AIMS Mathematics, 9(7): 18841–18859.
DOI:10.3934/math.2024917
Received: 14 March 2024
Revised: 11 May 2024
Accepted: 21 May 2024
Published: 05 June 2024

https://www.aimspress.com/journal/Math

Research article

A recent proximal gradient algorithm for convex minimization problem using double inertial extrapolations

Suparat Kesornprom[1,2], Papatsara Inkrong[3], Uamporn Witthayarat[3] and Prasit Cholamjiak[3,*]

[1] Research Center in Optimization and Computational Intelligence for Big Data Prediction, Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand
[2] Office of Research Administration, Chiang Mai University, Chiang Mai 50200, Thailand
[3] School of Science, University of Phayao, Phayao 56000, Thailand

* **Correspondence:** Email: prasit.ch@up.ac.th; Tel: +66843731038.

Abstract: In this study, we suggest a new class of forward-backward (FB) algorithms designed to solve convex minimization problems. Our method incorporates a linesearch technique, eliminating the need to choose Lipschitz assumptions explicitly. Additionally, we apply double inertial extrapolations to enhance the algorithm's convergence rate. We establish a weak convergence theorem under some mild conditions. Furthermore, we perform numerical tests, and apply the algorithm to image restoration and data classification as a practical application. The experimental results show our approach's superior performance and effectiveness, surpassing some existing methods in the literature.

Keywords: forward-backward algorithm; minimization problem; inertial extrapolation; weak convergence
Mathematics Subject Classification: 46E20, 46N40, 65K05, 68T07, 90C25

1. Introduction

This work focuses on solving a convex minimization problem in a real Hilbert space $\mathcal{H}$ equipped with the product $\langle\cdot,\cdot\rangle$ and the norm $\|\cdot\|$. Let $f, g : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ be proper, convex, and lower semicontinuous. Additionally, assume that f is Fréchet differentiable on $\mathcal{H}$. In this context, the convex minimization problem is formulated as follows:

$$\min_{u\in\mathcal{H}}\{f(u) + g(u)\}. \tag{1.1}$$

We denote Γ^* as the solution set of problem (1.1). This problem encompasses various real-world challenges, including machine learning, signal and image processing, and the projection and distance

between sets. Problem (1.1) is recognized for the fixed-point equation:

$$u = (I + \beta\partial g)^{-1}(I - \beta\nabla f)(u), \tag{1.2}$$

where $\beta > 0$, ∇f represents the gradient of the function f, and ∂g stands for the subdifferential function g. It is noted that $\partial g(u) = \{b \in \mathcal{H} : g(z) \geq g(u) + \langle b, z - u\rangle, \quad \text{for all } z \in \mathcal{H}\}$. In this connection, (1.2) presents the possibility of iterating

$$u_{n+1} = (I + \beta_n\partial g)^{-1}(I - \beta_n\nabla f)(u_n), \tag{1.3}$$

where the step size β_n lies in an appropriate bounded interval. This technique is referred to the FB method, encompassing the proximal point algorithm [9, 19, 22, 26] and the gradient method [8, 29, 31]. To establish the convergence of algorithms based on (1.3), ∇f is commonly considered Lipschitz continuous. The step size β_n is constrained to be bounded below and less than constants associated with the Lipschitz constant that is not known in practice. Combettes and Wajs [6] suggested the iterative sequence (1.4), derived from the classical FB iteration (1.3). Let $\mu \in \left(0, \min\left\{1, \frac{1}{L}\right\}\right)$ and let $u_0 \in \mathbb{R}^N$. For $n \geq 1$, define

$$\begin{cases} w_n = u_n - \beta_n\nabla f(u_n), \\ u_{n+1} = u_n + \lambda_n\left((I + \beta_n\partial g)^{-1}(w_n) - u_n\right), \end{cases} \tag{1.4}$$

where $\beta_n \in \left[\mu, \frac{2}{L} - \mu\right]$, $\lambda_n \in [L, 1]$, and L is the Lipschitz constant of ∇f.

Cruz and Nghia [3] introduced a rapid FB method utilizing linesearch iterates to solve (1.4). An essential benefit of this method is eliminating the Lipschitz condition during computation. Given that $\beta_n > 0,\ \delta > 0,\ \sigma \in (0, 1)$, $\gamma \in \left(0, \frac{1}{2}\right)$, $s_1 = 1$, and $u_0 = u_1 \in \operatorname{dom} g$, calculate

$$\begin{cases} s_{n+1} = \dfrac{1 + \sqrt{1 + 4s_n^2}}{2}, \\ w_n = u_n + \dfrac{s_n - 1}{s_{n+1}}(u_n - u_{n-1}), \\ u_{n+1} = (I + \beta_n\partial g)^{-1}(I - \beta_n\nabla f)(w_n). \end{cases} \tag{1.5}$$

where $\beta_n = \delta\sigma^{k_n}$ and k_n is the smallest nonnegative integer such that

$$\beta_n \left\|\nabla f\left(u_{n+1}\right) - \nabla f\left(w_n\right)\right\| \leq \gamma \left\|u_{n+1} - w_n\right\|.$$

It has been demonstrated that the sequence $\{u_n\}$ converges weakly to a minimizer of $f + g$.

In 2019, Kankam et al. [17] developed the FB method by using a new linesearch form. Let $u_n \in \operatorname{dom} g,\ \delta > 0,\ \sigma \in (0, 1)$, and $\gamma > 0$. For $n \geq 1$, set

$$\begin{cases} w_n = (I + \beta_n\partial g)^{-1}(I - \beta_n\nabla f)(u_n), \\ u_{n+1} = (I + \beta_n\partial g)^{-1}(I - \beta_n\nabla f)(w_n), \end{cases} \tag{1.6}$$

where $\beta_n = \delta\sigma^{k_n}$ and k_n is the smallest nonnegative integer such that

$$\beta_n \cdot \max\left\{\left\|\nabla f\left(u_{n+1}\right) - \nabla f\left(w_n\right)\right\|, \left\|\nabla f\left(w_n\right) - \nabla f\left(u_n\right)\right\|\right\} \leq \gamma\left(\left\|u_{n+1} - w_n\right\| + \left\|w_n - u_n\right\|\right).$$

It was verified that $\{u_n\}$ converges weakly to a solution. Moreover, numerical experiments in signal recovery are also provided. It was also shown that this method had better convergence than a classical FB and the FB using linesearch defined in [3].

The inertial extrapolation is known for improving the convergence rate of iterative methods, with Polyak [24] being the first to consider it. After that, Nesterov [20] popularized it by solving programming problems, and it was later developed by Beck and Teboulle [2] for structured convex minimization problems. Based on the insight presented in [18], it was observed that utilizing more than two points, specifically u_n and u_{n-1}, can improve acceleration. This enhancement is explained by employing a double inertial extrapolation, outlined as

$$w_n = u_n + \alpha(u_n - u_{n-1}) + \beta(u_{n-1} - u_{n-2}),$$

where $\alpha > 0$ and $\beta > 0$ can provide acceleration. In [25], the limitations of one inertial acceleration in ADMM were examined along with an alternative called adaptive acceleration for ADMM. At the same time, Jolaoso et al. [15] proposed an alternative to the conventional one inertial FB splitting method. They present the utilization of double inertial extrapolation in the FB splitting method to increase the convergence speed. Additionally, Polyak [23] also examined the potential of multi-step inertial methods in improving the optimization speed despite the absence of established convergence or rate outcomes in [23]. Recently, many studies have been conducted on multi-step inertial methods, and specific results have been presented in [5, 7, 14].

From this connection, we aim to combine a new linesearch in (1.6) with a double inertial extrapolation for modifying the FB splitting method and proving its weak convergence. The contributions of this work are summarized below.

- We present a new modified FB algorithm with double inertial extrapolations designed explicitly for convex minimization problems in real Hilbert spaces. Our proposed method has a forward assessment of ∇f and a backward evaluation of ∂g in two steps in each iteration. Furthermore, double inertial extrapolations are incorporated into our algorithm to expedite convergence.
- The proposed algorithms achieve weak convergence under some mild conditions through a linesearch technique in [17] for updating β_n. The main advantage of the technique is that the Lipschitz constants of the gradient of functions do not require computation. It is noted that our linesearch is different from that of Cruz and Nghia [3].
- We show numerical experiments on image restoration and data classification to illustrate the validity of our proposed algorithm. The numerical results reveal that our method is efficient and outperforms related methods in the literature.

The rest of this paper is organized as follows: Section 2 contains some useful definitions and lemmas used for our proof. In Section 3, we give a proof of our proposed algorithm's weak convergence. Section 4 contains numerical tests on image restoration and data classification. Finally, we give a conclusion in Section 5.

2. Preliminaries

In this section, we collect some necessary definitions and lemmas that will be used in our analysis. We use $u_n \rightharpoonup v$ to indicate that $\{u_n\}$ converges weakly to v. Let $k : \mathcal{H} \to \mathbb{R}$ be a proper, lower semicontinuous, and convex function. The domain of a function k is denoted by $\operatorname{dom} k := \{u \in \mathcal{H} | k(u) < +\infty\}$.

We also recognize the following definitions:

(1) $k : \mathcal{H} \to \mathbb{R}$ is *convex* if

$$k(\rho u + (1-\rho)v) \leq \rho k(u) + (1-\rho)k(v),$$

$\forall \rho \in (0,1)$ and $u, v \in \mathcal{H}$.

(2) A differentiable function k is *convex* if and only if

$$k(u) + \langle \nabla k(u), v - u \rangle \leq k(v), \quad \forall v \in \mathcal{H}.$$

(3) An element $g \in \mathcal{H}$ is designated as a subgradient of $k : \mathcal{H} \to \mathbb{R}$ at u when

$$k(u) + \langle \nabla g, v - u \rangle \leq k(v), \quad \forall v \in \mathcal{H}.$$

(4) The proximal operator $\text{prox}_g : \mathcal{H} \to \mathcal{H}$ associated with g is defined as follows:

$$\text{prox}_g(w) = (I + \partial g)^{-1}(w), \quad \forall w \in \mathcal{H}.$$

It is known that the proximal operator is a single-valued function with a full domain. Furthermore, as derived in [3], we know that

$$\frac{w - \text{prox}_{\beta g}(w)}{\beta} \in \partial g(\text{prox}_{\beta g}(w)), \quad \forall w \in \mathcal{H},\ \beta > 0. \tag{2.1}$$

Fact 2.1. *([1], Proposition 17.2). Let $k : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ be a proper, lower semicontinuous, and convex function. Then, for $u \in \text{dom}\, k$ and $v \in \mathcal{H}$,*

$$k'(u, v - u) + k(u) \leq k(v).$$

Lemma 2.2. *[27] The following hold:*
(i) $\|ca + (1-c)b\|^2 = c\|a\|^2 + (1-c)\|b\|^2 - c(1-c)\|a - b\|^2$, $\forall c \in [0,1]$ and $\forall a, b \in \mathcal{H}$;
(ii) $\|a \pm b\|^2 = \|a\|^2 \pm 2\langle a, b \rangle + \|b\|^2, \quad \forall a, b \in \mathcal{H}$.

Lemma 2.3. *[4] The subdifferential operator ∂k of a convex function k is maximal monotone. Furthermore, the graph of ∂k,* $\text{Graph}(\partial k) = \{(u, v) \in \mathcal{H} \times \mathcal{H} : v \in \partial k(v)\}$, *is demiclosed, that is, if a sequence $\{u_n, v_n\} \subset \text{Graph}(\partial k)$ is such that $\{u_n\}$ converges weakly to u and $\{v_n\}$ converges strongly to v, then $(u, v) \in \text{Graph}(\partial k)$.*

Lemma 2.4. *[21] Let C be a subset of $\mathcal{H}$ and $\{u_n\}$ be a sequence in $\mathcal{H}$ such that*
(i) for any $v \in C$, $\lim_{n\to\infty} \|u_n - v\|$ exists;
(ii) each weak-cluster point of $\{u_n\}$ is in C.
Then $\{u_n\}$ converges weakly to a point in C.

3. Algorithm and convergence theorem

Next, we modify double inertial terms and a new linesearch for solving (1.1). Assume that Γ^* is nonempty. Moreover, we suppose the following:

(A1) $f, g : \mathcal{H} \longrightarrow \mathbb{R} \cup \{+\infty\}$ are proper, lower semicontinuous, and convex functions and that f is differentiable on $\mathcal{H}$.

(A2) ∇f is uniformly continuous and bounded on a bounded subset of $\mathcal{H}$.

We begin by presenting the following supporting lemma, which is motivated by [11].

Lemma 3.1. *[13] Let $\psi_{-1} \geq 0$, $\psi_0 \geq 0$, $\{\psi_n\}$, $\{\theta_n\}$, and $\{\alpha_n\}$ be sequences of nonnegative real numbers that satisfy the following conditions:*

$$\psi_{n+1} \leq (1+\theta_n)\psi_n + (\theta_n + \alpha_n)\psi_{n-1} + \alpha_n\psi_{n-2}, \quad n \in \mathbb{N}.$$

Then

$$\psi_{n+1} \leq K \cdot \prod_{j=1}^{n}(1+2\theta_j+2\alpha_j), \quad \text{where } K = \max\{\psi_{-1}, \psi_0, \psi_1\}, \quad n \in \mathbb{N}.$$

Furthermore, if $\sum_{n=1}^{\infty}\theta_n < +\infty$ and $\sum_{n=1}^{\infty}\alpha_n < +\infty$, then $\{\psi_n\}$ is bounded.

Algorithm 3.2. *Forward–backward algorithm by double inertial extrapolations*

Initialization : *Let $0 < \gamma < \frac{1}{4}$, $\delta > 0$, $\sigma \in (0,1)$. Let $\{\alpha_n\}$ and $\{\theta_n\}$ be real nonnegative sequences. Let $u_{-1}, u_0, u_1 \in \mathcal{H}$.*

Step 1. *Given u_n, u_{n-1}, u_{n-2}, compute*

$$z_n = u_n + \theta_n(u_n - u_{n-1}) + \alpha_n(u_{n-1} - u_{n-2}),$$
$$w_n = (I + \beta_n \partial g)^{-1}(I - \beta_n \nabla f)z_n,$$

where $\beta_n = \delta\sigma^{k_n}$ and k_n is the smallest non-negative integer such that

$$\beta_n \cdot \max\left\{\|\nabla f(u_{n+1}) - \nabla f(w_n)\|, \|\nabla f(z_n) - \nabla f(w_n)\|\right\} \leq \gamma\left(\|u_{n+1} - w_n\| + \|z_n - w_n\|\right). \tag{3.1}$$

Step 2. *Calculate*

$$u_{n+1} = (I + \beta_n \partial g)^{-1}(I - \beta_n \nabla f)w_n.$$

*Then, update in **Step 1.***

Remark 3.3. *Algorithm 3.2 is based on double inertial extrapolations and the FB method. Moreover, our method combines a linesearch technique that can eliminate the need of a Lipschitz constant. This can enhance the convergence speed of the proposed algorithm compared with other methods.*

Lemma 3.4. *[16] Linesearch* (3.1) *stops after finitely many steps.*

Theorem 3.5. *Let $\{u_n\}$ be defined by **Algorithm** 3.2. If $\sum_{n=1}^{\infty}\theta_n < +\infty$, $\sum_{n=1}^{\infty}\alpha_n < +\infty$, and $\liminf_{n\to\infty}\beta_n > 0$, then*

(i) for each $v^* \in \Gamma^*$, we have $\|u_{n+1} - v^*\| \leq K \cdot \prod_{j=1}^{n}(1+2\theta_j+2\alpha_j)$, where $K = \max\{\|u_{-1} - v^*\|, \|u_0 - v^*\|, \|u_1 - v^*\|\}$;

(ii) the sequence $\{u_n\}$ converges weakly to a point in Γ^*.

Proof. (i) Let v^* be a solution in Γ^*. From $\dfrac{v^* - \text{prox}_{\beta_n g}(v^*)}{\beta_n} \in \partial g(\text{prox}_{\beta_n g}(v^*))$ and $w_n = \text{prox}_{\beta_n g}(z_n - \beta_n \nabla f(z_n))$, we can deduce that

$$\frac{z_n - w_n}{\beta_n} - \nabla f(z_n) = \frac{z_n - \text{prox}_{\beta_n g}(z_n - \beta_n \nabla f(z_n))}{\beta_n} - \nabla f(z_n) \in \partial g(w_n).$$

By the convexity of g, we obtain

$$g(v^*) - g(w_n) \geq \left\langle \frac{z_n - w_n}{\beta_n} - \nabla f(z_n), v^* - w_n \right\rangle. \tag{3.2}$$

Also, from $\dfrac{v^* - \text{prox}_{\beta_n g}(v^*)}{\beta_n} \in \partial g(\text{prox}_{\beta_n g}(v^*))$ and $u_{n+1} = \text{prox}_{\beta_n g}(w_n - \beta_n \nabla f(w_n))$, we see that

$$\frac{w_n - u_{n+1}}{\beta_n} - \nabla f(w_n) = \frac{w_n - \text{prox}_{\beta_n g}(w_n - \beta_n \nabla f(w_n))}{\beta_n} - \nabla f(w_n) \in \partial g(u_{n+1}).$$

It follows from the convexity of g that

$$g(v^*) - g(u_{n+1}) \geq \left\langle \frac{w_n - u_{n+1}}{\beta_n} - \nabla f(w_n), v^* - u_{n+1} \right\rangle. \tag{3.3}$$

By Fact 2.1, we get

$$f(v^*) - f(y) \geq \langle \nabla f(y), v^* - y \rangle, \quad y \in \mathcal{H}. \tag{3.4}$$

For any $v^* \in \mathcal{H}$, set $y = z_n$ in (3.4). Then

$$f(v^*) - f(z_n) \geq \langle \nabla f(z_n), v^* - z_n \rangle. \tag{3.5}$$

Also, if $y = w_n$ in (3.4), then we obtain

$$f(v^*) - f(w_n) \geq \langle \nabla f(w_n), v^* - w_n \rangle. \tag{3.6}$$

So, from (3.2), (3.3), (3.5), and (3.6), we get

$$\begin{aligned}
&g(v^*) - g(u_{n+1}) + g(v^*) - g(w_n) + f(v^*) - f(w_n) + f(v^*) - f(z_n) \\
&\geq \left\langle \frac{w_n - u_{n+1}}{\beta_n} - \nabla f(w_n), v^* - u_{n+1} \right\rangle + \left\langle \frac{z_n - w_n}{\beta_n} - \nabla f(z_n), v^* - w_n \right\rangle \\
&\quad + \langle \nabla f(w_n), v^* - w_n \rangle + \langle \nabla f(z_n), v^* - z_n \rangle \\
&= \frac{1}{\beta_n} \langle w_n - u_{n+1}, v^* - u_{n+1} \rangle + \langle \nabla f(w_n), u_{n+1} - v^* \rangle + \frac{1}{\beta_n} \langle z_n - w_n, v^* - w_n \rangle \\
&\quad + \langle \nabla f(z_n), w_n - v^* \rangle + \langle \nabla f(w_n), v^* - w_n \rangle + \langle \nabla f(z_n), v^* - z_n \rangle \\
&= \frac{1}{\beta_n} \langle w_n - u_{n+1}, v^* - u_{n+1} \rangle + \frac{1}{\beta_n} \langle z_n - w_n, v^* - w_n \rangle + \langle \nabla f(w_n) - \nabla f(u_{n+1}), u_{n+1} - w_n \rangle \\
&\quad + \langle \nabla f(u_{n+1}), u_{n+1} - w_n \rangle + \langle \nabla f(z_n) - f(w_n), w_n - z_n \rangle + \langle \nabla f(w_n), w_n - z_n \rangle \\
&\geq \frac{1}{\beta_n} \langle w_n - u_{n+1}, v^* - u_{n+1} \rangle + \frac{1}{\beta_n} \langle z_n - w_n, v^* - w_n \rangle + \|\nabla f(w_n) - \nabla f(u_{n+1})\| \, \|u_{n+1} - w_n\| \\
&\quad + f(u_{n+1}) + f(w_n) + \|\nabla f(z_n) - f(w_n)\| \, \|w_n - z_n\| + f(w_n) - f(z_n),
\end{aligned}$$

where the last inequality is observed from Fact 2.1. It can be deduced that

$$\frac{1}{\beta_n}[\langle w_n - u_{n+1}, u_{n+1} - v^*\rangle + \langle z_n - w_n, w_n - v^*\rangle]$$

$$\begin{aligned} &\geq (f+g)(u_{n+1}) - (f+g)(v^*) + (f+g)(w_n) - (f+g)(v^*) \\ &\quad - \|\nabla f(w_n) - \nabla f(u_{n+1})\| \, \|u_{n+1} - w_n\| - \|\nabla f(z_n) - f(w_n)\| \, \|w_n - z_n\| . \end{aligned} \tag{3.7}$$

This shows that

$$\frac{1}{\beta_n}[\langle w_n - u_{n+1}, u_{n+1} - v^*\rangle + \langle z_n - w_n, w_n - v^*\rangle]$$

$$\begin{aligned} &\geq (f+g)(u_{n+1}) - (f+g)(v^*) + (f+g)(w_n) - (f+g)(v^*) \\ &\quad - \frac{\gamma}{\beta_n} [\|w_n - u_{n+1}\| + \|w_n - z_n\|] \, \|u_{n+1} - w_n\| \\ &\quad - \frac{\gamma}{\beta_n} [\|w_n - u_{n+1}\| + \|w_n - z_n\|] \, \|w_n - z_n\| \\ &= (f+g)(u_{n+1}) - (f+g)(v^*) + (f+g)(w_n) - (f+g)(v^*) \\ &\quad - \frac{\gamma}{\beta_n} \|w_n - u_{n+1}\|^2 - \frac{\gamma}{\beta_n} \|w_n - z_n\|^2 - \frac{2\gamma}{\beta_n} \|u_{n+1} - w_n\| \, \|w_n - z_n\| . \end{aligned}$$

We know that

$$2\langle w_n - u_{n+1}, u_{n+1} - v^*\rangle = \|w_n - v^*\|^2 - \|w_n - u_{n+1}\|^2 - \|u_{n+1} - v^*\|^2,$$

and

$$2\langle z_n - w_n, w_n - v^*\rangle = \|z_n - v^*\|^2 - \|z_n - w_n\|^2 - \|w_n - v^*\|^2.$$

So, we have

$$\begin{aligned} \|z_n - v^*\|^2 - \|u_{n+1} - v^*\|^2 &\geq 2\beta_n \left[(f+g)(u_{n+1}) - (f+g)(v^*) + (f+g)(w_n) - (f+g)(v^*)\right] \\ &\quad + (1-4\gamma)\|w_n - u_{n+1}\|^2 + (1-4\gamma)\|z_n - w_n\|^2. \end{aligned} \tag{3.8}$$

Using (3.8), we obtain

$$\begin{aligned} \|u_{n+1} - v^*\|^2 &\leq \|z_n - v^*\|^2 - (1-4\gamma)\|w_n - u_{n+1}\|^2 - (1-4\gamma)\|z_n - w_n\|^2 \\ &\quad + 2\beta_n \left[(f+g)(u_{n+1}) - (f+g)(v^*) + (f+g)(w_n) - (f+g)(v^*)\right]. \end{aligned}$$

Since $0 < \gamma < \frac{1}{4}$, we get

$$\begin{aligned} \|u_{n+1} - v^*\|^2 &\leq \|z_n - v^*\|^2 - (1-4\gamma)\|w_n - u_{n+1}\|^2 - (1-4\gamma)\|z_n - w_n\|^2 \\ &\quad + 2\beta_n \left[(f+g)(u_{n+1}) - (f+g)(v^*) + (f+g)(w_n) - (f+g)(v^*)\right] \\ &\leq \|z_n - v^*\|^2. \end{aligned} \tag{3.9}$$

This gives

$$\begin{aligned} \|u_{n+1} - v^*\| &\leq \|u_n + \theta_n(u_n - u_{n-1}) + \alpha_n(u_{n-1} - u_{n-2}) - v^*\| \\ &\leq \|u_n - v^*\| + \theta_n\|u_n - u_{n-1}\| + \alpha_n\|u_{n-1} - u_{n-2}\| \end{aligned} \tag{3.10}$$

$$\begin{aligned} &\leq \|u_n - v^*\| + \theta_n [\|u_n - v^*\| + \|u_{n-1} - v^*\|] \\ &\quad +\alpha_n[\|u_{n-1} - v^*\| + \|u_{n-2} - v^*\|] \\ &= (1 + \theta_n)\|u_n - v^*\| + (\theta_n + \alpha_n)\|u_{n-1} - v^*\| + \alpha_n\|u_{n-2} - v^*\|. \end{aligned}$$

Using Lemma 3.1, we obtain

$$\|u_{n+1} - v^*\| \leq K \cdot \prod_{j=1}^{n}(1 + 2\theta_j + 2\alpha_j), \tag{3.11}$$

where $K = \max\{\|u_{-1} - v^*\|, \|u_0 - v^*\|, \|u_1 - v^*\|\}$.

(ii) Since $\sum_{n=1}^{\infty} \theta_n < +\infty$ and $\sum_{n=1}^{\infty} \alpha_n < +\infty$, $\{u_n\}$ is bounded according to Lemma 3.1 and (3.11). Moreover, we have $\sum_{n=1}^{\infty} \theta_n\|u_n - u_{n-1}\| < +\infty$ and $\sum_{n=1}^{\infty} \alpha_n\|u_{n-1} - u_{n-2}\| < +\infty$. By (3.10), we conclude that $\lim_{n\to\infty} \|u_n - v^*\|$ exists. On the other hand, we see that

$$\begin{aligned} \|z_n - v^*\|^2 &= \|u_n + \theta_n(u_n - u_{n-1}) + \alpha_n(u_{n-1} - u_{n-2}) - v^*\|^2 \\ &= \|(u_n - v^*) + \theta_n(u_n - u_{n-1}) + \alpha_n(u_{n-1} - u_{n-2})\|^2 \\ &= \|(u_n - v^*) + \theta_n(u_n - u_{n-1})\|^2 + \alpha_n^2\|u_{n-1} - u_{n-2}\|^2 \\ &\quad +2\langle u_n - v^* + \theta_n(u_n - u_{n-1}), \alpha_n(u_{n-1} - u_{n-2})\rangle \\ &= \|u_n - v^*\|^2 + \theta_n^2\|u_n - u_{n-1}\|^2 + 2\langle u_n - v^*, \theta_n(u_n - u_{n-1})\rangle \\ &\quad +\alpha_n^2\|u_{n-1} - u_{n-2}\|^2 + 2\langle u_n - v^*, \alpha_n(u_{n-1} - u_{n-2})\rangle \\ &\quad +2\langle \theta_n(u_n - u_{n-1}), \alpha_n(u_{n-1} - u_{n-2})\rangle \\ &\leq \|u_n - v^*\|^2 + \theta_n^2\|u_n - u_{n-1}\|^2 + 2\theta_n\|u_n - v^*\|\|u_n - u_{n-1}\| \\ &\quad +\alpha_n^2\|u_{n-1} - u_{n-2}\|^2 + 2\alpha_n\|u_n - v^*\|\|u_{n-1} - u_{n-2}\| \\ &\quad +2\theta_n\alpha_n\|u_n - u_{n-1}\|\|u_{n-1} - u_{n-2}\|. \end{aligned} \tag{3.12}$$

From (3.9) and (3.12), we get

$$\begin{aligned} \|u_{n+1} - v^*\|^2 &\leq \|u_n - v^*\|^2 + \theta_n^2\|u_n - u_{n-1}\|^2 + 2\theta_n\|u_n - v^*\|\|u_n - u_{n-1}\| \\ &\quad +2\alpha_n\|u_n - v^*\|\|u_{n-1} - u_{n-2}\| + 2\theta_n\alpha_n\|u_n - u_{n-1}\|\|u_{n-1} - u_{n-2}\| \\ &\quad +\alpha_n^2\|u_{n-1} - u_{n-2}\|^2 - (1 - 4\gamma)\|w_n - u_{n+1}\|^2 - (1 - 4\gamma)\|z_n - w_n\|^2 \\ &\quad -2\beta_n\left[(f + g)(u_{n+1}) - (f + g)(v^*) + (f + g)(w_n) - (f + g)(v^*)\right] \\ &\leq \|u_n - v^*\|^2 + \theta_n^2\|u_n - u_{n-1}\|^2 + 2\theta_n\|u_n - v^*\|\|u_n - u_{n-1}\| \\ &\quad +2\alpha_n\|u_n - v^*\|\|u_{n-1} - u_{n-2}\| + 2\theta_n\alpha_n\|u_n - u_{n-1}\|\|u_{n-1} - u_{n-2}\| \\ &\quad +\alpha_n^2\|u_{n-1} - u_{n-2}\|^2 - (1 - 4\gamma)\|w_n - u_{n+1}\|^2 - (1 - 4\gamma)\|z_n - w_n\|^2. \end{aligned} \tag{3.13}$$

Since $\sum_{n=1}^{\infty} \theta_n < +\infty$, $\sum_{n=1}^{\infty} \alpha_n < +\infty$, and $\lim_{n\to\infty} \|u_n - v^*\|$ exists, from (3.13) we obtain

$$\lim_{n\to\infty} \|w_n - u_{n+1}\| = 0, \tag{3.14}$$

and

$$\lim_{n\to\infty} \|z_n - w_n\| = 0. \tag{3.15}$$

We see that

$$\begin{aligned} \|z_n - u_n\| &= \|u_n + \theta_n(u_n - u_{n-1}) + \alpha_n(u_{n-1} - u_{n-2}) - u_n\| \\ &\leq \theta_n\|u_n - u_{n-1}\| + \alpha_n\|u_{n-1} - u_{n-2}\| \\ &\to 0. \end{aligned} \tag{3.16}$$

From (3.15) and (3.16), we have

$$\begin{aligned} \|w_n - u_n\| &\leq \|w_n - z_n\| + \|z_n - u_n\| \\ &\to 0. \end{aligned} \tag{3.17}$$

From (3.14) and (3.17), we get

$$\begin{aligned} \|u_{n+1} - u_n\| &\leq \|u_{n+1} - w_n\| + \|w_n - u_n\| \\ &\to 0. \end{aligned}$$

Let a be a weak cluster point of $\{u_n\}$. Then there exists a subsequence $\{u_{n_j}\}$ of $\{u_n\}$ such that $u_{n_j} \rightharpoonup a \in \mathcal{H}$. Moreover, we get $u_{n_j+1} \rightharpoonup a$. Next, we show that $a \in \Gamma^*$. Let $(p, q) \in \text{Graph}(\nabla f + \partial g)$, that is, $q - \nabla f(p) \in \partial g(p)$. Since $\{u_{n_j}\}$ is bounded and $\lim_{j\to\infty} \|u_{n_j+1} - w_{n_j}\| = 0$, by (A2), we have

$$\lim_{j\to\infty} \|\nabla f(u_{n_j+1}) - \nabla f(w_{n_j})\| = 0. \tag{3.18}$$

From

$$u_{n_j+1} = (I + \beta_{n_j}\partial g)^{-1}(I - \beta_{n_j}\nabla f)w_{n_j},$$

it follows that

$$(I - \beta_{n_j}\nabla f)w_{n_j} \in (I + \beta_{n_j}\partial g)u_{n_j+1}.$$

Hence,

$$\frac{1}{\beta_{n_j}}(w_{n_j} - u_{n_j+1} - \beta_{n_j}\nabla f(w_{n_j})) \in \partial g(u_{n_j+1}).$$

By the monotonicity of ∂g, we obtain

$$\left\langle p - u_{n_j+1}, q - \nabla f(p) - \frac{1}{\beta_{n_j}}(w_{n_j} - u_{n_j+1} - \beta_{n_j}\nabla f(w_{n_j}))\right\rangle \geq 0.$$

Consequently, we have

$$\begin{aligned} \langle p - u_{n_j+1}, q\rangle &\geq \left\langle p - u_{n_j+1}, \nabla f(p) + \frac{1}{\beta_{n_j}}(w_{n_j} - u_{n_j+1} - \beta_{n_j}\nabla f(w_{n_j}))\right\rangle \\ &= \langle p - u_{n_j+1}, \nabla f(p) - \nabla f(w_{n_j})\rangle + \left\langle p - u_{n_j+1}, \frac{1}{\beta_{n_j}}(w_{n_j} - u_{n_j+1})\right\rangle \\ &= \langle p - u_{n_j+1}, \nabla f(p) - \nabla f(u_{n_j+1})\rangle + \langle p - u_{n_j+1}, \nabla f(u_{n_j+1}) - \nabla f(w_{n_j})\rangle \end{aligned}$$

$$
\begin{aligned}
&+\left\langle p-u_{n_j+1}, \frac{1}{\beta_{n_j}}(w_{n_j}-u_{n_j+1})\right\rangle \\
\geq\ & \langle p-u_{n_j+1}, \nabla f(u_{n_j+1})-\nabla f(w_{n_j})\rangle \\
&+\left\langle p-u_{n_j+1}, \frac{1}{\beta_{n_j}}(w_{n_j}-u_{n_j+1})\right\rangle. \qquad (3.19)
\end{aligned}
$$

From (3.18), taking the limit as $j \to \infty$ in (3.19), we have

$$\langle p-a, q\rangle \geq 0.$$

Hence, by the maximal monotonicity of $\nabla f + \partial g$ and Lemma 2.3, we have $0 \in (\nabla f + \partial g)(a)$, i.e., $a \in (\nabla f + \partial g)^{-1}(0) = \Gamma^*$. By Lemma 2.4, it is concluded that $\{u_n\}$ converges weakly to a point in Γ^*. Thus, we complete the proof. □

4. Numerical experiments

4.1. Image restoration

Next, we present numerical experiments to evaluate the performance of Algorithm 3.2 and to compare it with Iteration (1.5) and Iteration (1.6).

We consider the problem of image restoration, which can be formulated by:

$$b = Dx + \epsilon, \qquad (4.1)$$

where an original image is denoted as $x \in \mathbb{R}^{N\times 1}$, a degraded image is represented by $b \in \mathbb{R}^{M\times 1}$, a noise term is given by $\epsilon \in \mathbb{R}^{M\times 1}$, and $D \in \mathbb{R}^{M\times N}$ is a blurring matrix. It is worth noting that problem (4.1) can be equivalently formulated as the convex minimization model

$$\min_{x\in\mathbb{R}^N} \frac{1}{2}\|b-Dx\|_2^2 + \eta\|x\|_1,$$

where $\eta > 0$. We set $\nabla f(x) = D^t(Dx-b)$ and $\partial g(x) = \partial(\eta\|x\|_1)$. To measure the quality of the restored images, we use the peak signal-to-noise ratio (PSNR) [28]

$$PSNR := 20\log_{10}\left(\frac{255^2}{\|x^n - x\|_2^2}\right),$$

and the structural similarity index measure (SSIM) [30]

$$SSIM := \frac{(2\theta_x\theta_{x_r}+c_1)(2\sigma_{xx_r}+c_2)}{(\theta_x^2+\theta_{x_r}^2+c_1)(\sigma_x^2+\sigma_{x_r}^2+c_2)}.$$

In these equations, all parameters are defined as in [28, 30].

In this example, the initial points u_{-1}, u_0, and u_1 are the blurred image with the stopping criterion at the 500^{th} iteration. The regularization parameter is chosen as $\eta = 10^{-5}$. All parameters are selected as in Table 1 with $s_{n+1} = \dfrac{1+\sqrt{1+4s_n^2}}{2}$.

Table 1. Numerical comparison of PSNR and SSIM values of each algorithm.

	θ_n	α_n	β_n	δ	σ	γ	s_1
Iteration (1.5)	$\frac{s_n - 1}{s_{n+1}}$	-	0.5	-	-	-	1
Iteration (1.6)	-	-	-	1.5	0.75	0.75	-
Algorithm 3.2	if $1 \leq n \leq 200$, $\theta_n = \frac{s_n - 1}{s_{n+1}}$ else, $\theta_n = \frac{1}{(3n+1)^2}$	$\frac{1}{(10n+1)^2}$	-	1.4	0.86	0.24	1

First, we apply the toolbox in our test and degrade the original image by motion blur with an angle of 90 and a motion length of 25. After that, we present the restored images at 500 iterations for each algorithm in Table 2 and show them in Figure 1. Moreover, the performance measurements for PSNR and SSIM are illustrated in Figure 2.

Table 2 shows that the PSNR value of our algorithm is 42.9128, which is higher than other iterations. This means that images recovered using our algorithm will have image quality closer to the original than images recovered using other algorithms. For SSIM, it is recovered perfectly if the value is 1. In this example, the SSIM value is found to be 0.9834, which is also higher than other iterations. So, our proposed method has the best performance among other iterations.

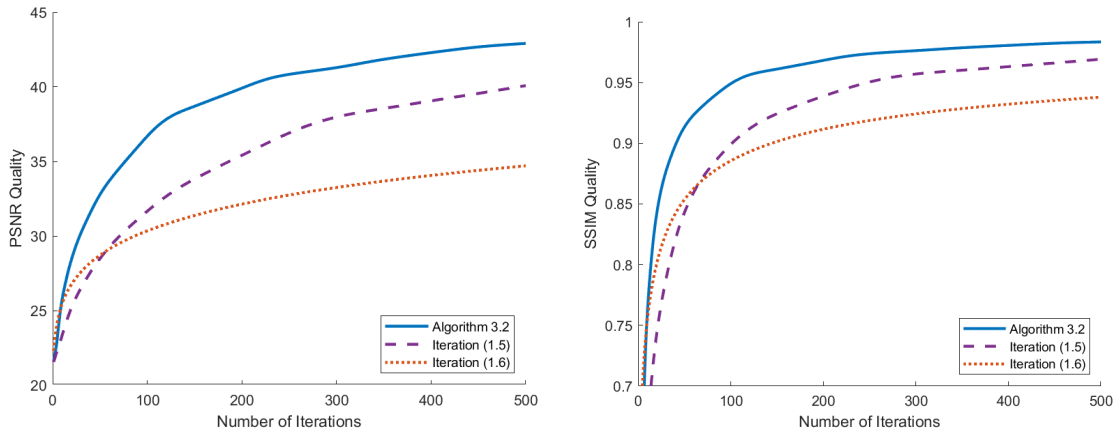
Table 2. Numerical comparison of PSNR and SSIM values of each iteration.

	Iteration (1.5)	Iteration (1.6)	Algorithm 3.2
PSNR	40.0807	34.7060	42.9128
SSIM	0.9692	0.9379	0.9834

(a) Cat **(b)** Observed image **(c)** Algorithm 3.2

(d) Iteration (1.5) **(e)** Iteration (1.6)

Figure 1. The Cat image (909×608) reconstructed by each iteration.



(a) PSNR Quality **(b)** SSIM Quality

Figure 2. Measuring the quality of the Cat image.

4.2. Data classification

In this part, we focus on the data classification problem. Several real-world problems considered in this work can be cast in the framework of a convex minimization problem. We will explain the process of formulating machine learning (ML) problems, particularly classification problems. Furthermore, we present the evidence which indicates that the introduced methods outperform some of the approaches in the existing literature.

Before delving into our work, we will briefly overview an extreme learning machine (ELM) applied to data classification problems. Let $P := \{(u_n, z_n) : u_n \in \mathbb{R}^n, z_n \in \mathbb{R}^m, n = 1, 2, ..., N\}$ represent a training set comprising N distinct samples, where u_n is input training data, and z_n is a training target. For any single hidden layer of ELM, the output at the i-th hidden node is

$$h_i(u) = \mathcal{W}(\langle a_i, u\rangle + b_i),$$

where $\mathcal{W}$ denotes an activation function, and a_i and b_i represent the parameters for the weight and bias at the i-th hidden node, respectively. The output function of the ELM for single-hidden layer feedforward neural networks (SLFNs) with M hidden nodes is

$$O_n = \sum_{i=1}^{M} \omega_i h_i(u_n),$$

where ω_i stands for the optimal output weight at the i-th hidden node. The matrix $\mathcal{A}$, corresponding to the output of the hidden layer, is defined as

$$\mathcal{A} = \begin{bmatrix} \mathcal{W}(\langle a_1, u_1\rangle + b_1) & \cdots & \mathcal{W}(\langle a_M, u_1\rangle + b_M) \\ \vdots & \ddots & \vdots \\ \mathcal{W}(\langle a_1, u_N\rangle + b_1) & \cdots & \mathcal{W}(\langle a_M, u_N\rangle + b_M) \end{bmatrix}.$$

The primary objective of ELM is to determine the optimal output weight $\omega = [\omega_1^T, \omega_2^T, ..., \omega_M^T]^T$ such that $\mathcal{A}\omega = \mathcal{T}$, where $\mathcal{T} = [t_1^T, t_2^T, ..., t_N^T]^T$ represents the training target data. In certain situations,

finding $\omega = \mathcal{A}^*\mathcal{T}$, where $\mathcal{A}^*$ is the Moore-Penrose generalized inverse of $\mathcal{A}$, can be challenging, especially when the matrix $\mathcal{A}$ is non-existent. Consequently, obtaining such a solution ω through convex minimization can overcome this difficulty.

In this part, we conduct experiments on the classification problem, which can be formulated as the convex minimization problem

$$\min_{\omega\in\mathbb{R}^M}\left\{\frac{1}{2}\|\mathcal{A}\omega-\mathcal{T}\|_2^2+\beta\|\omega\|_1\right\}, \tag{4.2}$$

where β is a regularization parameter. This problem is called the regularization of least squares problem by L_1, or the least absolute shrinkage and selection operator (LASSO).

We employ the binary cross-entropy loss function [12] with sigmoid activation function defined by

$$Loss = -\frac{1}{\rho}\sum_{j=1}^{\rho} u_j \log \hat{u}_j + (1-u_j)\log(1-\hat{u}_j), \tag{4.3}$$

where ρ is the number of scalar values in the model output, u_j is the corresponding target value, and $\hat{u}_j$ is the j-th scalar values in the model output.

The precision and recall can justify the performance evaluation in classification. Recall, also known as the true positive rate, measures the accuracy of predictions in positive classes and represents the percentage of correctly predicted positive observations. Accuracy and F1-score can be calculated using the following equations [10]:

$$\text{Precision (Pre)} = \frac{TP}{TP+FP}\times 100\%,$$

$$\text{Recall (Rec)} = \frac{TP}{TP+FN}\times 100\%,$$

$$\text{Accuracy (Acc)} = \frac{TP+TN}{TP+FP+TN+FN}\times 100\%,$$

$$\text{F1-score} = \frac{2\times(\text{Pre}\times\text{Rec})}{\text{Pre}+\text{Rec}}\times 100\%,$$

where a confusion matrix for original and predicted classes are indicated in terms of TP =True Positive, TN =True Negative, FP =False Positive and FN =False Negative.

First, we will mention Chronic Kidney Disease (CKD), which is among the rapidly growing diseases, and which poses a significant threat to health and life. CKD is characterized by the gradual failure of the kidneys to perform their essential functions, such as blood filtration. Given that kidney damage occurs progressively over an extended period, it qualifies as a "chronic" disease. The functionality of the urinary organs is also compromised. As the disease advances, it may accumulate waste in the blood, giving rise to additional health complications. Associated symptoms include hypertension, a decline in blood count levels, weakened bones, nerve damage, and, ultimately, an increased risk of heart and blood vessel diseases.

In this numerical experiment, the dataset is sourced from the UCI Dataset for CKD, encompassing sample data from 400 patients. The dataset comprises 13 attributes and 1 class attribute. The considered features include blood pressure, specific gravity, albumin, sugar, red blood cells, blood urea, serum creatinine, sodium, potassium, hemoglobin, white blood cell count, red blood cell count,

and hypertension. The visualization of the CKD dataset is presented in Table 3. The output classes, namely 'Diseased' (meaning the patient has CKD) and 'Not Diseased' (meaning the patient is healthy), consist of 250 and 150 instances, respectively. For implementing machine learning algorithms, we used 280 data for training and 120 data for testing, as illustrated in Table 4.

Table 3. Overview of all attributes used in training the models.

Attributes	Mean	SD	Min	Max
Blood pressure	0.2035	0.1037	0	1
Specific gravity	1.0177	0.0054	1.005	1.025
Albumin	1.015	1.2723	0	5
Sugar	0.395	1.0400	0	5
Red blood cell	0.8825	0.3224	0	1
Blood Urea	0.1435	0.1265	0	1
Serum creatinine	0.0353	0.0743	0	1
Sodium	0.8393	0.0581	0	1
Pottasium	0.0478	0.0634	0	1
Hemoglobin	12.5269	2.7162	3.1	17.8
white blood cell count	0.2565	0.1043	0	1
Red blood cell count	4.7083	0.8403	2.1	8
Hypertension	0.3694	0.4820	0	1

(SD: Standard deviation)

Table 4. Class distribution in dataset.

	Class	Number of instances	Training dataset	Testing dataset
1	Diseased	250	175	75
2	Not Diseased	150	105	45

Specifically, we employ our algorithm to optimize weight parameters in training data for machine learning, and we focus on the extreme machine learning (ELM). The computational process begins with the activation function set as sigmoid and hidden nodes L set to 180. We compare the outcomes of our algorithm with those obtained from Iteration (1.5) and Iteration (1.6). The inertial points are set as $u_{-1} = (1, 1, 1, ..., 1)$, $u_0 = (1, 1, 1, ..., 1)$, and $u_1 = (0, 0, 0, ..., 0)$. The parameters for each algorithm used in the comparison are as follows:

- Iteration (1.5): $s_1 = 1$;
- Iteration (1.6): $\delta = 1.1$, $\sigma = 0.24$ and $\gamma = 0.11$;
- Algorithm 3.2: if $0 \leq n \leq 300$, we set $\theta_n = \dfrac{s_n - 1}{s_{n+1}}$, $s_{n+1} = \dfrac{1 + \sqrt{1 + 4s_n^2}}{2}$, else $\theta_n = \dfrac{1}{(10n+1)^2}$, $\alpha_n = \dfrac{1}{(5n+1)^2}$, $s_1 = 1$, $\delta = 1.4$, $\sigma = 0.87$, and $\gamma = 0.24$.

We compare the performance of each algorithm for the number of iterations at 500 and 1000. The

numerical results are presented in Tables 5 and 6. It shows that our algorithm, Algorithm 3.2, performs the best in terms of accuracy at both 500 and 1000 iterations.

Table 5. The performance of each algorithm for 500 iterations.

Algorithms	Pre	Rec	F1-score	Acc
Iteration (1.5)	80	97.78	88	90
Iteration (1.6)	81.48	97.78	88.89	90.83
Algorithm 3.2	83.33	100	90.91	92.5

Table 6. The performance of each algorithm for 1000 iterations.

Algorithms	Pre	Rec	F1-score	Acc
Iteration (1.5)	80	97.78	88	90
Iteration (1.6)	83.33	100	90.91	92.50
Algorithm 3.2	84.91	100	91.84	93.33

Next, we show the performance with the highest accuracy of each algorithm for prediction in terms of iterations. The comparison is shown in Table 7.

Table 7. The performance of each algorithm with training accuracy > 91 and testing accuracy > 96.

Algorithms	Iterations	Pre	Rec	F1-score	Acc
Iteration (1.5)	60	97.67	93.33	95.45	96.67
Iteration (1.6)	56	97.67	93.33	95.45	96.67
Algorithm 3.2	19	97.67	93.33	95.45	96.67

From Table 7, we observe that Algorithm 3.2 has fewer iterations than Iteration (1.5) and Iteration (1.6) with the stopping criteria as training accuracy > 91 and testing accuracy > 96. Moreover, the proposed algorithm provides more accuracy than other algorithms. This shows that our algorithm has the highest probability of correctly classifying the CKD dataset compared to other algorithms.

Next, the convergence behavior of the accuracy and the loss of training data and testing data for overfitting of Algorithm 3.2 are shown graphs in Figure 3 and Figure 4, respectively. We can see that the graphs have loss of training data and testing data.

In Figures 3 and 4, it shows the convergence behaviour of accuracy and loss of Algorithm 3.2. We see that graphs have a good fitting model which means that Algorithm 3.2 suitably learns the training dataset classification.

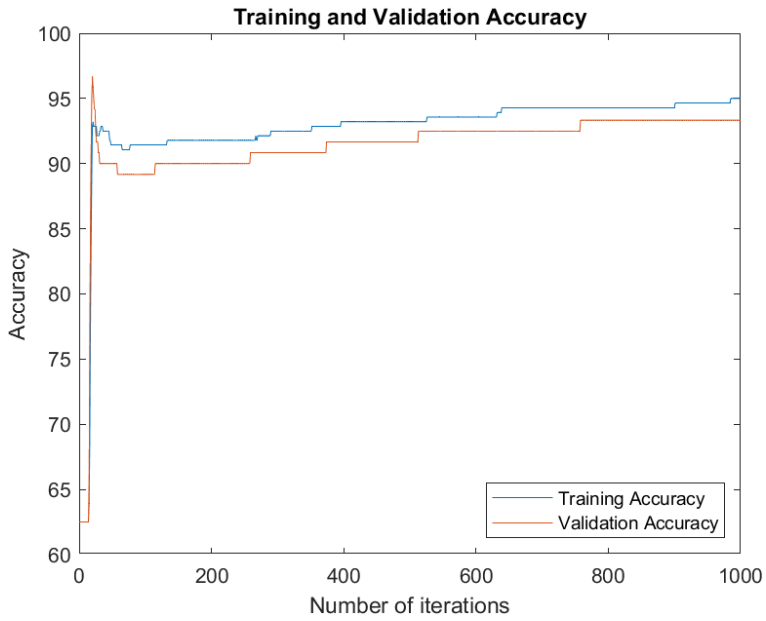


Figure 3. Accuracy of Algorithm 3.2.

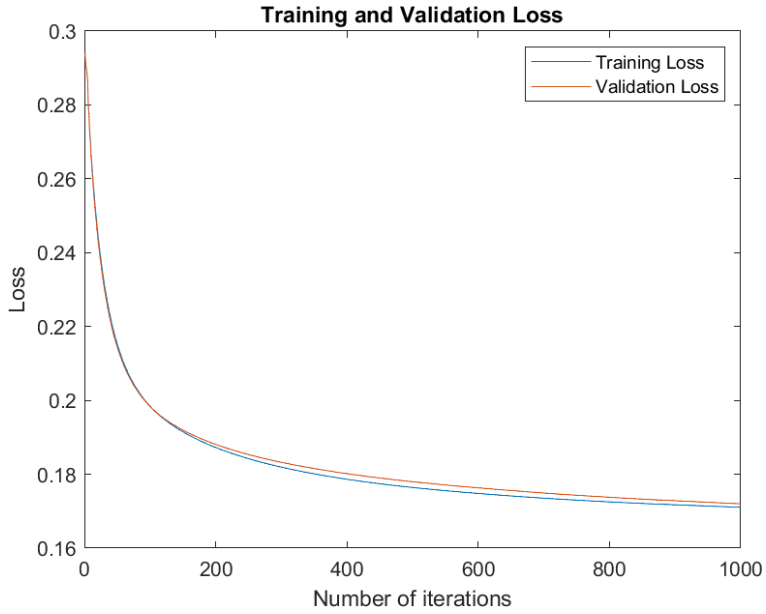


Figure 4. Loss of Algorithm 3.2.

5. Conclusions

In this study, we presented a recent FB algorithm for convex minimization problems. Our approach incorporates a linesearch technique to eliminate the need for explicit Lipschitz assumptions and employs double inertial extrapolations to accelerate the algorithm's convergence. Furthermore, we established a weak convergence theorem under reasonable assumptions. Additionally, we conducted numerical tests in image restoration and data classification, demonstrating the superiority of our

algorithm over existing methods in the literature. The experimental results validate the effectiveness and improved performance of our proposed approach.

Author contributions

Prasit Cholamjiak: Conceptualization, Supervision, Writing–review & editing; Uamporn Witthayarat: Writing–review & editing; Suparat Kesornprom: Data curation, Formal Analysis; Papatsara Inkrong: Software; Writing–original draft.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors sincerely thank the anonymous reviewers for their suggestions that improve the manuscript substantially. Suparat Kesornprom has received funding support from Fundamental Fund 2024 (FF030/2567), Chiang Mai University, Chiang Mai, Thailand. Uamporn Witthayarat has received funding support from School of Science, University of Phayao, a grant no. PBTSC65001. Moreover, Prasit Cholamjiak was supported by the National Research Council of Thailand under grant no. N41A640094, University of Phayao and Thailand Science Research and Innovation Fund (Fundamental Fund 2024).

Data Availability

The dataset used in this research is publicly available at https://www.kaggle.com/datasets/abhia 1999/chronic-kidney-disease.

Conflict of interest

The authors declare no conflict of interest.

References

1. H. H. Bauschke, P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, New York: Springer, 2013. https://doi.org/10.1007/978-1-4419-9467-7
2. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.*, **2** (2009), 183–202. https://doi.org/10.1137/080716542
3. J. Y. Bello Cruz, T. T. A. Nghia, On the convergence of the forward–backward splitting method with linesearches, *Optim. Method. Softw.*, **31** (2016), 1209–1238. https://doi.org/10.1080/10556788.2016.1214959
4. R. S. Burachik , A. N. Iusem, *Set-valued mappings and enlargements of monotone operators*, New York: Springer, 2008. https://doi.org/10.1007/978-0-387-69757-4

5. P. L. Combettes, L. E. Glaudin, Quasi-nonexpansive iterations on the affine hull of orbits: from Mann's mean value algorithm to inertial methods, *SIAM J. Optim.*, **27** (2017), 2356–2380. https://doi.org/10.1137/17M112806X
6. P. L. Combettes, V. R. Wajs, Signal recovery by proximal forward-backward splitting, *Multiscale Model. Sim.*, **4** (2005), 1168–1200. https://doi.org/10.1137/050626090
7. Q. L. Dong, J. Z. Huang, X. H. Li, Y. J. Cho, T. M. Rassias, MiKM: multi-step inertial Krasnosel'skiĭ–Mann algorithm and its applications, *J. Glob. Optim.*, **73** (2019), 801–824. https://doi.org/10.1007/s10898-018-0727-x
8. J. C. Dunn, Convexity, monotonicity, and gradient processes in Hilbert space, *J. Math. Anal. Appl.*, **53** (1976), 145–158. https://doi.org/10.1016/0022-247X(76)90152-9
9. O. Guler, On the convergence of the proximal point algorithm for convex minimization, *SIAM J. Control Optim.*, **29** (1991), 403–419. https://doi.org/10.1137/0329022
10. J. Han, J. Pei, H. Tong, *Data mining: concepts and techniques*, 4 Eds., Morgan kaufmann, 2022.
11. A. Hanjing, S. Suantai, A fast image restoration algorithm based on a fixed point and optimization method, *Mathematics*, **8** (2020), 378. https://doi.org/10.3390/math8030378
12. Y. Ho, S. Wookey, The real-world-weight cross-entropy loss function: modeling the costs of mislabeling, *IEEE Access*, **8** (2019), 4806–4813. https://doi.org/10.1109/ACCESS.2019.2962617
13. P. Inkrong, P. Cholamjiak, Modified proximal gradient methods involving double inertial extrapolations for monotone inclusion, *Math. Method. Appl. Sci.*, in press. https://doi.org/10.1002/mma.10159
14. P. Inkrong, P. Cholamjiak, On multi-inertial extrapolations and forward-backward-forward algorithms, *Carpathian J. Math.*, **40** (2024), 293–305.
15. L. O. Jolaoso, Y. Shehu, J. C. Yao, R. Xu, double inertial parameters forward-backward splitting method: applications to compressed sensing, image processing, and SCAD penalty problems, *J. Nonlinear Var. Anal.*, **7** (2023), 627–646. https://doi.org/10.23952/jnva.7.2023.4.10
16. K. Kankam, N. Pholasa, P. Cholamjiak, Hybrid forward-backward algorithms using line search rule for minimization problem, *Thai J. Math.*, **17** (2019), 607–625.
17. K. Kankam, N. Pholasa, P. Cholamjiak, On convergence and complexity of the modified forward-backward method involving new line searches for convex minimization, *Math. Method. Appl. Sci.*, **42** (2019), 1352–1362. https://doi.org/10.1002/mma.5420
18. J. Liang, Convergence rates of first-order operator splitting methods, PhD thesis, Normandie Université, GREYC CNRS UMR 6072, 2016.
19. P. Martinet, Régularisation d'inéquations variationelles par approximations successives, *Rev. fr. autom. inform. rech. opér.*, **4** (1970), 154–159.
20. Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, *Dokl. Akad. Nauk SSSR*, **269** (1983), 543.
21. Z. Opial, Weak convergence of the sequence of successive approximations for nonexpansive mappings, *Bull. Amer. Math. Soc.*, **73** (1967), 591–597. https://doi.org/10.1090/S0002-9904-1967-11761-0

22. N. Parikh, S. Boyd, Proximal algorithms, *Foundations and Trends® in Optimization*, **1** (2014), 127–239. http://doi.org/10.1561/2400000003

23. B. T. Polyak, *Introduction to optimization*, New York: Optimization Software Inc., Publications Division, 1987.

24. B. T. Polyak, Some methods of speeding up the convergence of iteration methods, *USSR Comput. Math. Math. Phys.*, **4** (1964), 1–17. https://doi.org/10.1016/0041-5553(64)90137-5

25. C. Poon, J. Liang, Trajectory of alternating direction method of multipliers and adaptive acceleration, *Advances in neural information processing systems 32 (NeurIPS 2019)*, Vancouver, Canada, 2019, 7325–7333.

26. R. T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM J. Control Optim.*, **14** (1976), 877–898. https://doi.org/10.1137/0314056

27. W. Takahashi, *Introduction to nonlinear and convex analysis*, Yokohama Publishers, 2009.

28. K. H. Thung, P. Raveendran, A survey of image quality measures, *2009 international conference for technical postgraduates (TECHPOS)*, Kuala Lumpur, Malaysia, 2009, 1–4. https://doi.org/10.1109/TECHPOS.2009.5412098

29. C. Wang, N. Xiu, Convergence of the gradient projection method for generalized convex minimization, *Comput. Optim. Appl.*, **16** (2000), 111–120. https://doi.org/10.1023/A:1008714607737

30. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.*, **13** (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

31. H. K. Xu, Averaged mappings and the gradient-projection algorithm, *J. Optim. Theory Appl.*, **150** (2011), 360–378. https://doi.org/10.1007/s10957-011-9837-z

AIMS Press