



PARALLEL DOUBLE INERTIAL MANN-TYPE ALGORITHM FOR BLIND IMAGE DEBLURRING

PAPATSARA INKRONG¹, PAPINWICH PAIMSANG¹, PRASIT CHOLAMJIAK¹, YANN SAVOYE²,
WATCHARAPORN CHOLAMJIAK^{1,*}

¹School of Science, University of Phayao, Phayao 56000, Thailand

²School of Computing and Mathematical Sciences, University of Leicester, United Kingdom

Abstract. The Mann algorithm has been extensively studied as one of the most fundamental iterative schemes designed to find a fixed point of an averaged operator. In this paper, we aim to develop a new parallel algorithm improving the Mann algorithm with double inertial extrapolations to find a common fixed point of a finite family of nonexpansive mappings. Our proposed algorithm allows iterations to be carried out simultaneously. We prove the weak convergence under suitable conditions, and we give an example in infinite-dimensional spaces. Finally, we apply our algorithm in the context of image restoration to deblur images without prior knowledge of the blurring operator.

Keywords. Deblur image; Mann iteration; Nonexpansive mapping; Inertial extrapolation; Weak convergence.

2020 MSC. 47H09, 90C30, 65F10.

1. INTRODUCTION

Restoration problem. The image restoration problem is to remove blurring and noise in a degraded image. The blur is generally modeled by using a system of linear equations. This celebrated problem is now widespread in various fields such as forensic research, medical imaging, photographic enhancement, and astronomical imaging. A common difficulty in image restoration is to restore images when the blurring operator involved is unknown. Traditional restoration procedures often require prior knowledge of the blurring process. In real-world situations, the blurring operator involved is, however, not available and has to be estimated.

Deblurring photographs requires a technique flexible enough to deal with ambiguities regarding the degradation process. To handle uncertainties in the data, optimization procedures are always needed to find the best approximation of the original image. The blur corresponds to the

*Corresponding author.

E-mail address: watcharaporn.ch@up.ac.th (W. Cholamjiak).

Received March 7, 2025; Accepted October 1, 2025.

presence of an area in an image lacking its sharpness. This degeneration occurs under several well-identified factors such as camera physical properties, subject movement, poor focusing, and narrow aperture that results in a shallow depth of field. Recently, efforts were made to overcome or reduce the difficulty of image restoration without prior knowledge of the blurring operator; see, e.g., [5, 19, 32, 39]. The four most common forms of blur effects in digital imaging are the average, out-of-focus, gaussian, and motion blur. In this paper, we only focus on the motion blur generated by relative movement between the camera and the subject producing stretching effects along the direction of motion.

Motivation and objectives. In this paper, we are motivated by recent interest in using Mann’s iteration process at the heart of an embarrassingly parallel algorithm to speed up the numerical solving process for image blind deblurring. Our work builds upon this process known to converge weakly in some class of infinite-dimensional Banach spaces. The purpose of this paper is to devise the Mann-type algorithm for finding a common fixed point for a finite family of nonexpansive mappings. In particular, the finite (namely countable) family of nonexpansive mappings is well-adapted to encompass a generated collection of motion blurs containing commonly accepted potential directions (and length) for the unknown motion blur without the estimation of unknown parameters precisely (as an initial guess).

In our algorithm, we combine Mann’s iteration process with double inertial extrapolations to further speed up the solving process while ensuring its convergence along iterations. The theoretical objective of this paper is to establish the weak convergence of the proposed method to find a common fixed point under suitable conditions and restrictions. Also, the objective is to provide an example in an infinite dimensional space to approximate the solution of the inclusion problem. Finally, the targeted application of our proposed algorithm is to perform image restoration without prior knowledge of the blurring operator. This real-world problem is recognized as a hard problem and classically known as blind image deblurring. In this paper, our goal is to improve the readability of blurry characters so that the blurred characters are visible enough to be detected by an optical character recognition system.

Contribution. Our main contribution is a new fixed point iteration method integrating Mann iteration with double inertial extrapolations. Our algorithm is embarrassingly parallel to be implemented on GPU. We establish the weak convergence of this method for finding a common fixed point for a finite family of nonexpansive mappings.

2. TECHNICAL BACKGROUND

In this section, we present several technical background related to inertial techniques as acceleration methods such as the fixed point theory (*i.e.* nonexpansive mappings and fixed point sets), iterative schemes (*i.e.* iteration process, Mann iteration, inertial terms and inertial extrapolation) and their generalizations (*i.e.* double inertial, proximal point algorithm).

Fixed point theory. The fixed point theory is an important optimization theory which is widely recognized across many fields such as medicine, economics, image processing, data classification, machine learning, and signal processing, and so on; see, e.g., [9, 10, 17, 28, 30].

Nonexpansive mapping. The Browder’s theorem [3] demonstrates the existence of a fixed point for a nonexpansive mapping. In this work, we consider a real Hilbert space $\mathcal{H}$ with an inner product $\langle \cdot, \cdot \rangle$ and the corresponding norm $\| \cdot \|$. Let $\mathcal{C}$ be a nonempty, convex, and

closed subset of $\mathcal{H}$. A mapping $\mathcal{S} : \mathcal{C} \rightarrow \mathcal{C}$ is called nonexpansive if, for each $u, w \in \mathcal{C}$, $\|\mathcal{S}u - \mathcal{S}w\| \leq \|u - w\|$.

In this paper, we study the celebrated fixed point problem that aims to find a point u such that $u = \mathcal{S}_i u$, where $\{\mathcal{S}_i\}_{i=1}^N$ is a finite family of nonexpansive mappings on $\mathcal{H}$. A point u is called a common fixed point of $\mathcal{S}_i$ if $u \in \bigcap_{i=1}^N \mathcal{F}(\mathcal{S}_i)$.

Fixed point set. For any mapping $\mathcal{S} : \mathcal{H} \rightarrow \mathcal{H}$, the fixed point set of $\mathcal{S}$ is denoted by $\mathcal{F}(\mathcal{S})$ and given by $\mathcal{F}(\mathcal{S}) := \{u \in \mathcal{H} : u = \mathcal{S}u\}$.

Iteration process. The initial point of any iteration scheme is denoted by u_0 . Picard [26] introduced the iteration process, as follows:

$$u_{k+1} = \mathcal{S}u_k, \quad k \geq 0.$$

Mann iteration. The Mann iteration [22] is an averaged process (by nature) generating a sequence $\{u_k\}$ recursively by:

$$u_{k+1} = \beta_k u_k + (1 - \beta_k) \mathcal{S}u_k, \quad k \geq 0,$$

where the initial u_0 is selected arbitrarily, and the sequence $\{\beta_k\}$ lies within the interval $(0, 1)$.

Inertial and momentum terms. In the context of the gradient algorithm, inertial terms were introduced by Polyak [27] to speed up the convergence of iterative algorithms. Also introduced by Polyak, the heavy ball method introduces a momentum term $\alpha_k(u_k - u_{k-1})$, given the starting points $u_0 = u_{-1} \in \mathbb{R}^m$,

$$u_{k+1} = u_k + \alpha_k(u_k - u_{k-1}) - \lambda_k \nabla g(u_k), \quad k \geq 0,$$

where $\{\alpha_k\} \in [0, \infty)$ and $\lambda_k > 0$ is the step-size parameter.

Inertial extrapolation. Mainge [23] modified the Mann algorithm with an inertial extrapolation by introducing the following algorithm. Let $u_{-1}, u_0 \in \mathcal{H}$,

$$\begin{aligned} w_k &= u_k + \alpha_k(u_k - u_{k-1}), \\ u_{k+1} &= \beta_k w_k + (1 - \beta_k) \mathcal{S}w_k, \quad k \geq 0, \end{aligned}$$

where $\{\alpha_k\} \in [0, \infty)$ and $\{\beta_k\} \subset [0, 1]$. It is worth noting that the iterative sequence $\{u_k\}$ defined by this algorithm converges weakly to a fixed point of $\mathcal{S}$ under certain mild assumptions.

Generalizing inertial methods. In recent years, the adoption of inertial techniques as acceleration methods has attracted significant attention in research. Various fast iterative algorithms incorporating inertial terms were developed. For instance, the inertial technique [27] was generalized by Iyiola and Shehu [16] with the introduction of the double inertial and proximal point algorithms as follows. Let $u_{-1}, u_0, w_0 \in \mathcal{H}$, and

$$\begin{aligned} u_{k+1} &= J_{\delta}^{\mathcal{A}} w_k, \\ w_{k+1} &= u_{k+1} + \alpha(u_{k+1} - u_k) + \lambda(u_k - u_{k-1}), \quad k \geq 0, \end{aligned}$$

where $\mathcal{A} : \mathcal{H} \rightarrow 2^{\mathcal{H}}$ is a maximal monotone operator and the parameters of α, λ and δ satisfying conditions as presented in [16].

Other improvements include the inertial extragradient methods introduced by Dong [11], Inertial Mann algorithms [12, 29, 33], inertial extragradient algorithms [14, 38], inertial forward-backward algorithms [1, 37], and the fast iterative shrinkage-thresholding algorithm (FISTA) [4]. These algorithms were proven to be valuable for both theoretical and numerical evaluations,

with applications in areas such as image processing, signal recovery, and machine learning. For further insights, we refer the readers to [2, 7, 8, 20, 31].

Parallel inertial methods. Parallel algorithms were used to speed up the computation by executing multiple tasks across various processing elements simultaneously. This class of algorithms is ideal for solving large-scale and data-intensive problems encountered in various fields such as computer science, engineering, and bioinformatics. Recently, Jun-On *et al.* [18] modified inertial extrapolation and parallel monotone hybrid method to find a common fixed point for a finite family of nonexpansive mappings in $\mathcal{H}$. Starting with u_{-1}, u_0 , the parallel algorithm is written as follows:

$$\begin{aligned} v_k &= u_k + \alpha_k (u_k - u_{k-1}), \\ w_{i,k} &= (1 - \beta_{i,k}) v_k + \beta_{i,k} \mathcal{S}_i v_k, \\ y_{i,k} &= (1 - \gamma_{i,k}) \mathcal{S}_i v_k + \gamma_{i,k} \mathcal{S}_i w_{i,k}, \\ u_{k+1} &= \arg \max_i \{ \|y_{i,k} - v_k\| : i = 1, \dots, N \}, \quad \text{for } k \geq 0, \end{aligned} \tag{2.1}$$

where $\{\mathcal{S}_i\}_{i=1}^N$ is nonexpansive mappings and the sequences $\{\alpha_k\}$, $\{\beta_{i,k}\}$ and $\{\gamma_{i,k}\}$ satisfy the conditions that are as presented in [18]. Applications in signal recovery of parallel methods include scenarios with unknown noise types using various blurred matrices and noise for the LASSO problem. Some results involving the parallel method for solving the fixed point problem can be found in [5, 6, 13, 34, 35, 36].

3. PRELIMINARIES

We recall specific definitions and lemmas that are fundamental in our work to derive our proposed proof.

An operator $\mathcal{P}_{\mathcal{C}}$ is said to be metric projection of $\mathcal{H}$ onto nonempty, convex, and closed subset $\mathcal{C}$ if, for every $u \in \mathcal{H}$, there exists a unique nearest point in $\mathcal{C}$ denoted by $\mathcal{P}_{\mathcal{C}}u$ such that $\|u - \mathcal{P}_{\mathcal{C}}u\| \leq \|u - w\|$ for all $w \in \mathcal{C}$. Let $\mathcal{H}$ be a real Hilbert space. It is known the following statements hold:

- (1) $\|u - w\|^2 = \|u\|^2 - \|w\|^2 - 2\langle u, w \rangle$, $\forall u, w \in \mathcal{H}$;
- (2) $\|u + w\|^2 \leq \|u\|^2 + 2\langle w, u + w \rangle$, $\forall u, w \in \mathcal{H}$;
- (3) $\|\alpha u + (1 - \alpha)w\|^2 = \alpha\|u\|^2 + (1 - \alpha)\|w\|^2 - \alpha(1 - \alpha)\|u - w\|^2$, $\forall \alpha \in [0, 1]$, $u, w \in \mathcal{H}$.

Lemma 3.1. (Opial Property [24]). We denote by $\mathcal{H}$ the Hilbert space. $\mathcal{H}$ is said to have the Opial property if for every weakly convergent sequence $\{u_k\}$ in $\mathcal{H}$ with a weak limit z , $\liminf_{k \rightarrow \infty} \|u_k - z\| < \liminf_{k \rightarrow \infty} \|u_k - w\|$, for all $w \in \mathcal{H}$ with $w \neq z$.

Lemma 3.2. (Goebel Sequence [15]). Let $\mathcal{C}$ be a nonempty, convex, and closed subset of $\mathcal{H}$, and let $\mathcal{S} : \mathcal{C} \rightarrow \mathcal{C}$ be a nonexpansive mapping. Assume that $\{u_k\}$ is a sequence in $\mathcal{C}$. If $u_k \rightharpoonup u^* \in \mathcal{C}$ and $\{\mathcal{S}u_k - u_k\} \rightarrow v \in \mathcal{C}$, then $\mathcal{S}u^* - u^* = v$.

Lemma 3.3. (Osilike Sequences [25]). Let $\{s_k\}$, $\{t_k\}$, and $\{\delta_k\}$ be nonnegative sequences such that $s_{k+1} \leq (1 + \delta_k)s_k + t_k$, $k \geq 1$. If $\sum_{k=1}^{\infty} \delta_k < +\infty$ and $\sum_{k=1}^{\infty} t_k < +\infty$, then $\lim_{k \rightarrow \infty} s_k$ exists.

Lemma 3.4. (Opial Sequences [24]). Let Ω be a subset of $\mathcal{H}$ and $\{u_k\}$ be a sequence in $\mathcal{H}$ that satisfy:

- (i) for every $p \in \Omega$, $\lim_{k \rightarrow \infty} \|u_k - p\|$ exists;
(ii) each weak-cluster point of the sequence $\{u_k\}$ is in Ω .
Then, $\{u_k\}$ converges weakly to an element in Ω .

4. PARALLEL DOUBLE INERTIAL MANN-TYPE ALGORITHM

In this section, we provide the convergence theorem of parallel double inertial Mann algorithm for nonexpansive mappings in a real Hilbert space. Let $\{\mathcal{S}_i\}_{i=1}^N$ be a family of nonexpansive mappings on $\mathcal{H}$, and $\bigcap_{i=1}^N \mathcal{F}(\mathcal{S}_i)$ be nonempty.

Algorithm 4.1. A modified parallel Mann algorithm ending with a double inertial method

Initialization : Let $w_{-1}, w_0, u_0 \in \mathcal{H}$ be arbitrary, $\alpha_k, \lambda_k \in (-\infty, \infty)$, $\beta_{i,k} \in (0, 1)$ for all $i = 1, \dots, N$ with $\sum_{i=1}^N \beta_{i,k} = 1$.

Iteration Step: for $k = 0$, compute

$$w_{k+1} = \beta_{0,k} u_k + \sum_{i=1}^N \beta_{i,k} \mathcal{S}_i u_k,$$

$$u_{k+1} = w_{k+1} + \alpha_k (w_{k+1} - w_k) + \lambda_k (w_k - w_{k-1}).$$

Then, set $k = k + 1$ and update in **Iteration Step**.

Theorem 4.2. Let $\mathcal{H}$ be a real Hilbert space and $\{\mathcal{S}_i\}_{i=1}^N : \mathcal{H} \rightarrow \mathcal{H}$ be a family of nonexpansive mappings such that $\bigcap_{i=1}^N \mathcal{F}(\mathcal{S}_i)$ is nonempty. Let $\{u_k\}$ be defined by Algorithm 4.1. Assume that the following conditions are satisfied:

- (i) $\sum_{k=1}^{\infty} |\alpha_k| \|w_{k+1} - w_k\| < \infty$;
(ii) $\sum_{k=1}^{\infty} |\lambda_k| \|w_k - w_{k-1}\| < \infty$;
(iii) $\liminf_{k \rightarrow \infty} \beta_{0,k}, \beta_{i,k} > 0$ for all $i = 1, 2, \dots, N$.

Then, $\{u_k\}$ converges weakly to a point in $\bigcap_{i=1}^N \mathcal{F}(\mathcal{S}_i)$.

Proof. Let $p \in \bigcap_{i=1}^N \mathcal{F}(\mathcal{S}_i)$. We find that $\|w_{k+1} - p\| \leq \beta_{0,k} \|u_k - p\| + \sum_{i=1}^N \beta_{i,k} \|\mathcal{S}_i u_k - p\| \leq \|u_k - p\|$ and $\|u_{k+1} - p\| \leq \|w_{k+1} - p\| + |\alpha_k| \|w_{k+1} - w_k\| + |\lambda_k| \|w_k - w_{k-1}\|$. Thus

$$\|u_{k+1} - p\| \leq \|u_k - p\| + |\alpha_k| \|w_{k+1} - w_k\| + |\lambda_k| \|w_k - w_{k-1}\|.$$

From Lemma 3.3, we have that $\lim_{k \rightarrow \infty} \|u_k - p\|$ exists. Thus $\{u_k\}$ is bounded, so is $\{w_k\}$.

Next, we show that $\{u_k\}$ converges weakly to a point in $\bigcap_{i=1}^N \mathcal{F}(\mathcal{S}_i)$. Consider

$$\begin{aligned} \|w_{k+1} - p\|^2 &= \beta_{0,k} \|u_k - p\|^2 + \sum_{i=1}^N \beta_{i,k} \|\mathcal{S}_i u_k - p\|^2 - \beta_{0,k} \sum_{i=1}^N \beta_{i,k} \|\mathcal{S}_i u_k - u_k\|^2 \\ &\leq \|u_k - p\|^2 - \beta_{0,k} \sum_{i=1}^N \beta_{i,k} \|\mathcal{S}_i u_k - u_k\|^2. \end{aligned} \quad (4.1)$$

Note that

$$\begin{aligned} \|u_{k+1} - p\|^2 &= \|w_k + \alpha_k (w_{k+1} - w_k) + \lambda_k (w_k - w_{k-1}) - p\|^2 \\ &\leq \|w_{k+1} - p\|^2 + 2 \langle \alpha_k (w_{k+1} - w_k) + \lambda_k (w_k - w_{k-1}), u_{k+1} - p \rangle. \end{aligned} \quad (4.2)$$

Combining (4.1) and (4.2), we obtain

$$\begin{aligned} \|u_{k+1} - p\|^2 &\leq \|u_k - p\|^2 + 2\langle \alpha_k(w_{k+1} - w_k) + \lambda_k(w_k - w_{k-1}), u_{k+1} - p \rangle \\ &\quad - \beta_{0,k} \sum_{i=1}^N \beta_{i,k} \|\mathcal{S}_i u_k - u_k\|^2. \end{aligned}$$

Hence, we have

$$\begin{aligned} \beta_{0,k} \sum_{i=1}^N \beta_{i,k} \|\mathcal{S}_i u_k - u_k\|^2 &\leq \|u_k - p\|^2 - \|u_{k+1} - p\|^2 \\ &\quad + 2\langle \alpha_k(w_{k+1} - w_k) + \lambda_k(w_k - w_{k-1}), u_{k+1} - p \rangle. \end{aligned}$$

By assumptions (i)-(iii) and the fact that $\lim_{k \rightarrow \infty} \|u_k - p\|$ exists, we obtain

$$\lim_{k \rightarrow \infty} \|\mathcal{S}_i u_k - u_k\| = 0, \quad \text{for all } i = 1, 2, \dots, N. \quad (4.3)$$

Since $\{u_k\}$ is bounded, we suppose that z is a weak sequential cluster point of $\{u_k\}$. It follows by (4.3) with Lemma 3.2 that $z \in \bigcap_{i=1}^N \mathcal{F}(\mathcal{S}_i)$. By using Lemma 3.4, we conclude that $\{u_k\}$ converges weakly to an element in $\bigcap_{i=1}^N \mathcal{F}(\mathcal{S}_i)$. $\square$

Remark 4.3. From Algorithm 4.1, we see that

- (i) To use projection $\mathcal{P}_{\mathcal{C}}$, we can set $\mathcal{S}_1 = \mathcal{P}_{\mathcal{C}}$ to obtain $\mathcal{C} = \mathcal{F}(\mathcal{S}_1)$;
- (ii) for the inclusion problem: find $u \in \mathcal{H}$ such that $0 \in (\mathcal{A} + \mathcal{B})u$, where $\mathcal{A} : \mathcal{H} \rightarrow 2^{\mathcal{H}}$ and $\mathcal{B} : \mathcal{H} \rightarrow \mathcal{H}$ with the solution set $\text{zer}(\mathcal{A} + \mathcal{B})$, the resolvent operator of $\mathcal{A}$ is defined by $J_{\delta}^{\mathcal{A}} = (I + \delta \mathcal{A})^{-1}$ where $\delta > 0$. It is well-known that if $\mathcal{A}$ is maximally monotone and $\mathcal{B}$ is $\mathcal{L}$ -Lipschitz monotone, then we can set $\mathcal{S}_i = J_{\delta_i}^{\mathcal{A}}(I - \delta_i \mathcal{B}_i)$, where $\delta_i \in (0, \frac{2}{\mathcal{L}})$.

From the above remark, we can modify our algorithm to find a solution in $\bigcap_{i=2}^N \text{zer}(\mathcal{A}_i + \mathcal{B}_i) \cap \mathcal{C}$.

Algorithm 4.4. *Parallel Mann algorithm with two inertial*

Initialization : Let $w_{-1}, w_0, u_0 \in \mathcal{H}$ be arbitrary, $\alpha_k, \lambda_k \in (-\infty, \infty)$, $\beta_{i,k} \in (0, 1)$ for all $i = 0, 1, \dots, N$ with $\sum_{i=0}^N \beta_{i,k} = 1$.

Iteration Step for $k = 0$, compute

$$\begin{aligned} w_{k+1} &= \beta_{0,k} u_k + \beta_{1,k} \mathcal{P}_{\mathcal{C}} u_k + \sum_{i=2}^N \beta_{i,k} J_{\delta_i}^{\mathcal{A}_i} (u_k - \delta_i \mathcal{B}_i u_k), \\ u_{k+1} &= w_{k+1} + \alpha_k (w_{k+1} - w_k) + \lambda_k (w_k - w_{k-1}). \end{aligned}$$

Then, set $k = k + 1$ and update in **Iteration Step**.

5. NUMERICAL EXPERIMENTS

To support our main theorem, we now give an example in infinitely dimensional spaces $L_2[0, 1]$, where the L_2 -norm is defined by $\|u(t)\| = \sqrt{\int_0^1 |u(t)|^2 dt}$ with the inner product given by $\langle u, w \rangle = \int_0^1 u(t)w(t) dt$ where $u(t) \in L_2[0, 1]$. The experiments were performed by using MATLAB R2022A on an HP Laptop equipped with an Intel(R) Core(TM) i7-1165G7 CPU and 16.00 GB of RAM.

Example 5.1. Let $\mathcal{H} = L_2[0, 1]$. Define mappings $\mathcal{T}_1, \mathcal{T}_2 : \mathcal{H} \rightarrow \mathcal{H}$ by $\mathcal{T}_1 u(t) = \frac{2}{3}u(t)$ and $\mathcal{T}_2 u(t) = \frac{1}{2}u(t)$, where $u(t) \in L_2[0, 1]$.

In the first part of this example, we select the optimal parameters for our algorithm by considering the number of iterations and CPU times. Table 1 shows the results when $\lambda = 0.1$ with various values of a_k and α . In Table 2, we set $a_k = \frac{1}{(k+10)^5}$ with different λ values. Additionally, we plot the errors in Figures 1 and 2.

The initial points $u_0 = 2\log(t) + 1$, $w_{-1} = 2t^2 + t - 1$, and $w_0 = 2\sin(t)$. The sequences $\beta_{0,k}, \beta_{1,k}, \beta_{2,k}, \beta_{3,k}$ are 0.25 and the parameters of α_k and λ_k are selected by

$$\alpha_k = \begin{cases} \min \left\{ \frac{a_k}{\|w_{k+1} - w_k\|}, \alpha \right\}, & \text{if } w_{k+1} \neq w_k, k > N; \\ \alpha, & \text{otherwise,} \end{cases} \quad (5.1)$$

and

$$\lambda_k = \begin{cases} \min \left\{ \frac{\bar{\lambda}_k}{\|w_k - w_{k-1}\|}, \lambda \right\}, & \text{if } w_k \neq w_{k-1}, k > N; \\ \lambda, & \text{otherwise.} \end{cases} \quad (5.2)$$

Moreover, we use Cauchy error as the stop criterion defined by $\|u_{k+1} - u_k\|_{L_2}^2 < 1.5 \times 10^{-4}$.

TABLE 1. Errors plots with different a_k and α for $\lambda = 0.1$

a_k	0	$\frac{1}{(k+1)^2}$	$\frac{10}{(k+1)^2}$	$\frac{1}{(k+10)^2}$	$\frac{1}{(k+10)^5}$
No. Iter	11	13	15	11	11
CPU time(s)	1.9189	4.3841	5.5889	3.8600	1.6713
α	0.1	0.2	0.5	0.7	0.9
No. Iter	14	13	20	24	32
CPU time(s)	2.0337	1.8994	2.8697	3.6784	4.6759

Table 1 shows that the number of iterations and CPU times are minimized when $\lambda = 0.1$ and $a_k = \frac{1}{(k+10)^5}$. This indicates that these parameters are suitable choices for our algorithm.

TABLE 2. Errors plots with different λ for $a_k = \frac{1}{(k+10)^5}$

λ	0	0.05	0.1	0.15	0.2	0.3
No. Iter	16	14	11	14	15	18
CPU time(s)	2.3234	2.0627	1.6336	2.0553	2.2353	2.5481
λ	0.4	0.5	0.6	0.7	0.8	0.9
No. Iter	19	23	25	28	34	45
CPU time(s)	2.7443	3.2889	3.6294	4.1100	5.1567	6.6451

In Table 2, for $a_k = \frac{1}{(k+10)^5}$ and different values of λ , it is found that if $\lambda = 0.1$, then the number of iterations and CPU times are the lowest. Therefore, we will choose $a_k = \frac{1}{(k+10)^5}$ and $\lambda = 0.1$ for our algorithm to compare with other algorithms.

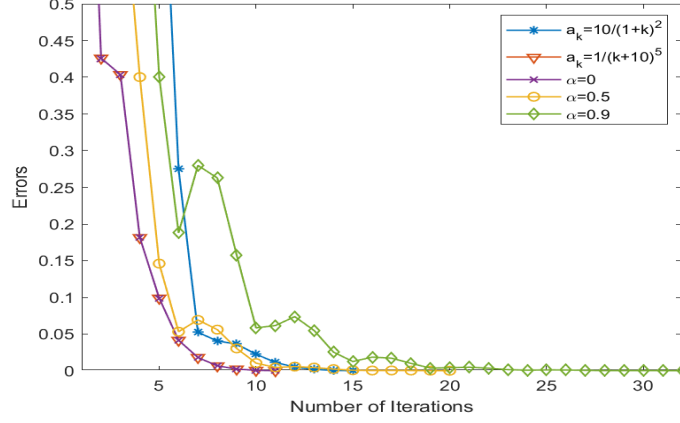


FIGURE 1. Numerical error plots (using L_2 norm in-between consecutive iterations) for different values of a_k and α .

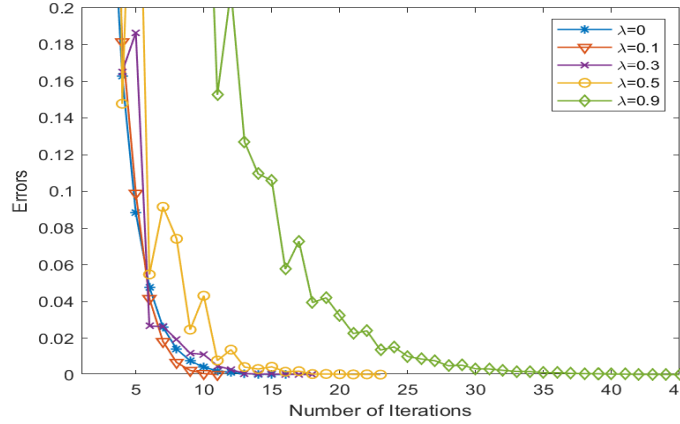


FIGURE 2. Numerical error plots (using L_2 norm in-between consecutive iterations) for different values of λ .

Next, we present numerical experiments of the infinite space between Algorithm 4.1 and Algorithm 2.1, where the parameters of α_k and λ_k are defined by (5.1) and (5.2), respectively. We choose $a_k = \frac{1}{(k+10)^5}$, $\lambda = 0.1$ for our algorithm and $a_k = \frac{1}{(k+100)^2}$, $\beta_{i,k} = 0.99$, $\gamma_{i,k} = \frac{k}{k+1}$ for Algorithm (2.1).

TABLE 3. Comparison between our algorithm and Algorithm 2.1

Method	Number of iterations	CPU Time(s)
Algorithm 1.1	16	1.7968
Algorithm 3.1	11	1.6144

Table 3 shows a comparison of iterations and CPU time between our algorithm and Algorithm 2.1. Our algorithm takes 1.6144 seconds for computation with tolerance of 1.5×10^{-4} , which is less than that of Algorithm 2.1. Next, we present a plot of the Cauchy error for our algorithm and Algorithm 2.1 in Figure 3.

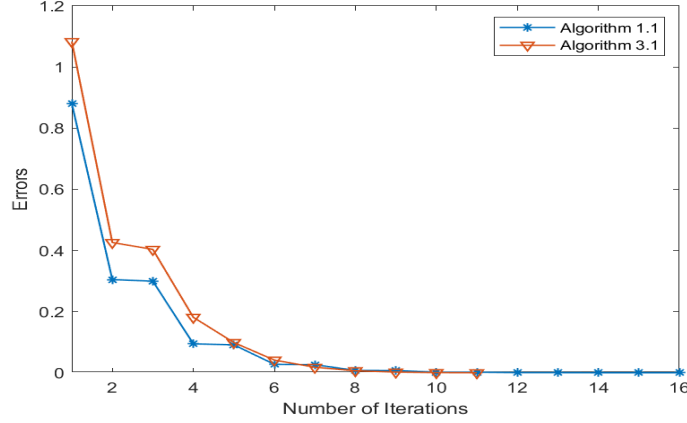


FIGURE 3. Error plots of Algorithm 4.1 and Algorithm 2.1

Example 5.2. Let $\mathcal{H} = L_2[0, 1]$. Define

- $\mathcal{C} = \{u(t) \in L_2[0, 1] : \|u(t) - e^t\| \leq 2\}$, with

$$\mathcal{P}_{\mathcal{C}} = \begin{cases} e^t + 2 \frac{u(t) - e^t}{\|u(t) - e^t\|}, & \text{if } u(t) \in \mathcal{C}; \\ u(t), & \text{if } u(t) \notin \mathcal{C}. \end{cases}$$

- $\mathcal{A}_i : L_2[0, 1] \rightarrow L_2[0, 1]$ by $\mathcal{A}_1 u(t) = 2u(t)$ and $\mathcal{A}_2(t) = 5u(t)$, $\forall u(t) \in L_2[0, 1]$,
- $\mathcal{B}_i : L_2[0, 1] \rightarrow L_2[0, 1]$ by $\mathcal{B}_1 u(t) = \frac{2}{3}u(t)$ and $\mathcal{B}_2 u(t) = \frac{1}{2}u(t)$, $\forall u(t) \in L_2[0, 1]$.

We obtain $J_{\delta_1}^{\mathcal{A}_1}(u_k - \delta_1 \mathcal{B}_1 u_k) = \frac{u_k - \delta_1 \mathcal{B}_1 u_k}{1 + 2\delta_1}$, and $J_{\delta_2}^{\mathcal{A}_2}(u_k - \delta_2 \mathcal{B}_2 u_k) = \frac{u_k - \delta_2 \mathcal{B}_2 u_k}{1 + 5\delta_2}$, where $\delta_1, \delta_2 > 0$, $u(t) \in L_2[0, 1]$ and $t \in [0, 1]$. By employing the Algorithm 4.4, we can find a solution in $\bigcap_{i=2}^3 (\mathcal{A}_i + \mathcal{B}_i)^{-1}(0) \cap \mathcal{C}$. The initial points $u_0 = 2 \sin(t) - 1$, $w_{-1} = \log(t) + 1$, $w_0 = \cos(t)$, the sequences $\beta_{0,k}, \beta_{1,k}, \beta_{2,k}, \beta_{3,k} = 0.25$, and δ_i is chosen in $(0, \frac{2}{\|\mathcal{A}_i\|^2})$. The parameters α_k and λ_k are selected by:

$$\alpha_k = \begin{cases} \frac{1}{k^2 \|w_{k+1} - w_k\|}, & \text{if } w_{k+1} \neq w_k, k > N; \\ \alpha, & \text{otherwise,} \end{cases} \quad (5.3)$$

and

$$\lambda_k = \begin{cases} \frac{1}{k^2 \|w_k - w_{k-1}\|}, & \text{if } w_k \neq w_{k-1}, k > N; \\ \lambda, & \text{otherwise.} \end{cases} \quad (5.4)$$

Moreover, we use Cauchy error as the stop criterion defined by $\|u_{k+1} - u_k\|_{L_2}^2 < 10^{-4}$. Next, we present numerical experiments of infinite dimension space for $\lambda = 0.05$ and various values of α in Table 4 and Figure 4. After that, we set $\alpha = 0.1$ with different λ . The results are shown in Table 5 and Figure 5.

TABLE 4. Numerical errors (using L_2 norm in-between consecutive iterations) in function of different values of α with fixed $\lambda = 0.05$.

α	0	0.1	0.2	0.3	0.4
No. Iter	10	8	11	11	13
CPU time(s)	25.5281	16.0457	19.1868	20.6780	22.0805
α	0.5	0.6	0.7	0.8	0.9
No. Iter	14	17	17	17	21
CPU time(s)	40.3871	38.6881	34.5391	34.8597	40.2402

TABLE 5. Numerical errors (using L_2 norm in-between consecutive iterations) in function of different values of λ with fixed $\alpha = 0.1$.

λ	0	0.05	0.1	0.15	0.2	0.3
No. Iter	11	8	11	12	12	14
CPU time(s)	13.2278	17.6773	16.1834	22.5096	23.8145	35.4561
λ	0.4	0.5	0.6	0.7	0.8	0.9
No. Iter	17	17	22	25	27	30
CPU time(s)	31.6287	40.9257	35.9614	55.7404	65.2204	58.8247

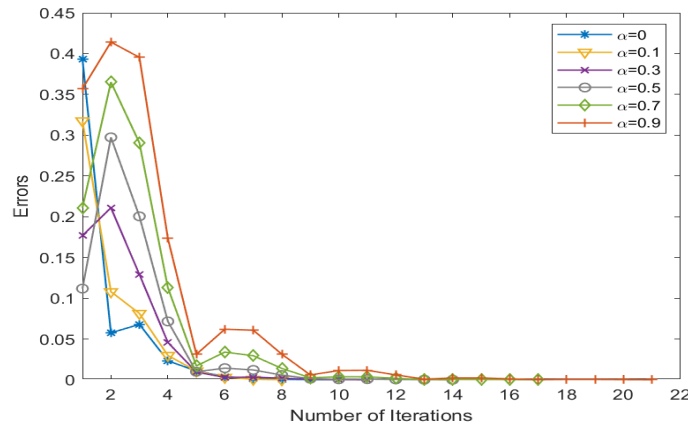


FIGURE 4. Numerical error plots (using L_2 norm in-between consecutive iterations) for different values of α .

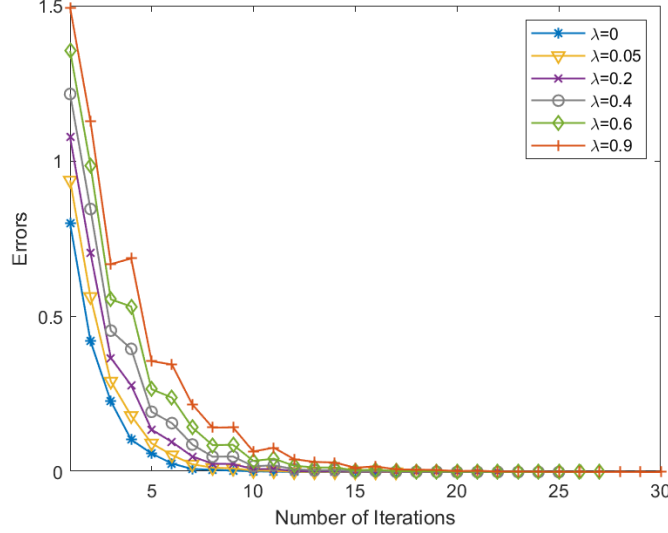


FIGURE 5. Numerical error plots (using L_2 norm in-between consecutive iterations) for different values of λ .

As seen in Table 4, the number of iterations and the CPU time tend to increase for all sequences when λ increases. The observed pattern indicates that larger values of α leads to slower convergence of our algorithm. In Table 5, it can be observed that the number of iterations and the CPU time are increasing for all sequences when λ increases. The observed result indicates that larger values of λ decrease the speed of convergence of the proposed method.

6. APPLICATION TO BLIND IMAGE DEBLURRING

The image recovery problem can be conceptualized as a linear equation represented by:

$$b = \mathcal{D}u + \rho, \quad (6.1)$$

where the original image is denoted as $u \in \mathbb{R}^{n \times 1}$, the degraded image is labeled as $b \in \mathbb{R}^{m \times 1}$, the noise term is given by $\rho \in \mathbb{R}^{m \times 1}$, and the blurring matrix is represented as $\mathcal{D} \in \mathbb{R}^{m \times n}$. Problem (6.1) is similar to the following convex minimization model: $\min_{u \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{D}u - b\|_2^2$. We focus on the following minimization problem:

$$\min_{u \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{D}_1 u - b_1\|_2^2, \min_{u \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{D}_2 u - b_2\|_2^2, \dots, \min_{u \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{D}_N u - b_N\|_2^2,$$

where u is the original image, b_i is the blurred image by the blurred matrix $\mathcal{D}_i$ for all $i = 1, 2, \dots, N$. For $b_i = b$, we have

$$\min_{u \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{D}_1 u - b\|_2^2, \min_{u \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{D}_2 u - b\|_2^2, \dots, \min_{u \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{D}_N u - b\|_2^2, \quad (6.2)$$

where the blurring operator is selected randomly by $\mathcal{D}$ without knowing the actual blurring operator beforehand. This method aims to find a more practical way to restore images in real life. Accordingly, (6.2) can be transformed to the inclusion problems (4.3) as $\mathcal{S}_i u = J_{\delta_i^{\mathcal{A}_i}}(u - \delta_i \mathcal{B}_i u)$, where $\mathcal{B}_i(u) = \mathcal{D}_i^T(b - \mathcal{D}_i u)$, $\mathcal{A}_i(u) = 0$, and $\delta_i \in (0, \frac{2}{\|\mathcal{D}_i\|_2^2})$. The parameters of Algorithm 4.1

are defined by $\beta_{0,k} = 0.1$, $\beta_{1,k} = \beta_{2,k} = \dots = \beta_{10,k} = 0.09$, $\delta_i = \frac{1.99}{\|\mathcal{D}_i\|^2}$ for $i = 1, 2, 3, \dots, 10$. The conditions of α_k , λ_k are given by (5.3) and (5.4), respectively. In this case, we choose $\alpha = 0.89$, $\lambda = 0.09$ and the initial points w_{-1}, w_0, u_0 are blurred images.

We use a parallel approach to solve the motion blur problem. Ten blur matrices are used in each concurrent cycle. We set $\mathcal{D}_1 = (L, \theta)$, $\mathcal{D}_2 = (L+1, \theta)$, $\mathcal{D}_3 = (L+2, \theta)$, $\mathcal{D}_4 = (L, \theta)$, $\mathcal{D}_5 = (L, \theta - 1)$, $\mathcal{D}_6 = (L+1, \theta - 1)$, $\mathcal{D}_7 = (L+2, \theta - 1)$, $\mathcal{D}_8 = (L, \theta + 1)$, $\mathcal{D}_9 = (L+1, \theta + 1)$, and $\mathcal{D}_{10} = (L+2, \theta + 1)$, where θ is the angle and L is length of motion blur. We consider 5 cases with $\theta = 135, 140, 145, 150$ that are shown in Table 6.

TABLE 6. Groups of motion blurring matrix with different length (L) and angles (θ) to be run in parallel.

		Case I	Case II	Case III	Case IV	Case V
Group I	(L, θ)	(67,135)	(69,135)	(71,135)	(73,135)	(75,135)
Group II	(L, θ)	(67,140)	(69,140)	(71,140)	(73,140)	(75,140)
Group III	(L, θ)	(67,145)	(69,145)	(71,145)	(73,145)	(75,145)
Group IV	(L, θ)	(67,150)	(69,150)	(71,150)	(73,150)	(75,150)

First, the input image to the algorithm is the blurred image of a car license plate. Then, we apply our parallel algorithm using groups of motion blurring matrix with different length (L) and angles (θ) to recover the image. Next, we present the blurred images with 5000 iterations for our algorithm in Figures 6, 8, 10 and 12. Moreover, Figures 7, 9, 11 and 13 show the plot of the Cauchy error $\|u_{n+1} - u_n\| < 10^{-5}$. We conduct numerical experiments of our proposed algorithm in image restoration. The experiments were conducted by MATLAB R2022A on a computer with an Intel(R) Core(TM) i9-9820X processor, graphics rtx2080 and 64.00 GB RAM.

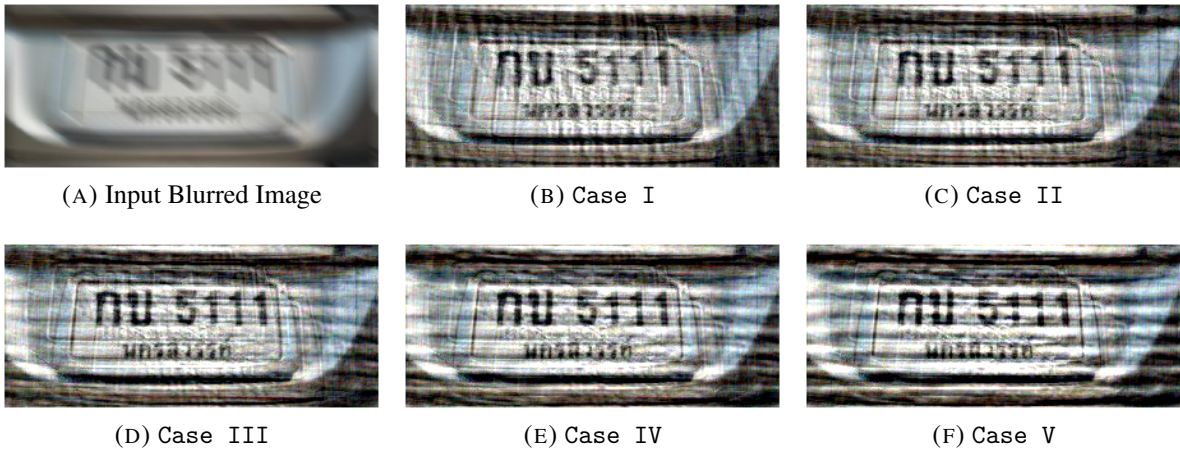


FIGURE 6. Restored images ($b-f$) of car license plates for Group I with respect to the input blurred image (a).

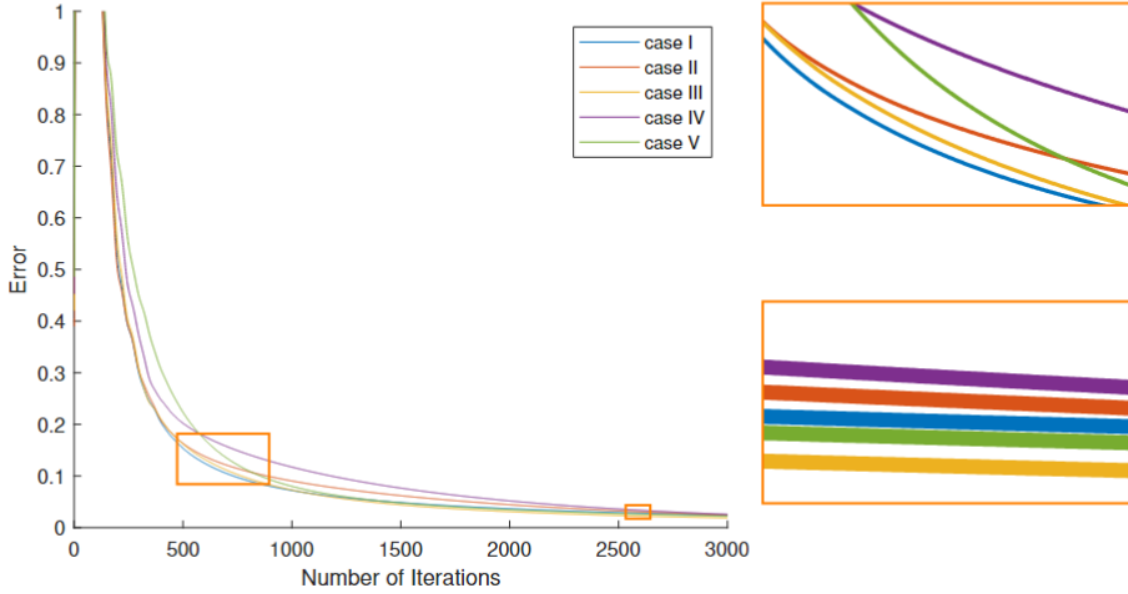


FIGURE 7. Numerical error plots (using L_2 norm in-between consecutive iterations) for Group I in function of the number of iterations. The top-right frame shows a close-up view of the error around the 600th iteration. The bottom-right frame shows a close-up view of the error around the 2600th iteration.

In Figure 6, the numbers on the car license plate are visible and readable. However, both the large and small letters on the license plate are not visible in every instance. This means that Group I is not yet suitable for recovering the image. In Figure 7, the graph of the Cauchy error is shown. From the graph, it can be observed that in every case, the error gradually decreases with each iteration, indicating that the calculated values are converging towards the true or desired value.

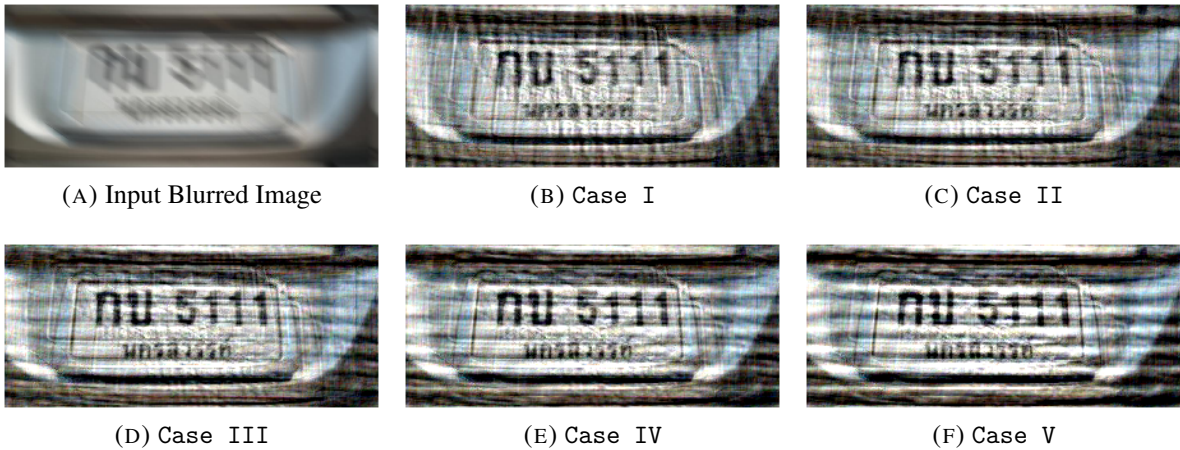


FIGURE 8. Restored images (*b-f*) of car license plates for Group II with respect to the input blurred image (*a*).

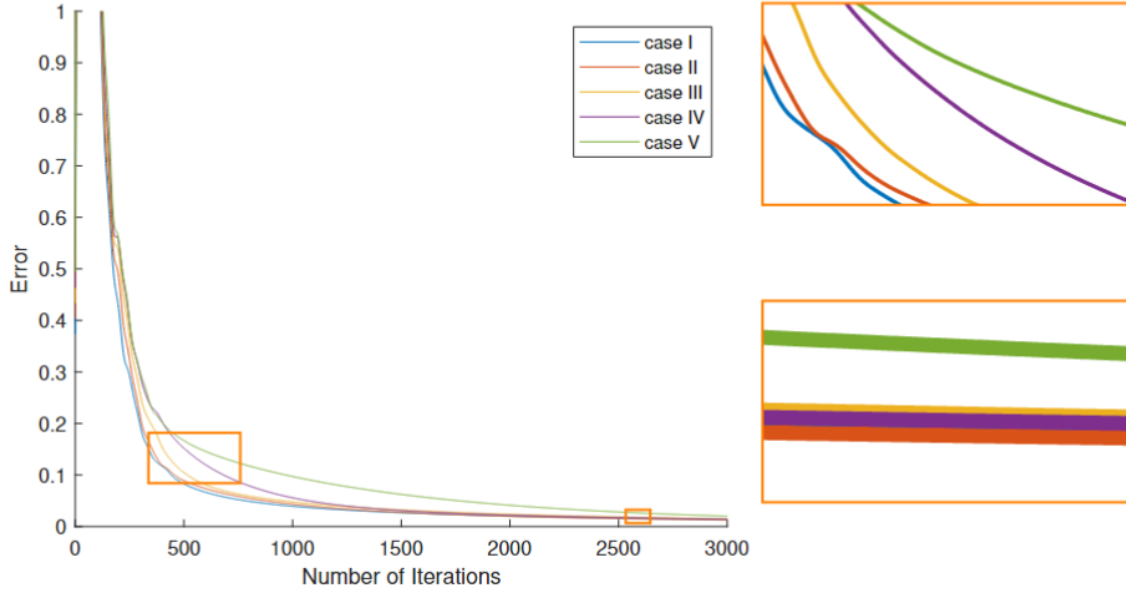


FIGURE 9. Numerical error plots (using L_2 norm in-between consecutive iterations) for Group II in function of the number of iterations. The top-right frame shows a close-up view of the error around the 500th iteration. The bottom-right frame shows a close-up view of the error around the 2600th iteration.

In Figure 8, the image recovered by Group II yields results similar to those of Group I, where the numbers are readable in every case, but the large and small letters are still unreadable. Moreover, in Figure 9, it is observed that the Cauchy error graph gradually decreases in each case.

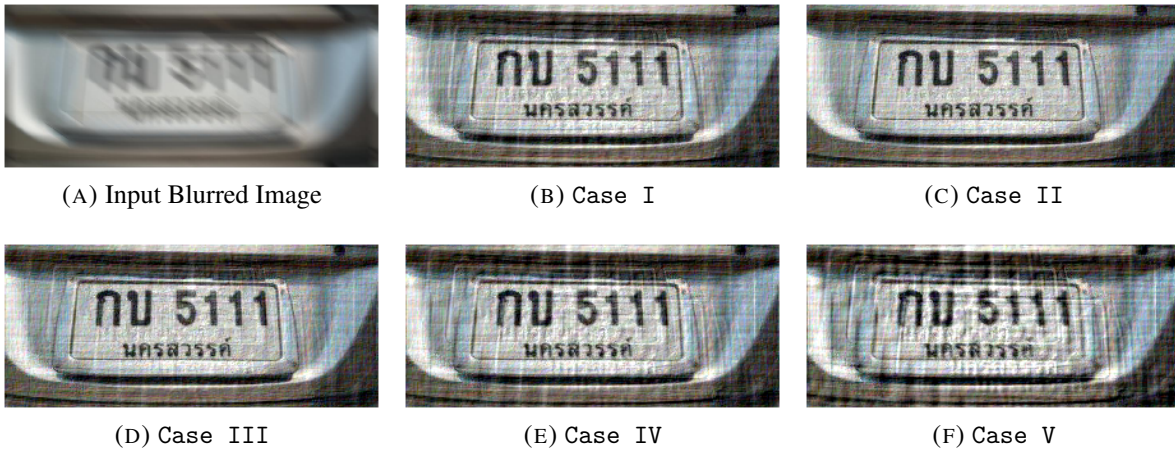


FIGURE 10. Restored images (*b-f*) of car license plates for Group III with respect to the input blurred image (*a*).

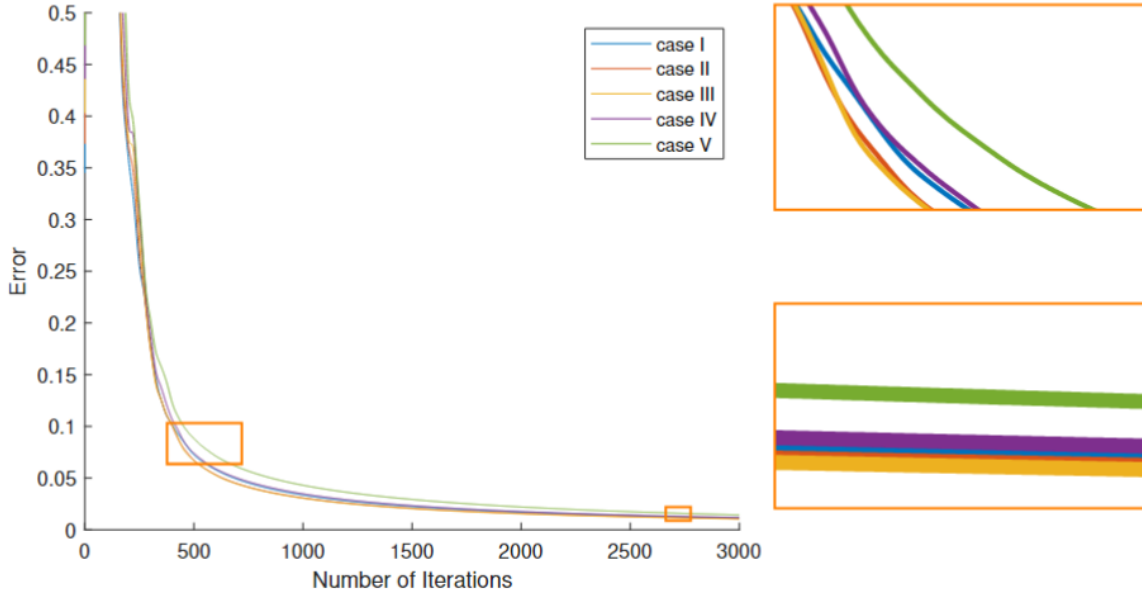


FIGURE 11. Numerical error plots (using L_2 norm in-between consecutive iterations) for Group III in function of the number of iterations. The top-right frame shows a close-up view of the error around the 600th iteration. The bottom-right frame shows a close-up view of the error around the 2700th iteration.

In Figure 10, we observe that images recovered in Group III allow the numbers and letters on the license plate to be clearly readable in each case. This indicates that Group III is favorable for the recovery of motion blur. Furthermore, the results in Figure 11 also support convergence in each case.

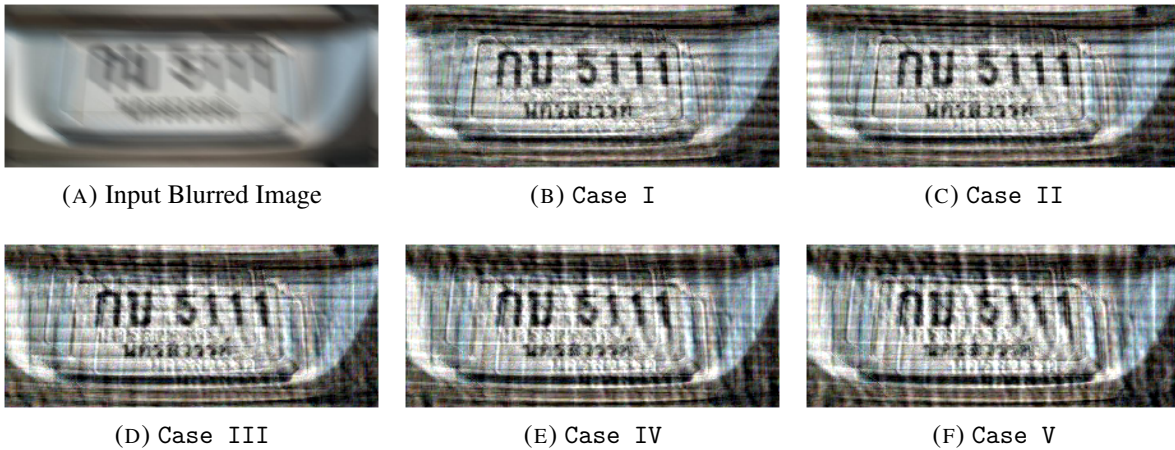


FIGURE 12. Restored images (*b-f*) of car license plates for Group IV with respect to the input blurred image (*a*).

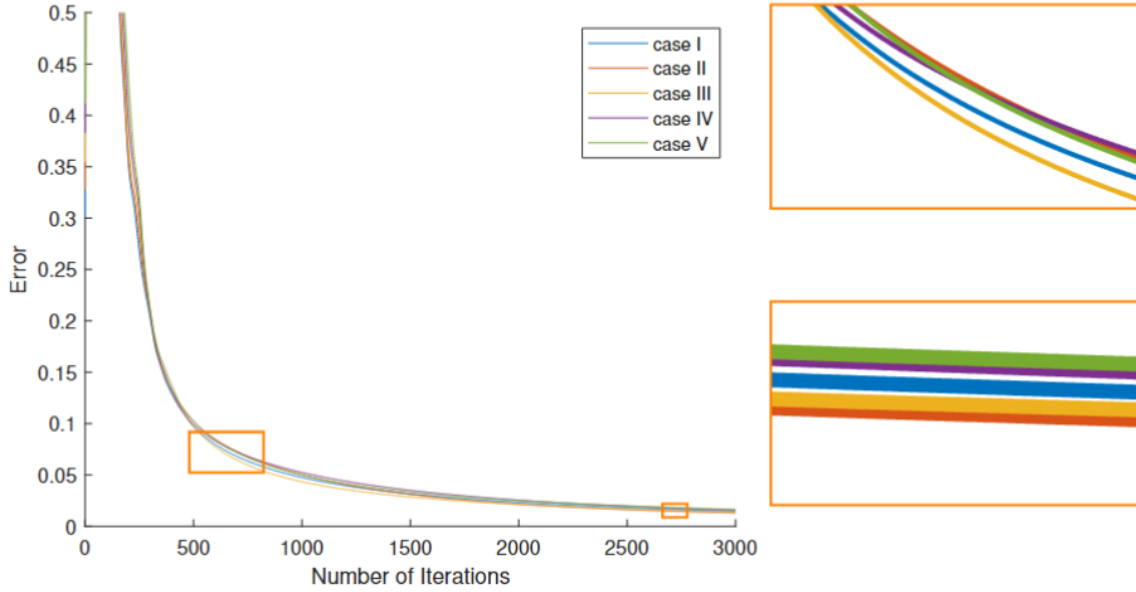


FIGURE 13. Numerical error plots (using L_2 norm in-between consecutive iterations) for Group IV in function of the number of iterations. The top-right frame shows a close-up view of the error around the 600th iteration. The bottom-right frame shows a close-up view of the error around the 2700th iteration.

Figure 12 shows the Group IV used for image recovery. The results indicate that for the Cases I–II, large numbers and letters are readable but small letters remain hardly identifiable. Additionally, some numbers are unclear in Cases III–IV–V and neither the large or small letters can be identified clearly. Consequently, the Group IV is suitable for image identification. Finally, we observe that Group III is the most favorable group for recovering the license plate image among all groups. In all cases, all parts of the license plate’s details are apparent. It is observed that Cases I–II–III show more subtle details than Cases IV–V.

7. CONCLUSION

In this paper, we developed a novel parallel algorithm expanding the Mann algorithm with double inertial extrapolations to identify a common fixed point for a finite family of nonexpansive mappings. As part of the convergence analysis, we established the weak convergence of our algorithm under some conditions. Furthermore, we presented an approach to approximate the solution of the inclusion problem in an infinite dimensional space. Our algorithm was utilized to enhance the readability of characters present in a collection of blurred car license plate images by selecting well-known blur matrices for different potential motion blurs. The key benefit of our algorithm is that the propose approach does not require the explicit estimation of the direction and length of the unknown motion blur in the input image. Finally, our experiments demonstrated the effectiveness of our algorithm to address the problem of blind image deblurring allowing intricate details to be clearly discernible in various image components.

Acknowledgments

The authors sincerely thank the anonymous reviewers for their suggestions that improve the manuscript substantially. This research was supported by University of Phayao and Thailand Science Research and Innovation Fund (Fundamental Fund 2026, Grant No. 2251/2568).

REFERENCES

- [1] F. Alvarez, H. Attouch, An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping, *Set-Valued Anal.* 9 (2001) 3-11.
- [2] N.T. An, P.D. Dong, X. Qin, Robust feature selection via nonconvex sparsity-based methods, *J. Nonlinear Var. Anal.* 5 (2021) 59-77.
- [3] F.E. Browder, Existence of periodic solutions for nonlinear equations of evolution, *Proc. Natl. Acad. Sci. USA* 53 (1965), 1100-1103.
- [4] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.* 2 (2009) 183-202.
- [5] W. Chalamjiak, S.A. Khan, D. Yambangwai, K.R. Kazmi, Strong convergence analysis of common variational inclusion problems involving an inertial parallel monotone hybrid method for a novel application to image restoration, *Rev. Real Acad. Cienc. Exactas Fís. Nat. Ser. A Math.* 114 (2020) 1-20.
- [6] P. Chalamjiak, S. Suantai, P. Sunthayuth, An explicit parallel algorithm for solving variational inclusion problem and fixed point problem in Banach spaces, *Banach J. Math. Anal.* 14 (2020) 20-40.
- [7] S.H. Chan, X. Wang, O.A. Elgendy, Plug-and-play ADMM for image restoration: Fixed-point convergence and applications, *IEEE Trans. Comput. Imaging* 3 (2016) 84-98.
- [8] L.C. Ceng, On inertial subgradient extragradient rule for monotone bilevel equilibrium problems, *Fixed Point Theory* 24 (2023) 101-126.
- [9] C. Chairatsiripong, D. Yambangwai, T. Thianwan, Convergence analysis of M-iteration for $\mathcal{G}$ -nonexpansive mappings with directed graphs applicable in image deblurring and signal recovering problems, *Demonstr. Math.* 56 (2023) 20220234.
- [10] P.L. Combettes, J.C. Pesquet, Fixed point strategies in data science, *IEEE Trans. Signal Process.* 69 (2021), 3878-3905.
- [11] Q.L. Dong, Y.Y. Lu, J. Yang, The extragradient algorithm with inertial effects for solving the variational inequality, *Optimization*. 65 (2016) 2217-2226.
- [12] Y. Dong, Q. Luo, On an inertial Krasnosel'skii-Mann iteration, *Appl. Set-Valued Anal. Optim.* 6 (2024) 103-112.
- [13] B.V. Dinh, N.T.T. Ha, H.D. Manh, N.H. Nam, Inertial extragradient and Ishikawa iterative methods for approximating common solutions to pseudomonotone equilibrium problems and fixed points of quasicontractive operators, *Commun. Optim. Theory* 2025 (2025) 32.
- [14] J. Fan, L. Liu, X. Qin, A subgradient extragradient algorithm with inertial effects for solving strongly pseudomonotone variational inequalities, *Optimization* 69 (2020) 2199-2215.
- [15] K. Goebel, *Topics in Metric Fixed Point Theory*, Cambridge Stud. Adv. Math. 28, Cambridge University Press, Cambridge, 1990.
- [16] O.S. Iyiola, Y. Shehu, Convergence results of two-step inertial proximal point algorithm, *Appl. Numer. Math.* 182 (2022), 57-75.
- [17] H. Iiduka, Fixed point optimization algorithm and its application to power control in CDMA data networks, *Math. Program.* 133 (2012) 227-242.
- [18] N. Jun-On, R. Suparatulatorn, M. Gamal, W. Chalamjiak, An inertial parallel algorithm for a finite family of $\mathcal{G}$ -nonexpansive mappings applied to signal recovery, *AIMS Math.* 7 (2022) 1775-1790.
- [19] S.A. Khan, K.R. Kazmi, D. Yambangwai, W. Chalamjiak, A hybrid projective method for solving system of equilibrium problems with demicontractive mappings applicable in image restoration problems, *Math. Meth. Appl. Sci.* 43 (2020), 3413-3431.
- [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] L. Liu, S.Y. Cho, A Bregman projection algorithm with self adaptive step sizes for split variational inequality problems involving non-Lipschitz operators, *J. Nonlinear Var. Anal.* 8 (2024) 396-417.

- [22] W.R. Mann, Mean value methods in iteration, *Proc. Am. Math. Soc.* 4 (1953) 506-510.
- [23] P.E. Mainge, Convergence theorems for inertial KM-type algorithms, *J. Comput. Appl. Math.* 219 (2008), 223-236.
- [24] Z. Opial, Weak convergence of the sequence of successive approximations for nonexpansive mappings, *Bull. Amer. Math. Soc.* 73 (1967), 591-597.
- [25] M.O. Osilike, S.C. Aniagbosor, Weak and strong convergence theorems for fixed points of asymptotically nonexpansive mappings, *Math. Comput. Model.* 32 (2000) 1181-1191.
- [26] É. Picard, Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives, *J. Math. Pures Appl.* 6 (1890) 145-210.
- [27] B.T. Polyak, Some methods of speeding up the convergence of iteration methods, *USSR Comput. Math. Math. Phys.* 4 (1964) 1-17.
- [28] P. Paimsang, D. Yambangwai, T. Thianwan, A novel Noor iterative method of operators with property (E)(E) as concerns convex programming applicable in signal recovery and polynomiography, *Math. Methods Appl. Sci.* 47 (2024) 9571-9588.
- [29] X. Qin, An inertial Krasnosel'skiĭ-Mann iterative algorithm for accretive and nonexpansive mappings, *J. Nonlinear Convex Anal.* 26 (2025) 407-414.
- [30] X. Qin, A weakly convergent method for splitting problems with nonexpansive mappings, *J. Nonlinear Convex Anal.* 24 (2023) 1033-1043.
- [31] M. Raus, Y. Elshiaty, S. Petra, Accelerated Bregman divergence optimization with SMART: An information geometric point of view, *J. Appl. Numer. Optim.* 6 (2024) 1-40.
- [32] S. Suantai, P. Peeyada, D. Yambangwai, W. Chalamjiak, A parallel-viscosity-type subgradient extragradient-line method for finding the common solution of variational inequality problems applied to image restoration problems, *Mathematics* 8 (2020) 248.
- [33] Y. Shehu, J.C. Yao, Weak convergence of two-step inertial iteration for countable family of quasi-nonexpansive mappings, *Ann. Math. Sci. Appl.* 7 (2022) 259-279.
- [34] F. Su, L.L. Liu, X.H. Li, Q.L. Dong, A multi-step inertial asynchronous sequential algorithm for common fixed point problems, *J. Nonlinear Var. Anal.* 8 (2024) 473-484.
- [35] R. Suparatulatorn, W. Chalamjiak, A. Gibali, T. Moukhtonglang, A parallel Tseng's splitting method for solving common variational inclusion applied to signal recovery problems, *Adv. Differ. Equ.* 2021 (2021) 1-19.
- [36] R. Suparatulatorn, S. Suantai, W. Chalamjiak, Hybrid methods for a finite family of G -nonexpansive mappings in Hilbert spaces endowed with graphs, *AKCE Int. J. Graphs Comb.* 14 (2017) 101-111.
- [37] B. Tan, X. Qin, On relaxed inertial projection and contraction algorithms for solving monotone inclusion problems, *Adv. Comput. Math.* 50 (2024), 59.
- [38] B. Tan, X. Qin, S.Y. Cho, Revisiting subgradient extragradient methods for solving variational inequalities, *Numer. Algo.* 90 (2022), 1593-1615.
- [39] D. Yambangwai, S.A. Khan, H. Dutta, W. Chalamjiak, Image restoration by advanced parallel inertial forward-backward splitting methods, *Soft Comput.* 25 (2021) 6029-6042.