Inertial modified forward-backward method (IMFB)**

**Initialization:** Let $x_0 = x_1 \in H$, $\lambda_1 > 0$, $\theta_1 > 0$ and $\delta \in (0, 1)$.

**Iterative step:** For $n \geq 1$, calculate $x_{n+1}$ as follows:

**Step 1. Compute the inertial step:**

$$w_n = x_n + \theta_n(x_n - x_{n-1}).$$

**Step 2. Compute the forward-backward step:**

$$y_n = \text{prox}_{\lambda_n g}(w_n - \lambda_n \nabla f(w_n)).$$

**Step 3. Compute the $x_{n+1}$ step:**

$$x_{n+1} = y_n - \lambda_n(\nabla f(y_n) - \nabla f(w_n)).$$
where

\[
\lambda_{n+1} = \begin{cases} 
\min\{ \delta \frac{||w_n - y_n||}{||\nabla f(w_n) - \nabla f(y_n)||}, \lambda_n \} & \text{if } ||\nabla f(w_n) - \nabla f(y_n)|| \neq 0; \\
\lambda_n & \text{otherwise.}
\end{cases}
\]

(3.4)

Set \( n = n + 1 \) and return to Step 1.

**Remark 3.1.** It is easy to see that the sequence \( \{\lambda_n\} \) is non-increasing. From the Lipschitz continuity of \( \nabla f \), there exists \( L > 0 \) such that \( ||\nabla f(w_n) - \nabla f(y_n)|| \leq L||w_n - y_n|| \). Hence,

\[
\lambda_{n+1} = \min\left\{ \frac{\delta ||w_n - y_n||}{||\nabla f(w_n) - \nabla f(y_n)||}, \lambda_n \right\} \geq \min\left\{ \frac{\delta}{L}, \lambda_n \right\}.
\]

(3.5)

By the definition of \( \{\lambda_n\} \), it implies that the sequence \( \{\lambda_n\} \) is bounded from below by \( \min\{\lambda_0, \frac{\delta}{\lambda}\} \). So, we obtain \( \lim_{n \to \infty} \lambda_n = \lambda > 0 \).

**Lemma 3.1.** Let \( \{x_n\} \) be generated by Algorithm 3.1. Then

\[
||x_{n+1} - x^*||^2 \leq ||w_n - x^*||^2 - \left(1 - \frac{\delta^2 \lambda_n^2}{\lambda_{n+1}^2}\right)||y_n - w_n||^2, \forall x^* \in \Omega.
\]

(3.6)

**Proof.** Let \( x^* \in \Omega \). Then

\[
||x_{n+1} - x^*||^2 = ||y_n - \lambda_n(\nabla f(y_n) - \nabla f(w_n)) - x^*||^2 \\
= ||y_n - x^*||^2 + \lambda_n^2||\nabla f(y_n) - \nabla f(w_n)||^2 \\
- 2\lambda_n \langle y_n - x^*, \nabla f(y_n) - \nabla f(w_n) \rangle \\
= ||y_n - w_n + w_n - x^*||^2 + \lambda_n^2||\nabla f(y_n) - \nabla f(w_n)||^2 \\
- 2\lambda_n \langle y_n - x^*, \nabla f(y_n) - \nabla f(w_n) \rangle \\
= ||w_n - x^*||^2 + ||y_n - w_n||^2 + 2\langle w_n - x^*, y_n - w_n \rangle \\
- 2\lambda_n \langle y_n - x^*, \nabla f(y_n) - \nabla f(w_n) \rangle + \lambda_n^2||\nabla f(y_n) - \nabla f(w_n)||^2 \\
= ||w_n - x^*||^2 + ||y_n - w_n||^2 + 2\langle w_n - y_n + y_n - x^*, y_n - w_n \rangle \\
- 2\lambda_n \langle y_n - x^*, \nabla f(y_n) - \nabla f(w_n) \rangle + \lambda_n^2||\nabla f(y_n) - \nabla f(w_n)||^2 \\
= ||w_n - x^*||^2 + ||y_n - w_n||^2 - 2\langle y_n - w_n, y_n - w_n \rangle \\
+ 2\langle y_n - x^*, y_n - w_n \rangle - 2\lambda_n \langle y_n - x^*, \nabla f(y_n) - \nabla f(w_n) \rangle \\
+ \lambda_n^2||\nabla f(y_n) - \nabla f(w_n)||^2 \\
= ||w_n - x^*||^2 + ||y_n - w_n||^2 - 2||y_n - w_n||^2 + 2||y_n - x^*, y_n - w_n|| \\
- 2\lambda_n \langle y_n - x^*, \lambda_n(\nabla f(y_n) - \nabla f(w_n)) \rangle + \lambda_n^2||\nabla f(y_n) - \nabla f(w_n)||^2 \\
= ||w_n - x^*||^2 - ||y_n - w_n||^2 \\
- 2\langle y_n - x^*, w_n - y_n + \lambda_n(\nabla f(y_n) - \nabla f(w_n)) \rangle \\
+ \lambda_n^2||\nabla f(y_n) - \nabla f(w_n)||^2.
\]

(3.7)

Note that

\[
\lambda_{n+1} = \min\left\{ \frac{\delta ||w_n - y_n||}{||\nabla f(w_n) - \nabla f(y_n)||}, \lambda_n \right\} \leq \frac{\delta ||w_n - y_n||}{||\nabla f(w_n) - \nabla f(y_n)||}.
\]

(3.8)
It follows that
\[ \|\nabla f(w_n) - \nabla f(y_n)\| \leq \frac{\delta}{\lambda_{n+1}} \|w_n - y_n\|. \] (3.9)

Combining (3.7) and (3.9), we obtain
\[
\|x_{n+1} - x^*\|^2 \leq \|w_n - x^*\|^2 - 2\langle y_n - x^*, w_n - y_n + \lambda_n(\nabla f(y_n) - \nabla f(w_n))\rangle + (1 - \frac{\delta^2}{\lambda^2})\|y_n - w_n\|^2.
\] (3.10)

From (3.2), we see that
\[
w_n - \lambda_n \nabla f(w_n) \in (I + \lambda_n \partial g)y_n.
\] Since \(\partial g\) is maximal monotone, then there is \(u_n \in \partial g(y_n)\) such that
\[
w_n - \lambda_n \nabla f(w_n) = y_n + \lambda_n u_n.
\] (3.11)

This shows that
\[
u_n = \frac{1}{\lambda_n}(w_n - \lambda_n \nabla f(w_n) - y_n).
\] (3.12)

Since \(0 \in (\nabla f + \partial g)(x^*)\) and \(\nabla f(y_n) + u_n \in (\nabla f + \partial g)y_n\), we get
\[
\langle \nabla f(y_n) + u_n, y_n - x^* \rangle \geq 0.
\] (3.13)

Substituting (3.12) into (3.13), we have
\[
\frac{1}{\lambda_n} \langle w_n - \lambda_n \nabla f(w_n) - y_n + \lambda_n \nabla f(y_n), y_n - x^* \rangle \geq 0.
\] (3.14)

This implies that \(\langle w_n - \lambda_n \nabla f(w_n) - y_n + \lambda_n \nabla f(y_n), y_n - x^* \rangle \geq 0\). Using (3.10), we derive
\[
\|x_{n+1} - x^*\|^2 \leq \|w_n - x^*\|^2 - (1 - \frac{\delta^2}{\lambda^2})\|y_n - w_n\|^2.
\] (3.15)

\[\square\]

**Lemma 3.2.** Let \(\{x_n\}\) be generated by Algorithm 3.1. If \(\sum_{n=1}^{\infty} \theta_n < \infty\), then \(\lim_{n \to \infty} \|x_n - x^*\|\) exists for all \(x^* \in \Omega\).

**Proof.** Let \(x^* \in \Omega\). From Lemma 3.1, we see that
\[
\|x_{n+1} - x^*\| \leq \|w_n - x^*\|. \] (3.16)

So, we have
\[
\|x_{n+1} - x^*\| \leq \|w_n - x^*\|
\]
From (3.15) and (3.20), we have
\[ \sum \]
Hence
\[ \sum \]

By Lemma 2.3, we conclude that
\[ \sum \]

Lemma 3.3. Let \{x_n\} be generated by Algorithm 3.1. If \( \sum \theta_n < \infty \), then
\[ \sum \]

Proof. We see that
\[ \sum \]
From (3.15) and (3.20), we have
\[ \sum \]
Note that \( \theta_n \|x_n - x_{n-1}\| \rightarrow 0 \) and \( \lim \|x_n - x^*\| \) exists by Lemma 3.2. From (3.1) and (3.21), we have \( \|w_n - x_n\| \rightarrow 0 \) and \( \|w_n - y_n\| \rightarrow 0 \), respectively. It is easy to see that \( \|x_n - y_n\| \rightarrow 0 \). Since \( \nabla f \) is uniformly continuous, we obtain
\[ \sum \]
From (3.3) and (3.22), we get
\[ \sum \]
Thus, we have
\[ \sum \]
\[ \sum \]
Theorem 3.1. Let \( \{x_n\} \) be generated by Algorithm 3.1. If \( \sum_{n=1}^{\infty} \theta_n < \infty \), then \( \{x_n\} \) weakly converges to a point in \( \Omega \).

Proof. Since \( \{x_n\} \) is bounded, there exists a subsequence \( \{x_{n_k}\} \) of \( \{x_n\} \) such that \( x_{n_k} \rightharpoonup \bar{x} \in H \). From Lemma 3.3, we obtain \( x_{n_k} \rightharpoonup \bar{x} \). We note that

\[
y_{n_k} = \text{prox}_{\lambda_n g} (w_{n_k} - \lambda_n \nabla f(w_{n_k})).
\]

(3.25)

From (2.1), we obtain

\[
\frac{w_{n_k} - \lambda_n \nabla f(w_{n_k}) - y_{n_k}}{\lambda_n} \in \partial g(y_{n_k}).
\]

(3.26)

Hence

\[
\frac{w_{n_k} - y_{n_k}}{\lambda_n} - \nabla f(w_{n_k}) + \nabla f(y_{n_k}) \in \partial g(y_{n_k}) + \nabla f(y_{n_k}).
\]

(3.27)

Since \( \|x_n - y_n\| \to 0 \), we also have \( y_{n_k} \rightharpoonup \bar{x} \). Letting \( k \to \infty \) in (3.27) and using (3.22), by Lemma 2.1 and Remark 3.1, we get

\[
0 \in (\nabla f + \partial g)(\bar{x}).
\]

(3.28)

So \( \bar{x} \in \Omega \). From (3.21) we see that \( \{x_n\} \) is a quasi-Fejer sequence. Hence, by Lemma 2.4, we conclude that \( \{x_n\} \) weakly converges to a point in \( \Omega \). This completes the proof.

\[\square\]

4. Numerical experiment in image deblurring

The image deblurring can be modeled by the following linear equation system:

\[
Ax = b + v,
\]

where \( A \in \mathbb{R}^{N \times M} \) is the blurring matrix, \( x \in \mathbb{R}^N \) the original image, \( b \in \mathbb{R}^M \) the degraded image and \( v \in \mathbb{R}^M \) is the noisy.

An approximation of the clean image can be found by the following LASSO problem [27]:

\[
\min_{x \in \mathbb{R}^N} \left\{ \frac{1}{2} \|b - Ax\|_2^2 + \tau \|x\|_1 \right\},
\]

(4.1)

where \( \tau \) is a positive parameter, \( \| \cdot \|_1 \) is the \( \ell_1 \)-norm, and \( \| \cdot \|_2 \) is the Euclidean norm.

It is known that (4.1) can be written in the form (1.1) by defining \( f(x) = \frac{1}{2} \|b - Ax\|_2^2 \) and \( g(x) = \tau \|x\|_1 \). We compare our algorithm (IMFB) with FISTA, MFB, FRB, MFRB, IMFB, MSP and FMFB.

In method IMFB, we set \( t_0 = 1, t_n = \frac{1 + \sqrt{1 + 4t_{n-1}^2}}{2} \) and

\[
\theta_n = \begin{cases} 
\frac{t_{n-1}-1}{t_n}, & \text{if } 1 \leq n \leq 1000 \\
0 & \text{otherwise}.
\end{cases}
\]
The regularization parameters are chosen by $\tau = 10^{-5}$ and $x_0 = x_1 = (1, 1, 1, ..., 1) \in \mathbb{R}^N$. We set the following parameters in Table 1.

<table>
<thead>
<tr>
<th>Table 1. The parameters for each methods.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
</tr>
<tr>
<td>$\lambda_0 = 0.1$</td>
</tr>
<tr>
<td>$\lambda_1 = 0.5$</td>
</tr>
<tr>
<td>$\sigma = 0.1$</td>
</tr>
<tr>
<td>$\rho = 0.8$</td>
</tr>
<tr>
<td>$\beta = 0.5$</td>
</tr>
<tr>
<td>$\gamma = 1/\beta$</td>
</tr>
<tr>
<td>$\alpha_n = 1/(n+1)$</td>
</tr>
<tr>
<td>$\psi = 2$</td>
</tr>
<tr>
<td>$r(x_n) = \frac{1}{2}x_n$</td>
</tr>
</tbody>
</table>

In this example, we set all parameters as in Table 1. For the experiments, we use the sizes $251 \times 189$ for RGB images which are blurred by the following blur types:

(i) Motion blur with motion length of 45 pixels and motion orientation $180^\circ$.

(ii) Gaussian blur of filter size $5 \times 5$ with standard deviation 5.

(iii) Out of focus with radius 7.

We add Poisson noise and use a Fast Fourier Transform (FFT) for converting it to the frequency domain. Structural similarity index measure (SSIM) [30] is used for measuring the similarity between two images. Peak-signal-to-noise ratio (PSNR) in decibel (dB) [28] is defined by

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right)$$

where $MSE=\|x_n - x\|^2$ and $x$ is the original image. It is noted that, a higher PSNR generally indicates that the reconstruction is of higher quality. The resultant SSIM index is a decimal value between 0 and 1, and value 1 is indicates perfect structural similarity.

The numerical experiments have been carried out in Matlab environment (version R2020b) on MacBook Pro M1 with ram 8 GB. For the results recovering the degraded RGB images, we limit the iterations to 1,000. We report the numerical results in Table 2.

In Table 2, we see that IMFB has a higher PSNR than FISTA, MFB, FRB, MFRB, IMFB, MSP, FMFB for the same number of iterations. Moreover, SSIM of IMFB is closer to 1 than other methods. This shows that our algorithm has a better convergence than other methods for this example. However, we observe that IMFB has a less CPU time than other methods.
Table 2. The comparison of PSNR, SSIM and CPU time in seconds for each methods of the restored images.

| Methods | Motion blur | | | Gaussian blur | | | Out of focus | | |
|---------|-------------|---------|-------------|-------------|---------|-------------|---------|-------|-------------|---------|-------------|---------|
|         | PSNR | SSIM | CPU | PSNR | SSIM | CPU | PSNR | SSIM | CPU | PSNR | SSIM | CPU |
| FISTA   | 25.1122 | 0.7694 | 48.0184 | 34.3744 | 0.9320 | 47.2637 | 30.9043 | 0.8672 | 47.2831 |
| MFB      | 24.8516 | 0.7640 | 71.2241 | 34.9546 | 0.9405 | 71.4400 | 28.3107 | 0.8152 | 70.4725 |
| FRB      | 27.5733 | 0.8536 | 113.2112 | 38.6119 | 0.9703 | 112.5153 | 31.8188 | 0.8886 | 111.3155 |
| MFRB     | 25.5158 | 0.7893 | 77.6740 | 36.0870 | 0.9515 | 70.3914 | 29.3660 | 0.8412 | 69.7562 |
| IMFB     | 33.6105 | 0.9453 | 42.8921 | 41.1854 | 0.9818 | 42.7064 | 34.8978 | 0.9293 | 42.9258 |
| MSP      | 34.8218 | 0.9560 | 92.9287 | 38.0609 | 0.9675 | 93.2503 | 32.0816 | 0.8941 | 93.2242 |
| FMFB     | 40.8550 | 0.9785 | 64.8279 | 43.9280 | 0.9888 | 64.9999 | 38.0780 | 0.9544 | 64.8632 |
| IMFB     | 46.7885 | 0.9920 | 75.7321 | 47.3368 | 0.9939 | 75.5435 | 41.0665 | 0.9743 | 74.7891 |

We next show the different types of blurred RGB images with the PSNR in Figure 1.

Motion blur

![Motion blur images](a) original image 20.4054 (b) blurred cropped image (c) cropped image (b)

Gaussian blur

![Gaussian blur images](c) original image 28.7148 (d) blurred cropped image (c) cropped image (d)

Out of focus

![Out of focus images](e) original image 23.5639 (f) blurred cropped image (e) cropped image (f)

Figure 1. (a), (c) and (e) show the original images for each blurred RGB images with noise, (b), (d) and (f) show the images degraded by each blurred.

We next show the restored images of RGB images for Motion blur with the PSNR in Figure 2.
Figure 2. Recovered images via the different methods for degraded images by Motion blur.

We next show the restored images of RGB images for Gaussian blur with the PSNR in Figure 3.

Figure 3. Recovered images via the different methods for degraded images by Gaussian blur.

We next show the restored images of RGB images for out of focus with the PSNR in Figure 4.
Figure 4. Recovered images via the different methods for degraded images by out of focus.

Figure 5. Graphs of PSNR and SSIM values for each blurred image and restored images by FISTA, MFB, FRB, MFRB, IMFB, MSP, FMFB and IMFB.
The results of the numerical experiments are summarized in Table 2. Figure 1 shows the original and blurred images for this experiment. In Figures 2–5, we report all results that include the recovered images via each algorithm. It is shown that IMFB outperforms FISTA, MFB, FRB, MFRB, IMFB, MSP and FMFB in terms of PSNR and SSIM.

5. Conclusions

In this paper, we established the convergence theorem of the iterates generated by a new modified inertial forward-backward algorithm with adaptive stepsizes under some suitable conditions for convex minimization problems. We applied our main result to image recovery. It was shown that our proposed method outperforms other methods in terms of PSNR and SSIM. In future work, we will study the convergence rate of the iteration.

Acknowledgments

This project is funded by the National Research Council of Thailand (NRCT) under grant No. N41A640094. The authors wish to thank University of Phayao and Thailand Science Research and Innovation grant No. FF65-UOE001.

Conflict of interest

The authors declare no conflict of interest.

References


©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)