J. Nonlinear Funct. Anal. 2022 (2022) 9 https://doi.org/10.23952/jnfa.2022.9

MATHRES

Journal of Nonlinear Functional Analysis

Available online at http://jnfa.mathres.org

A HYBRID INERTIAL PARALLEL SUBGRADIENT EXTRAGRADIENT-LINE ALGORITHM FOR VARIATIONAL INEQUALITIES WITH AN APPLICATION TO IMAGE RECOVERY

PRONPAT PEEYADA, WATCHARAPORN CHOLAMJIAK*, DAMRONGSAK YAMBANGWAI

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

Abstract. In this paper, we introduce a hybrid inertial parallel subgradient extragradient-line algorithm for approximating a common solution of variational inequality problems with monotone and L-Lipschitz continuous mappings, where L is unknown. Under some suitable conditions, we prove the strong convergence of the algorithm. We also present some numerical examples to demonstrate the performance of our algorithm, which is better than the algorithms mentioned in the literature. The novelty of our algorithm is that the algorithm is resilient and efficient the number of subproblems is large. Our algorithm can be applied to image recovery problems when an image has common types of blur effects.

Keywords. Common variational inequality problems; Hybrid algorithm; Inertial method; Image recovery; Parallel subgradient extragradient-line algorithm.

1. Introduction and Preliminaries

Let H be a real Hilbert space with inner product $\langle .,. \rangle$ and induced norm $\| \, . \, \|$. Let C be a nonempty, closed, and convex subset of H. In this paper, we consider the variational inequality problem (VIP) that consists of finding a point $x^* \in C$ such that

$$\langle Ax^*, x - x^* \rangle \geq 0, \quad \forall x \in C, \tag{1.1}$$

where A is a mapping of H into H. We denote $VI(C,A)$ is the solution set of VIP (1.1).

It is well known that the VIP (1.1) is equivalent to the fixed point problem, which consists of finding a point $x^* \in C$ such that

$$x^* = P_C(x^* - \lambda Ax^*),$$

where λ is any positive real number. The VIP (1.1), which is a fundamental problem in nonlinear analysis and optimization theory, finds many real applications, such as signal processing,

*Corresponding author.

E-mail address: pronpat.pee@gmail.com (P. Peeyada), watcharaporn.ch@up.ac.th (W. Cholamjiak), damrongsak.ya@up.ac.th (D. Yambangwai).

Received October 23, 2021; Accepted March 4, 2022.

image recovery, transportation problems, economics, and engineering; see, e.g., [2, 4, 5, 6, 7, 20, 24, 26] and the references therein.

Recently, projection-based methods have been extensively investigated to solve VIP (1.1); see, e.g., [6, 8, 16, 22, 23]. An important projection method, which is called the Extragradient Method (EGM), was proposed by Korpelevich [21] in 1976; see also [3]. The method reads

$$\begin{cases} y_n = P_C(x_n - \lambda A x_n), \\ x_{n+1} = P_C(x_n - \lambda A y_n), \end{cases} \tag{1.2}$$

where $\lambda \in (0, \frac{1}{L})$, and P_C denotes the metric projection from H onto C.

In recent years, the EGM (1.2) has received great attention from many authors, who improved it in various ways; see, e.g., [8, 10, 12, 13, 22, 30, 32] and the references therein. In 2011, Censor *et al.* [10] improved the EGM (1.2) in Hilbert spaces. Their method, called the subgradient extragradient method (SEGM), reads as follows

$$\begin{cases} x_1 \in C, \\ y_n = P_C(x_n - \lambda A x_n), \\ T_n = \{w \in H : \langle x_n - \lambda A x_n - y_n, w - y_n\rangle \le 0\}, \\ x_{n+1} = P_{T_n}(x_n - \lambda A y_n). \end{cases} \tag{1.3}$$

In (1.3), the second projection P_C of the EGM (1.2) was replaced with a projection onto a half-space T_n which can be calculated easier more than a projection onto a complex closed convex set C. Under the assumptions of monotonicity and continuity of the operator A, Censor *et al.* [10] obtained weak convergence results based on (1.3).

Recently, Alvarez and Attouch [1], and Censor *et al.* [10] used the inertial extrapolation term to speed up the rate of convergence of the SEGM for solving (1.1) in Hilbert spaces. Their algorithm, called inertial subgradient extragradient method (ISEGM), reads as follows

$$\begin{cases} w_n = x_n + \alpha_n(x_n - x_{n-1}), \\ y_n = P_C(w_n - \tau A w_n), \\ T_n = \{x \in H | \langle w_n - \tau A w_n - y_n, x - y_n\rangle \le 0\}, \\ x_{n+1} = P_{T_n}(w_n - \tau A y_n), \end{cases}$$

where $\tau > 0$ and $\alpha_n \ge 0$ are suitable parameters. Under several appropriate conditions imposed on these parameters, weak convergence result was established. It deserves mentioning that, in the above algorithm, the Lipschitz constant is known.

Our interest in this paper is to study common solutions of variational inequality problems (CVIP). The CVIP is stated as follows: Let C be a nonempty, closed, and convex subset of H. Let $A_i : H \to H$, $i = 1, 2, \ldots, N$ be mappings. The CVIP is to find $x^* \in C$ such that

$$\langle A_i x^*, x - x^*\rangle \ge 0, \; \forall x \in C, \; i = 1, 2, \ldots, N. \tag{1.4}$$

If $N = 1$, CVIP (1.4) becomes VIP (1.1).

Recently, Suantai *et al.* [28] investigated the viscosity-type subgradient extragradient-line method, introduced by Shehu and Iyiola [21], to solve the CVIP (1.4). This algorithm is now called the parallel viscosity-type subgradient extragradient-line method (PVSEGM). The strong convergence theorem was proved when each of the operator A_i is a Lipschitz continuous monotone mapping whose Lipschitz constant is unknown. This algorithm reads as follows

$$\begin{cases} y_n^i = P_C(x_n - \lambda_n^i A_i x_n), \ \ \lambda_n^i = \rho^{l_n^i}, \\ (l_n^i \text{ is the smallest nonegative integer } l^i \text{ such that } \lambda_n^i \|A_i x_n - A_i y_n^i\| \le \mu \|r_{\rho^{l_n^i}}(x_n)\|), \\ z_n^i = P_{T_n^i}(x_n - \lambda_n^i A y_n^i), \\ x_{n+1} = \alpha_n^0 f(x_n) + \sum_{i=1}^{N} \alpha_n^i z_n^i, \ \ n \ge 1, \end{cases} \tag{1.5}$$

where $T_n^i = \{z \in H : \langle x_n - \lambda_n^i A_i x_n - y_n^i, z - y_n^i\rangle \le 0\}$ with $\rho, \mu \in (0,1)$ and $\{\alpha_n\}_{n=1}^{\infty} \subseteq (0,1)$. The sequence $\{x_n\}_{n=1}^{\infty}$ generated by (1.5) was proved that it converges strongly to $x^* \in$ VI(C,A), where $x^* = P_{VI(C,A)} f(x^*)$ is the unique solution of the variational inequality

$$\left\langle (I-f)x^*, x - x^* \right\rangle \ge 0, \quad \forall x \in VI(C,A).$$

Since f is a strict contraction, its Lipschitz constant k is, in fact, strictly less than 1 under the following conditions

$$(C_1) \ \ \lim_{n\to\infty} \alpha_n^0 = 0 \ \text{ and } \ (C_2) \ \ \sum_{n=1}^{\infty} \alpha_n^0 = \infty.$$

The advantage of the PVSEGM was presented to solve the problem of multiblur effects in an image restoration. The image quality was improved sharper by using the PVSEGM in the resolution of common resolution (VIP) problems.

In this paper, motivated and inspired by the works in literature, and by the ongoing research in these directions, we combine hybrid inertial techniques with a parallel subgradient extragradient-line method for solving CVIP (1.4). Numerical experiments are also conducted to illustrate the efficiency of the proposed algorithms. Moreover, the problem of multiblur effects in an image is solved by our algorithm.

2. Main Results

In this section, we propose the hybrid inertial parallel subgradient extragradient-line method for solving CVIP (1.4). Let H be a real Hilbert space, and let C be a nonempty, closed, and convex subset of H. Let $A_i : H \to H$ be monotone mappings and L_i-Lipschitz continuous on H, but L_i is unknown for all $i = 1,2,\ldots,N$ such that $\Upsilon = \cap_{i=1}^{N} VI(C,A_i) \neq \emptyset$. Suppose that $\{x_n\}_{n=1}^{\infty}$ is generated in the following Algorithm 2.1:

Algorithm 2.1. *Take $\rho \in (0,1)$ and $\mu \in (0,1)$. Select arbitrary points $x_0, x_1 \in H$, and $\{\theta_n\} \subseteq [0,\theta]$ for some $\theta \in [0,1)$. Set $n := 1$.*

Step 1 *Compute $t_n = x_n + \theta_n(x_n - x_{n-1})$, $\forall n \ge 1$.*
Step 2 *Compute y_n^i, for all $i = 1,2,\ldots,N$, by $y_n^i = P_C(t_n - \lambda_n^i A_i t_n)$, $\forall n \ge 1$, where $\lambda_n^i = \rho^{k_n^i}$, and k_n^i is the smallest nonnegative integer such that*

$$\lambda_n^i \parallel A_i t_n - A_i y_n^i \parallel \le \mu \parallel t_n - y_n^i \parallel . \tag{2.1}$$

Step 3 *Compute* $z_n^i = P_{T_n^i}(t_n - \lambda_n^i A_i y_n^i)$, *where* $T_n^i := \{z \in H : \langle t_n - \lambda_n^i A_i t_n - y_n^i, z - y_n^i \rangle \leq 0\}$.
Step 4 *Compute*

$$\bar{u_n} = \alpha_n^0(t_n) + \sum_{i=1}^{N} \alpha_n^i z_n^i, \quad n \geq 1, \tag{2.2}$$

where $\alpha_n^i \in (0,1)$, $\forall i = 1,2,\ldots,N$ *and* $\sum_{i=0}^{N} \alpha_n^i = 1$, $\forall n \in N$.

Step 5 *Compute* $x_{n+1} = P_{C_{n+1}} x_1$, *where* $C_{n+1} := \{z \in C_n : \| \bar{u_n} - z \| \leq \| t_n - z \|\}$. *Set* $n+1 \to n$ *and go to Step 1.*

Lemma 2.2. *There exists a nonnegative integer* k_n^i *satisfying (2.1).*

Proof. We first show that $\{t_n\}$ is bounded. Since Υ is a nonempty, closed, and convex subset of H, there exists a unique $v \in \Upsilon$ such that $v = P_\Upsilon x_1$. From $x_n = P_{C_n} x_1$ and $x_{n+1} \in C_n$, for all $n \geq 1$, we obtain

$$\| x_n - x_1 \| \leq \| x_{n+1} - x_1 \|, \quad \forall n \geq 1. \tag{2.3}$$

On the other hand, as $\Upsilon \subset C_n$, we obtain

$$\| x_n - x_1 \| \leq \| v - x_1 \|, \quad \forall n \geq 1. \tag{2.4}$$

It follows from (2.3) and (2.4) that $\{x_n\}$ is bounded. By the definition of $\{x_n\}$, we obtain that $\{t_n\}$ is also bounded. For each $i = 1,2,\ldots,N$ and $n \in \mathbb{N}$, we let $y_{k^i}^i = P_C(t_n - \rho^{k^i} A_i t_n)$ for all $k^i \in \mathbb{N}$. We divide the proof into two cases as follows.

Case I: for each $i = 1,2,\ldots,N$, if $\| t_n - y_{n_0^i}^i \| = 0$ for some $n_0^i \geq 1$, then there exist k_n^i such that $k_n^i \leq n_0^i$ satisfying (2.1).

Case II: for each $i = 1,2,\ldots,N$, if $\| t_n - y_{n_1^i}^i \| \neq 0$ for all $n_1^i \geq 1$, then we assume the contrary that $\rho^{n_1^i} \| A_i t_n - A_i y_{n_1^i}^i \| > \mu \| t_n - y_{n_1^i}^i \|$. From [15, Lemma 6.3] and the fact that $\rho \in (0,1)$, we obtain

$$\begin{aligned} \| A_i t_n - A_i y_{n_1^i}^i \| &> \frac{\mu}{\rho^{n_1^i}} \| t_n - y_{n_1^i}^i \| \\ &\geq \frac{\mu}{\rho^{n_1^i}} \min\{1, \rho^{n_1^i}\} \| t_n - y_0^i \| \\ &= \mu \| t_n - y_0^i \| . \end{aligned} \tag{2.5}$$

By using the continuity of P_C and the fact that $\{t_n\}$ is bounded, we have that $y_{n_1^i}^i = P_C(t_n - \rho^{n_1^i} A_i t_n) \to P_C(t_n)$, $n_1^i \to \infty$ for all $i = 1,2,\ldots,N$. We consider two cases: $t_n \in C$ and $t_n \notin C$.

(i) If $t_n \in C$, then $t_n = P_C(t_n)$. Now, since $\| t_n - y_{n_1^i}^i \| \neq 0$ and $0 < \rho^{n_1^i} \leq 1$, it follows from [15, Lemma 6.3] that

$$0 < \| t_n - y_{n_1^i}^i \| \leq \max\{1, \rho^{n_1^i}\} \| t_n - y_0^i \| = \| t_n - y_0^i \| .$$

Taking $n_1^i \to \infty$ in (2.5) for each $i = 1,2,\ldots,N$, we have that $0 = \| A_i t_n - A_i t_n \| \geq \mu \| t_n - y_0^i \| > 0$. This is a contradiction, and hence (2.1) is well defined.

(ii) If $t_n \notin C$, then, for each $i = 1,2,...,N$, $\rho^{n_1^i} \| A_i t_n - A_i y_n^i \| \to 0$, as $n_1^i \to \infty$ while

$$\begin{aligned}\lim_{n_1^i\to\infty} \mu \| t_n - y^i_{n_1^i} \| &= \mu \lim_{n_1^i\to\infty} \| t_n - P_C(t_n - \rho^{n_1^i} A_i t_n) \| \\ &= \mu \| t_n - P_C(t_n) \| > 0.\end{aligned}$$

This is a contradiction due to $t_n \neq P_C(t_n)$. Therefore, the linesearch in Algorithm 2.1 is well defined and implementable. □

Theorem 2.3. *Assume that the conditions hold: (i) $\sum_{n=1}^{\infty} \theta_n \| x_n - x_{n-1} \| < \infty$, (ii) $\liminf_{n\to\infty} \alpha_n^i > 0$ for all $i = 1,2,...,N$. Then the sequence $\{x_n\}$ generated by Algorithm 2.1 converges strongly to $z \in \Upsilon$.*

Proof. We split the proof into five steps.

Step 1. Show that $\{x_n\}$ is well defined.

From $C_1 = C$, we see that C_1 is closed and convex. Assume that C_n is closed and convex. From the definition of C_{n+1} and [25, Lemma 1.3], we obtain that C_{n+1} is closed and convex. Let $x^* \in \Upsilon$ and $s_n^i = t_n - \lambda_n^i A_i y_n^i, \forall n \geq 1, i = 1,2,..,N$. Then,

$$\begin{aligned}\| z_n^i - x^* \|^2 &= \| P_{T_n^i}(s_n^i) - x^* \|^2 \\ &= \| P_{T_n^i}(s_n^i) - s_n^i \|^2 + 2\langle P_{T_n^i}(s_n^i) - s_n^i, s_n^i - x^* \rangle + \| s_n^i - x^* \|^2 . \end{aligned} \tag{2.6}$$

From $x^* \in \Upsilon \subseteq C \subseteq T_n^i$ and the characterization of the metric projection $P_{T_n^i}$, we have

$$2 \| s_n^i - P_{T_n^i}(s_n^i) \|^2 + 2\langle P_{T_n^i}(s_n^i) - s_n^i, s_n^i - x^* \rangle = 2\langle s_n^i - P_{T_n^i}(s_n^i), x^* - P_{T_n^i}(s_n^i) \rangle \leq 0. \tag{2.7}$$

This implies that $\| s_n^i - P_{T_n^i}(s_n^i) \|^2 + 2\langle P_{T_n^i}(s_n^i) - s_n^i, s_n^i - x^* \rangle \leq - \| s_n^i - P_{T_n^i}(s_n^i) \|^2$. By the definition of Algorithm 2.1, (2.6), and (2.7), we have

$$\begin{aligned}\| z_n^i - x^* \|^2 &\leq \| s_n^i - x^* \|^2 - \| s_n^i - z_n^i \|^2 \\ &= \| (t_n - x^*) - \lambda_n^i A_i y_n^i \|^2 - \| (t_n - z_n^i) - \lambda_n^i A_i y_n^i \|^2 \\ &= \| t_n - x^* \|^2 - \| t_n - z_n^i \|^2 + 2\lambda_n^i \langle -t_n + x^*, A_i y_n^i \rangle + 2\lambda_n^i \langle t_n - z_n^i, A_i y_n^i \rangle \\ &= \| t_n - x^* \|^2 - \| t_n - z_n^i \|^2 + 2\lambda_n^i \langle x^* - z_n^i, A_i y_n^i \rangle. \end{aligned} \tag{2.8}$$

By the monotonicity of the operator A_i, we have

$$\begin{aligned} 0 &\leq \langle A_i y_n^i - A_i x^*, y_n^i - x^* \rangle \\ &= \langle A_i y_n^i, y_n^i - x^* \rangle - \langle A_i x^*, y_n^i - x^* \rangle \\ &\leq \langle A_i y_n^i, y_n^i - z_n^i \rangle + \langle A_i y_n^i, z_n^i - x^* \rangle. \end{aligned}$$

Thus

$$\langle x^* - z_n^i, A_i y_n^i \rangle \leq \langle A_i y_n^i, y_n^i - z_n^i \rangle. \tag{2.9}$$

Substituting (2.9) in (2.8), we obtain

$$\begin{aligned}\| z_n^i - x^* \|^2 &\leq \| t_n - x^* \|^2 - \| t_n - z_n^i \|^2 + 2\lambda_n^i \langle A_i y_n^i, y_n^i - z_n^i \rangle \\ &= \| t_n - x^* \|^2 - \| t_n - y_n^i \|^2 - \| y_n^i - z_n^i \|^2 - 2\langle t_n - y_n^i, y_n^i - z_n^i \rangle \\ &\quad + 2\lambda_n^i \langle A_i y_n^i, y_n^i - z_n^i \rangle \\ &= \| t_n - x^* \|^2 - \| t_n - y_n^i \|^2 - \| y_n^i - z_n^i \|^2 \\ &\quad + 2\langle t_n - \lambda_n^i A_i y_n^i - y_n^i, z_n^i - y_n^i \rangle. \end{aligned} \tag{2.10}$$

Observe that

$$\begin{aligned}\langle t_n-\lambda_n^iA_iy_n^i-y_n^i,z_n^i-y_n^i\rangle &= \langle t_n-\lambda_n^iA_it_n-y_n^i,z_n^i-y_n^i\rangle+\langle\lambda_n^iA_it_n-\lambda_n^iA_iy_n^i,z_n^i-y_n^i\rangle\\ &\leq \langle\lambda_n^iA_it_n-\lambda_n^iA_iy_n^i,z_n^i-y_n^i\rangle.\end{aligned}$$

Using the last inequality in (2.10), we have that

$$\begin{aligned}\| z_n^i-x^*\|^2 &\leq \| t_n-x^*\|^2-\| t_n-y_n^i\|^2-\| y_n^i-z_n^i\|^2+2\langle\lambda_n^iA_it_n-\lambda_n^iA_iy_n^i,z_n^i-y_n^i\rangle\\ &\leq \| t_n-x^*\|^2-\| t_n-y_n^i\|^2-\| y_n^i-z_n^i\|^2+2\lambda_n^i\| A_it_n-A_iy_n^i\|\| z_n^i-y_n^i\|\\ &\leq \| t_n-x^*\|^2-\| t_n-y_n^i\|^2-\| y_n^i-z_n^i\|^2+2\mu\| t_n-y_n^i\|\| z_n^i-y_n^i\|\\ &\leq \| t_n-x^*\|^2-\| t_n-y_n^i\|^2-\| y_n^i-z_n^i\|^2+\mu(\| t_n-y_n^i\|^2+\| z_n^i-y_n^i\|^2)\\ &= \| t_n-x^*\|^2-(1-\mu)(\| t_n-y_n^i\|^2+\| y_n^i-z_n^i\|^2), \qquad (2.11)\end{aligned}$$

which implies that

$$\|\bar{u}_n-x^*\|^2\leq\alpha_n^0\| t_n-x^*\|^2+\sum_{i=1}^N\alpha_n^i\| z_n^i-x^*\|^2\leq\| t_n-x^*\|^2.$$

This shows that $\|\bar{u}_n-x^*\|\leq\| t_n-x^*\|$. Hence, $x^*\in C_n$, $\forall n\geq 1$. This implies that $\{x_n\}$ is well-defined.

Step 2. Show that $x_n\to\omega\in C$ as $n\to\infty$.

For $k>j$, since $x_k=P_{C_k}x_1\in C_k\subset C_j$, we have $\| x_k-x_j\|^2\leq\| x_k-x_1\|^2-\| x_j-x_1\|^2$. By (2.3), (2.4), and the fact that $\{x_n\}$ is bounded and nonincreasing, we have that $\lim_{n\to\infty}\| x_n-x_1\|$ exists. Hence, $\| x_k-x_j\|\to 0$ as $k,j\to\infty$, which means that $\{x_n\}$ is a Cauchy sequence. Hence, there exists $\omega\in C$ such that $x_n\to\omega$ as $n\to\infty$. In particular, we have $\lim_{n\to\infty}\| x_{n+1}-x_n\|=0$.

Step 3. Show that $\lim_{n\to\infty}\| x_n-y_n^i\|=\lim_{n\to\infty}\| y_n^i-z_n^i\|=0$ for all $i=1,2,\ldots,N$.

Let $x^*\in\Upsilon$. Then, we have from (2.2), (2.11), and [9, Lemma 2.1] that

$$\begin{aligned}\|\bar{u}_n-x^*\|^2 &= \|\alpha_n^0(t_n)+\sum_{i=1}^N\alpha_n^iz_n^i-x^*\|^2\\ &\leq \alpha_n^0\| t_n-x^*\|^2+\sum_{i=1}^N\alpha_n^i\| z_n^i-x^*\|^2\\ &= \| t_n-x^*\|^2-(1-\mu)\sum_{i=1}^N\alpha_n^i(\| t_n-y_n^i\|^2+\| y_n^i-z_n^i\|^2)\\ &= \| x_n-x^*\|^2+\theta_n^2\| x_n-x_{n-1}\|^2+2\langle x_n-x^*,\theta_n(x_n-x_{n-1})\rangle\\ &\quad -(1-\mu)\sum_{i=1}^N\alpha_n^i(\| x_n-y_n^i\|^2+\theta_n^2\| x_n-x_{n-1}\|^2\\ &\quad +2\langle x_n-y_n^i,\theta_n(x_n-x_{n-1})\rangle+\| y_n^i-z_n^i\|^2). \qquad (2.12)\end{aligned}$$

Since $x_{n+1}\in C_{n+1}\subset C_n$, we have

$$\|\bar{u}_n-x_{n+1}\|\leq\| t_n-x_{n+1}\|\leq\theta_n\| x_n-x_{n-1}\|+\| x_n-x_{n+1}\|\to 0,\ \text{as } n\to\infty.$$

This implies that

$$\|\bar{u}_n-x_n\| \leq \|\bar{u}_n-x_{n+1}\|+\| x_{n+1}-x_n\|\to 0,\ \text{as } n\to\infty. \qquad (2.13)$$

It follows form (2.12) that

$$\begin{aligned}&(1-\mu)\sum_{i=1}^{N}\alpha_n^i\big(\parallel x_n-y_n^i\parallel^2+\parallel y_n^i-z_n^i\parallel^2\big)\\ &\leq\parallel x_n-x^*\parallel^2-\parallel \bar{u}_n-x^*\parallel^2+\theta_n^2\parallel x_n-x_{n-1}\parallel^2+2\langle x_n-x^*,\theta_n(x_n-x_{n-1})\rangle\\ &\quad-(1-\mu)\sum_{i=1}^{N}\alpha_n^i\big(\theta_n^2\parallel x_n-x_{n-1}\parallel^2+2\langle x_n-y_n^i,\theta_n(x_n-x_{n-1})\rangle\big).\end{aligned}$$

By our assumptions (i), (ii) and (2.13), we obtain $\lim_{n\to\infty}\parallel y_n^i-z_n^i\parallel=\lim\limits_{n\to\infty}\parallel x_n-y_n^i\parallel=0$, $\forall i=1,2,\ldots,N$.

Step 4. Show that $\omega\in\Upsilon$.

Since $\parallel x_n-t_n\parallel=\theta_n\parallel x_n-x_{n-1}\parallel\to0$ and $x_n-y_n^i\to0$, we have $t_n-\omega$ and $y_n^i\to\omega$. Since $y_n^i\in C$, we obtain $\omega\in C$. For all $x\in C$, using the property of the projection P_C, we have (Since A_i is monotone)

$$\begin{aligned}0&\leq\langle y_n^i-t_n+\lambda_n^iA_it_n,x-y_n^i\rangle\\ &=\langle y_n^i-t_n,x-y_n^i\rangle+\langle\lambda_n^iA_it_n,x-x_n\rangle+\langle\lambda_n^iA_it_n,x_n-y_n^i\rangle\\ &=\langle y_n^i-x_n,x-y_n^i\rangle+\theta_n\langle x_n-x_{n-1},x-y_n^i\rangle.\end{aligned}\tag{2.14}$$

From [29, Remark 3.2], we know that $\inf\limits_{n\geq1}\lambda_n^i>0$ for all $i=1,2,\ldots,N$. Taking $n\to\infty$ in (2.14) yields that $\langle A_i\omega,x-\omega\rangle\geq0$, $\forall x\in C$. This implies that $\omega\in VI(C,A_i)$ for all $i=1,2,\ldots,N$. This completes the proof. □

Remark 2.4. Condition (i) is easily implemented in numerical computation since the value $\|x_n-x_{n-1}\|$ is known before choosing θ_n. Indeed, the parameter θ_n can be chosen such that

$$\theta_n=\begin{cases}\frac{\varepsilon_n}{\|x_n-x_{n-1}\|} & if\ x_n\neq x_{n-1},\\ \tau & otherwise,\end{cases}$$

where $\sum_{n=1}^{\infty}\varepsilon_n<\infty$ and $\tau\geq0$.

Base on the choice of the inertial parameter θ_n and the relation between Algorithm 2.1 where $A_i=A$ for all $i=1,2,\ldots,N$, Algorithm 2.1 is reduced to the following hybrid inertial subgradient extragradient algorithm.

Algorithm 2.5. *Take $\rho\in(0,1)$ and $\mu\in(0,1)$. Select arbitrary points $x_0,x_1\in H$ and $\{\theta_n\}\subseteq[0,\theta]$ for some $\theta\in[0,1)$. Set $n:=1$.*

Step 1 *Compute $t_n=x_n+\theta_n(x_n-x_{n-1})$, $\forall n\geq1$.*
Step 2 *Compute y_n by $y_n=P_C(t_n-\lambda_nAt_n)$, $\forall n\geq1$, where $\lambda_n=\rho^{k_n}$ and k_n is the smallest nonnegative integer such that $\lambda_n\parallel At_n-Ay_n\parallel\leq\mu\parallel t_n-y_n\parallel$.*
Step 3 *Compute $z_n=P_{T_n}(t_n-\lambda_nAy_n)$, where $T_n:=\{z\in H:\langle t_n-\lambda_nAt_n-y_n,z-y_n\rangle\leq0\}$.*
Step 4 *Compute $\bar{u_n}=\alpha_nt_n+(1-\alpha_n)z_n$, where $\alpha_n\in(0,1)$.*
Step 5 *Compute $x_{n+1}=P_{C_{n+1}}x_1$, where $C_{n+1}:=\{z\in C_n:\parallel\bar{u_n}-z\parallel\leq\parallel t_n-z\parallel\}$. Set $n+1\to n$ and go to Step1.*

We now give an example in Euclidean space $\mathbb{R}^3$ to support the our main theorem.

Example 2.6. Let $A_1, A_2 : \mathbb{R}^3 \to \mathbb{R}^3$ be defined by $A_1 x = 4x$ and $A_2 x = \begin{pmatrix} 10 & -5 & 5 \\ -5 & 10 & -5 \\ 5 & -5 & 10 \end{pmatrix}$ for all $x = (x_1, x_2, x_3) \in \mathbb{R}^3$. Let $C = \{x \in \mathbb{R}^3 | x_1^2 + x_2^2 + x_3^2 \leq 4\}$. The stopping criterion is defined by $\| x_n - x_{n-1} \| < 10^{-15}$.

(1) Choose

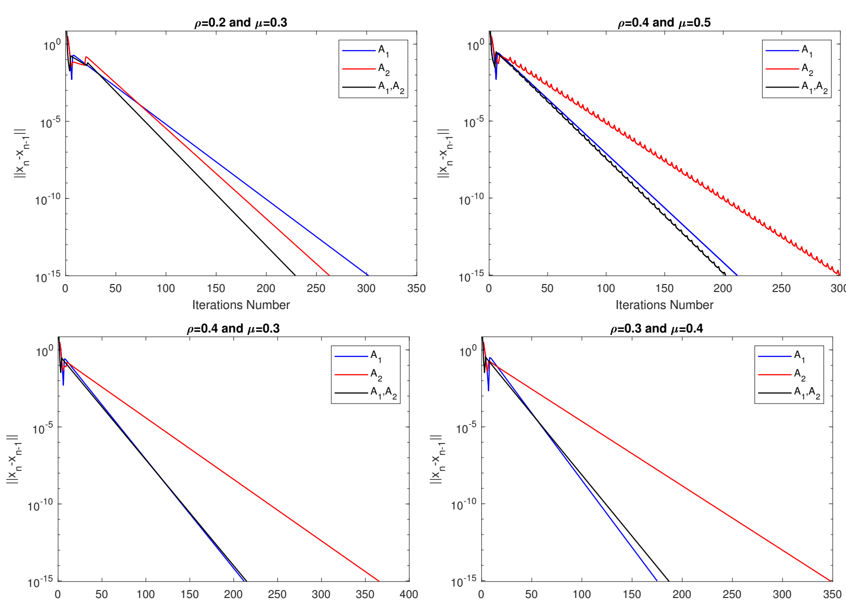
$$\theta_n = \begin{cases} 0.15 & \text{if } x_n \neq x_{n-1} \text{ and } n \leq 1000 \\ \frac{1}{n^2 \|x_n - x_{n-1}\|} & \text{if } x_n \neq x_{n-1} \text{ and } n > 1000 \\ 0 & \text{Otherwise,} \end{cases}$$

$\alpha_n^0 = \frac{n^2+1}{3n^2+n}$, and $\alpha_n^1 = 1 - \alpha_n^0$ for applying our Algorithm 2.1 in two cases when we put $A_i = A_1$ for all $i = 1, 2, \ldots, N$ in the first case and the second $A_i = A_2$ for all $i = 1, 2, \ldots, N$. Choose $\alpha_n^0 = \frac{n^2+1}{100n^2+n}$, $\alpha_n^1 = \frac{50n+2}{100n+1}$, and $\alpha_n^2 = 1 - (\alpha_n^0 + \alpha_n^1)$ for the third case that we put A_1, A_2 in our Algorithm 2.1.

(2) Choose $\alpha_n^0 = \frac{1}{(n+1)^{0.3}}$, $\alpha_n^1 = \frac{1}{2n}$, and $\alpha_n^2 = 1 - (\alpha_n^0 + \alpha_n^1)$ for PVSEGM in [28, Theorem 1] to compare the convergence of our Algorithm 2.1.

Table 1: Comparison of the methods in Theorem 2.3 and [28, Theorem 1] of Example 2.6 by choosing $x_0 = (-2, -4, 1)$ and $x_1 = (-1, 7, 6)$.

	A_1		A_2		A_1, A_2	
	CPU Time	Iter.No.	CPU Time	Iter.No.	CPU Time	Iter.No.
Algorithm 2.1						
$\rho = 0.2$, $\mu = 0.3$	0.0000049	302	0.0000226	263	0.0000392	229
$\rho = 0.4$, $\mu = 0.5$	0.0000055	212	0.0000335	300	0.000029	202
$\rho = 0.4$, $\mu = 0.3$	0.0000054	212	0.0000169	366	0.000026	215
$\rho = 0.3$, $\mu = 0.4$	0.0000048	175	0.0000163	348	0.000024	187
PVSEGM						
$\rho = 0.2$, $\mu = 0.1$	0.0000056	591	0.0000086	505	0.0000179	506



Figures 1-4: Error plots for the Table 1 in Example 2.6.

Remark 2.6 From Table 1 and Figures 1-4, we see that

(i) it can be clearly seen that the common solution of CVIP (1.4) with $N=2$ obtain the better number of iterations than the average iteration of $N=1$;

(ii) for the CPU Time of three in four cases when the parameters ρ and μ are different, we find that the case $N=2$ converges faster than $N=1$;

(iii) for the comparison between our Algorithm 2.1 and PVSEGM, we see that our Algorithm 2.1 performs the good CPU Time and number of iterations more than PVSEGM for each of all cases.

3. Application to Image Restoration Problems

The image restoration problem is the recovering process of a degraded version, which is a blurred and noisy image. This problem can be formulated in the linear equation system as follows:

$$b = Bx + \upsilon, \tag{3.1}$$

where $x \in \mathbb{R}^{n\times 1}$ is an original image, $b \in \mathbb{R}^{m\times 1}$ is the unknown image, υ is additive noise, and $B \in \mathbb{R}^{m\times n}$ is the blurring operation. The main goal of image restoration problem (3.1) is to find the original image x. In some case, finding $x = B^{-1}(b-\upsilon)$ maybe a difficult task, thus finding the solution x by mean of convex minimization can overcome such difficulty, which is known as the following least squares (LS) problem

$$\min_x \frac{1}{2}\|b - Bx\|_2^2, \tag{3.2}$$

where $\|.\|$ is ℓ_2-norm defined by $\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$. The solution of (3.2) can be estimated by many well known iteration methods [14, 18, 19, 31].

The main goal in digital image restoration is to find the unknown image that we do not know which one is the blurring matrix of this unknown image. This problem can be considered in the system of least squares problems:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|B_1x - b_1\|_2^2, \min_{x\in\mathbb{R}^n} \frac{1}{2}\|B_2x - b_2\|_2^2, \ldots, \min_{x\in\mathbb{R}^n} \frac{1}{2}\|B_Nx - b_N\|_2^2 \tag{3.3}$$

where x is the original true image, B_i is the blurred matrix, b_i is the blurred image by the blurred matrix B_i for all $i = 1,2,\ldots,N$. For solving (3.3), we can apply our main Algorithm 2.1 by setting $A_ix = B_i^T(B_ix - b_i)$ for all $x \in \mathbb{R}^n$ in Algorithm 2.1 since $B_i^T(B_ix - b_i)$ is Lipschitz continuous for each $i = 1,2,\ldots,N$. This algorithm reads as follows:

$$\begin{cases} t_n = x_n + \theta_n(x_n - x_{n-1}), \ \ \forall n \geq 1, \\ y_n^i = P_C(t_n - \lambda_n^i B_i^T(B_it_n - b_i)), \ \ \forall n \geq 1 \text{ and } \forall i = 1,2,\ldots,N, \\ (l_n^i \text{ is the smallest nonnegative integer such that} \\ \lambda_n^i \parallel B_i^T(B_it_n - b_i) - B_i^T(B_iy_n^i - b_i) \parallel \leq \mu \parallel t_n - y_n^i \parallel), \\ z_n^i = P_{T_n^i}(t_n - \lambda_n^i B_i^T(B_iy_n^i - b_i)), \\ \bar{u}_n = \alpha_n^0(t_n) + \sum_{i=1}^N \alpha_n^i z_n^i, \ n \geq 1, \\ x_{n+1} = P_{C_{n+1}}x_1, \end{cases}$$

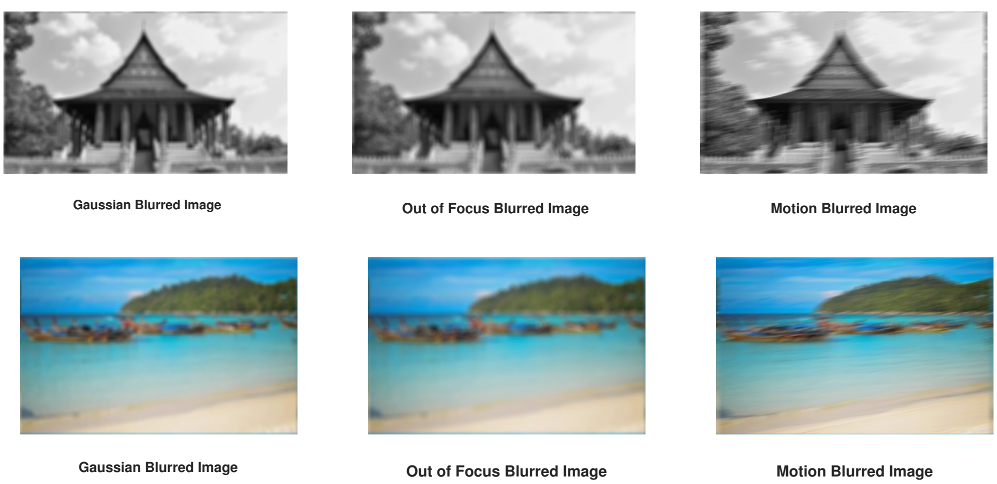
where $T_n^i = \{z \in H | \langle t_n - \lambda_n^i B_i t_n - y_n^i, z - y_n^i \rangle \leq 0\}$, $C_{n+1} = \{z \in C_n | \parallel \bar{u}_n - z \parallel \leq \parallel t_n - z \parallel\}$, $\rho, \mu, \alpha_n^i \in (0,1)$, and $\{\theta_n\} \subseteq [0, \theta]$ for some $\theta \in [0,1)$.

We will show the efficiency of our Algorithm 2.1 in image deblurring for the following three blur types:
Type 1: Gaussian blur of filter size 9×9 with standard deviation $\sigma = 4$ (blur matrix B_1).
Type 2: Out of focus blur (Disk) with radius $r = 6$ (blur matrix B_2).
Type 3: Motion blur specifying with motion length of 21 pixels (len $= 21$) and motion orientation 11° ($\theta = 11$) (blur matrix B_3).

The original Grey and RGB images are show in Figures 5-6.

Figures 5-6: The original Grey and RGB image of sizes 276×490, and $280 \times 440 \times 3$, respectively.

The different types of blurred Grey and RGB images degraded by the blurring matrices B_1, B_2 and B_3 are shown in figures 7-12.



Figures 7-12: The degraded Grey and RGB images by blurred matrices B_1, B_2, and B_3, respectively.

We apply the PVSEGM amd our Algorithm 2.1 to obtain the solution of deblurring problem with the three blurring matrices B_1, B_2, and B_3. The results of the PVSEGM and our Algorithm 2.1 are considered in following seven cases:
Case I: Inputting B_1 on the PVSEGM and Algorithm 2.1.
Case II: Inputting B_2 on the PVSEGM and Algorithm 2.1.
Case III: Inputting B_3 on the PVSEGM and Algorithm 2.1.

Case IV: Inputting B_1 and B_2 on the PVSEGM and Algorithm 2.1.
Case V: Inputting B_1 and B_3 on the PVSEGM and Algorithm 2.1.
Case VI: Inputting B_2 and B_3 on the PVSEGM and Algorithm 2.1.
Case VII: Inputting B_1, B_2 and B_3 on the PVSEGM and Algorithm 2.1.

The following parameters are used for our algorithm:

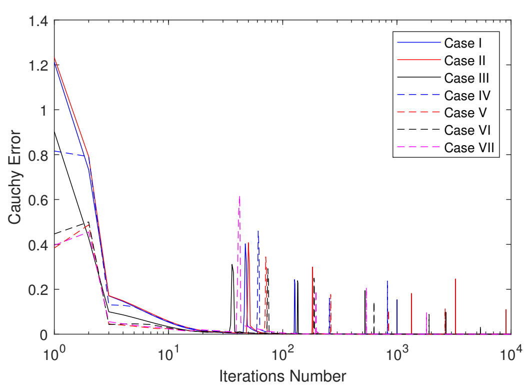
$$\theta_n = \begin{cases} 0.12 & \text{if } x_n \neq x_{n-1} \text{ and } n \leq 10,000 \\ \frac{1}{n^2\|x_n - x_{n-1}\|} & \text{if } x_n \neq x_{n-1} \text{ and } n > 10,000 \\ 0 & \text{Otherwise,} \end{cases}$$

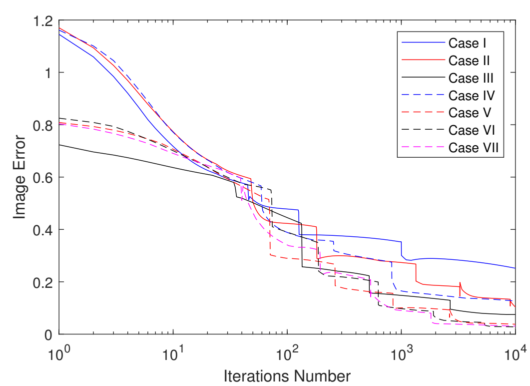
$\rho = 0.5$, and $\mu = 0.35$. We choose $\mu = 0.95$, $\rho = 0.5$, $\alpha_n^0 = 1 - \frac{3n}{3n+1}$, $\alpha_n^1 = \frac{n}{3n+1}$, $\alpha_n^2 = \frac{n}{3n+1}$, and $\alpha_n^3 = 1 - \alpha_n^0 - \alpha_n^1 - \alpha_n^2$ for PVSEGM.

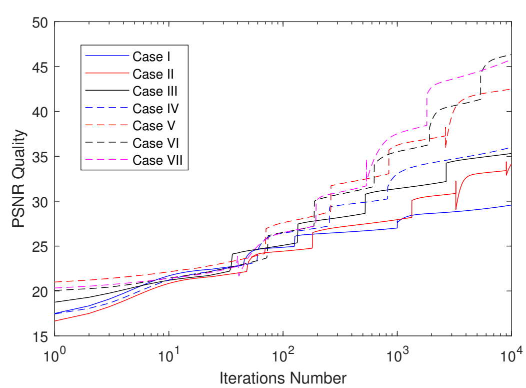
Table 2: Comparison of the number of iterations in Grey images.

Inputting	PSNR of 10000^{th}		Number of Iterations 33 PSNR	
	PVSEGM	Our Algorithm	PVSEGM	Our Algorithm
B_1	24.70720	29.57263	4921^{th}	50^{th}
B_2	26.47867	34.15647	2775^{th}	58^{th}
B_3	29.50780	35.32024	801^{th}	36^{th}
B_1, B_2	28.59585	36.01784	975^{th}	60^{th}
B_1, B_3	32.37244	42.50473	446^{th}	62^{th}
B_2, B_3	33.47745	46.33505	538^{th}	73^{th}
B_1, B_2, B_3	34.41830	45.79034	411^{th}	52^{th}

Moreover, the Cauchy error, the figure error, and the peak signal-to-noise ratio (PSNR) for recovering the processes of the degraded Grey images by using the proposed method within the first 10000^{th} iterations are shown in Figures 13-15.





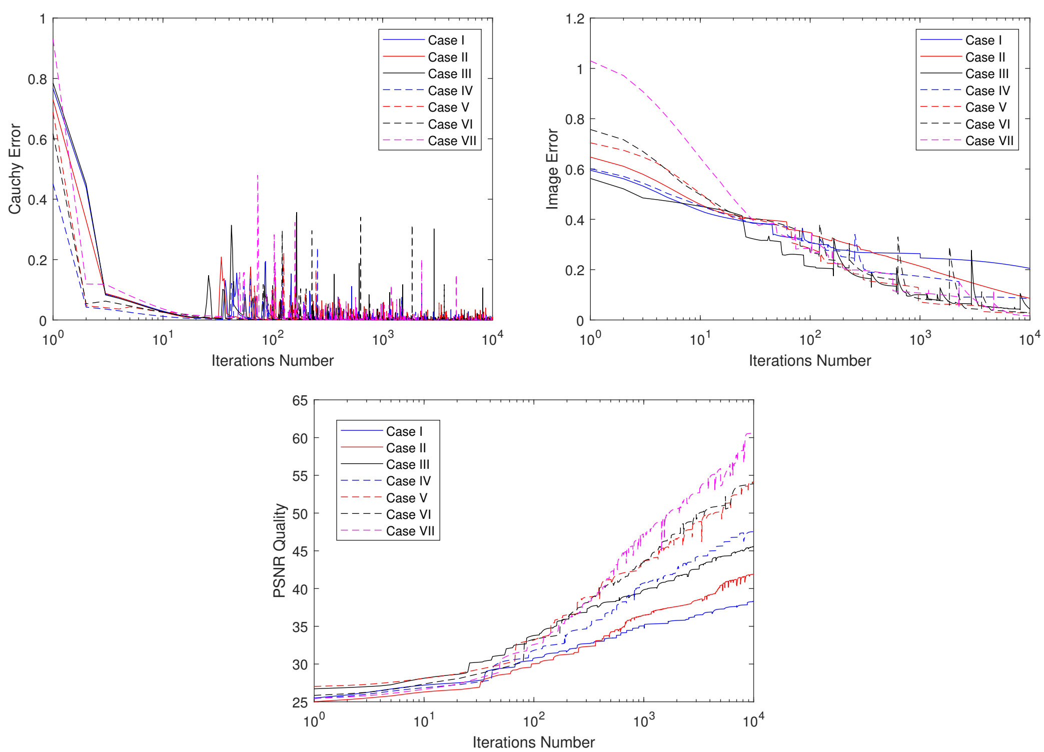


Figures 13-15: Cauchy error, Figure error, and PSNR quality plots of the proposed iteration in all cases of Grey images.

Table 3: Comparison of the number of iterations in RGB images.

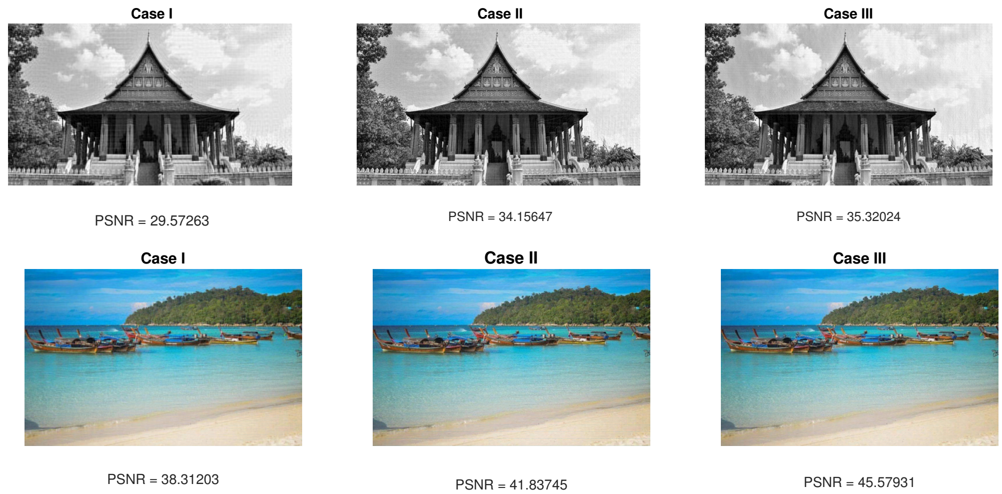
Inputting	PSNR of 10000^{th}		Number of Iterations 33 PSNR	
	PVSEGM	Our Algorithm	PVSEGM	Our Algorithm
B_1	33.47997	38.31203	6816^{th}	385^{th}
B_2	34.13544	41.83745	5800^{th}	364^{th}
B_3	37.89834	45.57931	1014^{th}	86^{th}
B_1, B_2	37.46071	47.54648	1253^{th}	190^{th}
B_1, B_3	41.57133	54.15965	509^{th}	86^{th}
B_2, B_3	41.77308	53.88841	634^{th}	87^{th}
B_1, B_2, B_3	43.52842	60.59668	474^{th}	122^{th}

Moreover, the Cauchy error, the figure error and the peak signal-to-noise ratio (PSNR) for recovering processes of the degraded RGB images by using the proposed method within the first 10000^{th} iterations are shown in Figures 16-18.



Figures 16-18: Cauchy error, Figure error, and PSNR quality plots of the proposed iteration in all cases of RGB images.

The figures of deblurring when the 10000^{th} iterations is the stopping criterion are shown in Figures 19-32 that be composed of the restored image and its PSNR.

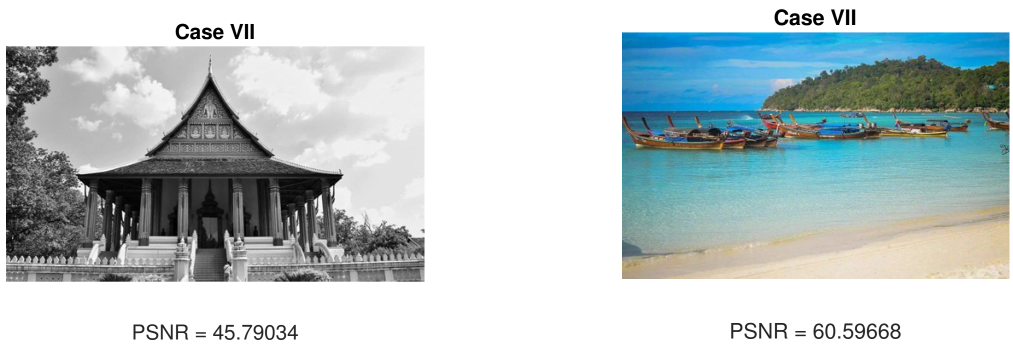


Figures 19-24: The reconstructed Grey and RGB images with their PSNR for Case I - Case III via our Algorithm 2.1 presented in 10000^{th} iterations, respectively.

It can be seen from Figures 25-30 that the quality of restored image by using our Algorithm 2.1 in solving the common solutions of deblurring problem (VIP) with ($N = 2$) has improved (compared with the previous result on Figures 19-24).

Figures 25-30: The reconstructed Grey and RGB images with their PSNR for Case IV - Case VI via our Algorithm 2.1 presented in 10000^{th} iterations respectively.

Finally, the common solution of deblurring problem (VIP) with ($N = 3$) by using the proposed algorithm is also tested (Inputting B_1, B_2 and B_3 on the proposed algorithm).



Figures 31-32: The reconstructed Grey and RGB images from the blurring operators B_1, B_2, and B_3 (Case VII) via our Algorithm 2.1 presented in 10000^{th} iterations, respectively.

Figures 31-32 show the reconstructed Grey and RGB images with thousand iteration. It has been found that the quality (PSNR) of the recovered Grey and RGB images obtained by this algorithm is highest compared to the previous two algorithm. The figures of deblurring when the 33 PSNR is the stopping criterion are shown in Figures 33-46 that be composed of the restored image and its number of iterations.

Figures 33-39: The reconstructed Grey images of all cases via our Algorithm 2.1 with PSNR = 29.

Figures 40-46: The reconstructed RGB images of all cases via our Algorithm 2.1 with PSNR = 38.

4. Conclusions

In this paper, solving common variational inequality problem was studied by combining the hybrid inertial technique with a parallel subgradient extragradient-line method. Under some suitable conditions imposed on parameters, we proved the strong convergence of the algorithm. Examples demonstrate the effectiveness of the proposed algorithm by the comparison with PVSEGM (see Table 1 and Figures 1-4). We applied our proposed algorithm to recover images compared to PVSEGM. When the PSNR of 10000^{th} and the number of iterations 33 PSNR are given, our algorithm is more efficient than PVSEGM (see Table 2 and 3). Moreover, our algorithm can solve image recovery under unknown situation of blur matrix type.

Acknowledgements
P. Peeyada and W. Cholamjiak would like to thank the Thailand Science Research and Innovation Fund and the University of Phayao (Grant No. FF65-UoE002). D. Yambangwai would like to thank the Thailand Science Research and Innovation Fund and the University of Phayao (Grant No. FF65-RIM072).

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] P.K. Anh, D.V. Hieu, Parallel hybrid methods for variational inequalities, equilibrium problems and common fixed point problems, Vietnam J. Math. 44 (2016), 351–374.
[3] A.S. Antipin, On a method for convex programs using a symmetrical modification of the Lagrange function, Ekonomika i Matematicheskie Metody 12 (1976), 1164-1173.
[4] N.T. An, P.D. Dong, X. Qin, Robust feature selection via nonconvex sparsity-based methods, J. Nonlinear Var. Anal. 5 (2021), 59-77.
[5] H.H. Bauschke, J.M. Borwein, On projection algorithms for solving convex feasibility problems, SIAM Rev. 38 (1996), 367-426.
[6] C. Baiocchi, A. Capelo, Variational and Quasivariational Inequalities: Applications to Free Boundary problems, Wiley, New York, 1984.
[7] Y. Censor, S. Petra, C. Schnorr, Superiorization vs. accelerated convex optimization: The superiorized / regularized least-squares case, J. Appl. Numer. Optim. 2 (2020), 15-62.
[8] Y. Censor, A. Gibali, S. Reich, Strong convergence of subgradient extragradient methods for the variational inequality problem in Hilbert space, Optim. Meth. Softw. 26 (2011), 827-845.
[9] C.E. Chidume, J.N. Ezeora, Krasnoselkii-type algorithm for family of multi-valued strictly pseudo-contractive mappings, Fixed Point Theory Appl. 2014 (2014), 111.
[10] Y. Censor, A. Gibali, S. Reich, The subgradient extragradient method for solving variational inequalities in Hilbert space, J. Optim. Theory Appl. 148 (2011), 318-335.
[11] Y. Censor, A. Gibali, S. Reich, Common solutions to variational inequalities, Set-Valued Var. Anal. 20 (2012), 229-247.
[12] Y. Censor, A. Gibali, S. Reich, Extensions of Korpelevich's extragradient method for the variational inequality problem in Euclidean space, Optimization 61 (2011), 1119-1132.
[13] Q.L. Dong, Y.Y. Lu, J. Yang, The extragradient algorithm with inertial effects for solving the variational inequality, Optimization 65 (2016), 2217-2226.
[14] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Dordrecht: Kluwer Academic Publishers, 2000.
[15] C. Fang, S. Chen, Some extragradient algorithms for variational inequalities, Advances in variational and hemivariational inequalities, pp. 145-171, Springer, Cham, 2015.
[16] J. Fan, L. Liu, X. Qin, A subgradient extragradient algorithm with inertial effects for solving strongly pseudomonotone variational inequalities, Optimization, 69 (2020), 2199-2215.
[17] A. Gibali, S. Reich, R. Zalas, Outer approximation methods for solving variational inequalities in Hilbert space, Optimization 66 (2017), 417-431.
[18] P.C. Hansen, Rank-deficient and discrete ill-posed problems, PA: SIAM Philadelphia, 1997.
[19] P.C. Hansen, Discrete Inverse Problems: Insight and Algorithms, PA: SIAM Philadelphia, 2010.
[20] D. Kinderlehrer, G. Stampacchia, An Introduction to Variational Inequalities and Their Applications, Academic, New York (1980).
[21] G.M. Korpelevich, The extragradient method for finding saddle points and other problems, Ekonomikai Matematicheskie Metody 12 (1976), 747-756.
[22] 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.
[23] L. Liu, S.Y. Cho, J.C. Yao, Convergence analysis of an inertial Tseng's extragradient algorithm for solving pseudomonotone variational inequalities and applications, J. Nonlinear Var. Anal. 5 (2021), 627-644.
[24] M.A. Mansour, J. Lahrache, N. Ziane, Weak approximate solutions to quasi-variational inequalities: Application to social Nash equilibria, Appl. Set-Valued Anal. Optim. 3 (2021), 149-164.
[25] C. Martinez-Yanes, H.K. Xu, Strong convergence of the CQ method for fixed point iteration processes, Nonlinear Anal. 64 (2006), 2400-2411.
[26] D.R. Sahu, J.C. Yao, M. Verma, K.K. Shukla, Convergence rate analysis of proximal gradient methods with applications to composite minimization problems, Optimization, 70 (2021), 75-100.

[27] M.V. Solodov, B.F. Svaiter, A new projection method for variational inequality problems, SIAM J. Control Optim. 37 (1999), 765-776.

[28] S. Suantai, P. Peeyada, D. Yambangwai, W. Cholamjiak, 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.

[29] W. Takahashi, Nonlinear Functional Analysis, Yokohama Publishers, Yokohama, 2000.

[30] D.V. Thong, D. Van Hieu, Modified subgradient extragradient method for variational inequality problems, Numer. Algorithms, 79 (2018), 597-610.

[31] C.R. Vogel, Computational Methods for Inverse Problems, PA: SIAM Philadelphia, 2002.

[32] J. Yang, H. Liu, Z. Liu, Modified subgradient extragradient algorithms for solving monotone variational inequalities, Optimization, 67 (2018), 2247-2258.