Solving common variational inequalities by hybrid inertial parallel subgradient extragradient-line algorithm for application to image deblurring

Pronpat Peeyada, Watcharaporn Cholamjiak; Damrongsak Yambangwai
School of Science, University of Phayao, Phayao 56000, Thailand

Abstract

In this paper, we propose hybrid inertial parallel subgradient extragradient-line algorithm for approximating a common solution of variational inequality problems with monotone and $L$-Lipschitz continuous mappings but $L$ is unknown and prove strong convergence under some mild conditions in Hilbert space. We then give numerical examples to demonstrate the performance of our algorithms better than some of the algorithms mentioned in the literature. The novelty of our algorithm is that we have shown the algorithm is resilient and has good quality when the number of subproblems is large, the algorithm can be applied to solve image deblurring when an image has common types of blur effects.

AMS Mathematics Subject Classification (2010): 46T99, 47H05, 47H10, 47J25, 49M30, 49M37.

Key words and Phrases: Common variational inequality problems; inertial methods; parallel subgradient extragradient-line algorithm; hybrid algorithms; image deblurring.

1 Introduction and preliminaries

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

$$\langle Ax^*, x - x^* \rangle \geq 0, \quad \forall x \in C, \quad (1.1)$$
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: find a point $x^* \in C$ such that
\[ x^* = P_C(x^* - \lambda A x^*), \]
where $\lambda$ is any positive real number. The VIP (1.1) is a fundamental problem in nonlinear analysis and optimization theory which is applied in many ways, such as signal processing, image recovery, transportation problems, economics, engineering, see [1, 4, 5, 17, 19, 20, 23, 26] and the references therein.

Projection type methods have been extensively used to solve VIP(1.1), see [4, 7, 10]. An important projection method which is called the Extragradient Method (EGM) was proposed by Korpelevich [21] in 1976, see also [3]. The method is generated by giving the current iterate $x_n$, compute
\[
\begin{align*}
    y_n &= P_C(x_n - \lambda Ax_n), \\
    x_{n+1} &= P_C(x_n - \lambda Ay_n),
\end{align*}
\]
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 by many authors, who improved it in various ways (see, for example, [7, 9, 10, 12, 13, 15, 18, 31, 34] and the references therein).

In 2011, Censor et al. [11] improved the EGM (1.2) for approximating a solution of the VIP(1.1) in Hilbert spaces. The method have been called the subgradient extragradient method (SEGM). Their method is of the form:
\[
\begin{align*}
    w_n &= x_n + \alpha_n(x_n - x_{n-1}), \\
    y_n &= P_C(w_n - \tau Aw_n), \\
    T_n &= \{ w \in H : \langle x_n - \lambda Ax_n - y_n, w - y_n \rangle \leq 0 \}, \\
    x_{n+1} &= P_{T_n}(x_n - \lambda Ay_n).
\end{align*}
\]
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. [11] obtained weak convergence results for solving VIP(1.1) using (1.3).

Recently, Alvarez and Attouch [2], and Censor et al. [11], used the inertial extrapolation term to speed up the rate of convergence of the SEGM for solving the VIP(1.1) in Hilbert spaces. This proposed algorithm have been called inertial subgradient extragradient method (ISEGM). The algorithm is designed by choosing $x_0, x_1 \in H$ and compute
\[
\begin{align*}
    w_n &= x_n + \alpha_n(x_n - x_{n-1}), \\
    y_n &= P_C(w_n - \tau Aw_n), \\
    T_n &= \{ x \in H : \langle w_n - \tau Aw_n - y_n, x - y_n \rangle \leq 0 \}, \\
    x_{n+1} &= P_{T_n}(w_n - \tau Ay_n),
\end{align*}
\]
where
\[ \alpha_n \in (0, 1], \quad \tau \geq 1, \quad \lambda \in (0, \frac{1}{L}). \]
where $\tau > 0$, $\alpha_n \geq 0$ are suitable parameters. Under several appropriate conditions imposed on these parameters, weak convergence result was established, here, the assumption of monotonicity and Lipschitz continuous which Lipschitz constant is known were required.

Our interest in this paper is to study of finding 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

$$\langle A_i x^*, x - x^* \rangle \geq 0, \quad \forall x \in C, \quad i = 1, 2, \ldots, N.$$  \hspace{1cm} (1.5)

If $N = 1$, CVIP (1.5) becomes VIP (1.1).

Very recently, Suantai et al. [28] motivated the viscosity-type subgradient extragradient-line method which introduced by Shehu and Iyiola [21] to solve the CVIP (1.5). This algorithm was called the parallel viscosity-type subgradient extragradient-line method (PVSEGM). The strong convergence theorem was proved when each of the operator $A_i$ is Lipschitz continuous monotone mapping that the Lipschitz constant is unknown. This algorithm start with $x_1 \in H$ and compute

$$\begin{align*}
y_n^i &= P_C(x_n - \lambda_n^i A_ix_n), \quad \lambda_n^i = \rho_n^i, \\
l_n^i &= \text{the smallest nonnegative integer } l^i \text{ such that } \lambda_n^i \|A_i x_n - A_i y_n^i\| \leq \mu \|r_{\rho_n^i}(x_n)\|, \\
z_n^i &= P_{T_n^i}(x_n - \lambda_n^i A_y_n^i), \\
x_{n+1} &= \alpha_0 f(x_n) + \sum_{i=1}^{N} \alpha_n z_n^i, \quad n \geq 1,
\end{align*}$$  \hspace{1cm} (1.6)

where $T_n^i = \{ z \in H : \langle x_n - \lambda_n^i A_i x_n - y_n^i, z - y_n^i \rangle \leq 0 \}$ with $\rho, \mu \in (0, 1)$ and $\{\alpha_n\}_{n=1}^{\infty} \subseteq (0, 1)$. The sequence $\{x_n\}_{n=1}^{\infty}$ generated by (1.6) 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

$$\langle (I - f)x^*, x - x^* \rangle \geq 0, \quad \forall x \in VI(C, A),$$  \hspace{1cm} (1.7)

where $f : C \to C$ be a strict contraction mapping with constant $k \in (0, 1]$ under the following conditions

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

The advantage of the PVSEGM was presented to solve the problem of multiblur effects in an image restoration. The resulting image quality is 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 introduce combining hybrid inertial techniques with a parallel subgradient extragradient-line method for solving CVIP (1.5). 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 applying our algorithm.
2 Main result

In this section, we propose the hybrid inertial parallel subgradient extragradient-line method for solving CVIP (1.5). Let $H$ be a real Hilbert space and $C$ be a nonempty closed 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 $i = 1, 2, ..., N$ such that $\bigcap_{i=1}^{N} VI(C, A_i) \neq \emptyset$. Suppose $\{x_n\}_{n=1}^{\infty}$ is generated in the following Algorithm 2.1:

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

**Step 1** Compute

$$t_n = x_n + \theta_n(x_n - x_{n-1}), \quad \forall n \geq 1.$$  

**Step 2** Compute $y_n^i$ for all $i = 1, 2, ..., N$ by

$$y_n^i = P_C(t_n - \lambda_n^i A_i t_n), \quad \forall n \geq 1,$$

where $\lambda_n^i = \rho_{l_n^i}$ and $l_n^i$ is the smallest nonnegative integer such that

$$\lambda_n^i \|A_i t_n - A_i y_n^i\| \leq \mu \|t_n - y_n^i\|. \quad (2.1)$$

**Step 3** Compute

$$z_n^i = P_{T_n}(t_n - \lambda_n^i A_n y_n^i),$$

where $T_n := \{z \in H : \langle t_n - \lambda_n^i A_i t_n - y_n^i, z - y_n^i \rangle \leq 0\}$.

**Step 4** Compute

$$\bar{u}_n = \alpha^0_n(t_n) + \sum_{i=1}^{N} \alpha_n^i z_n^i, \quad n \geq 1,$$

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

**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 $l_i^n$ satisfying (2.1).

Proof For each $i = 1, 2, \ldots, N$ and $n \in \mathbb{N}$, we let $y_i^l = P_C(t_n - \rho^l A_i t_n)$ for all $l \in \mathbb{N}$. We divide the proof into two cases as follows:

- **case I**: if $\| t_n - y_i^{n_0} \| = 0$ for some $n_0 \geq 1$, then we take $l_i^n = 0$ which satisfies (2.1).
- **case II**: if $\| t_n - y_i^{n_1} \| \neq 0$ for some $n_1 \geq 1$, then we assume the contrary that $\rho^{n_1} \| A_i t_n - A_i y_i^{n_1} \| > \mu \| t_n - y_i^{n_1} \|.$

Then, by Lemma 6.3 of [16] and the fact that $\rho \in (0, 1)$, we obtain

$$
\| A_i t_n - A_i y_i^{n_1} \| > \frac{\mu}{\rho^{n_1}} \| t_n - y_i^{n_1} \| \\
\geq \frac{\mu}{\rho^{n_1}} \min\{1, \rho^{n_1}\} \| t_n - y_i \| \\
= \mu \| t_n - y_i \|. 
$$

By using the continuity of $P_C$, we have that

$$
y_i^{n_1} = P_C(y_n - \rho^{n_1} A_i t_n) \rightarrow P_C(t_n), \; n_1 \rightarrow \infty \; \text{for all} \; i = 1, 2, \ldots, N.
$$

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 $\| t_n - y_i^{n_1} \| \neq 0$ and $\rho^{n_1} \leq 1$, it follows from Lemma 6.3 of [16] again, we have

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

Taking $n_1 \rightarrow \infty$ in (2.1), we have that

$$
0 = \| A_i t_n - A_i t_n \| \geq \mu \| t_n - y_i \| > 0.
$$

This is a contradiction and hence (2.1) is well defined.

(ii) If $t_n \notin C$, then

$$
\rho^{n_1} \| A_i t_n - A_i y_i^{n_1} \| \rightarrow 0, \; \text{as} \; n_1 \rightarrow \infty
$$

while

$$
\lim_{n_1 \rightarrow \infty} \mu \| t_n - y_i^{n_1} \| = \mu \lim_{n_1 \rightarrow \infty} \| t_n - P_C(t_n - \rho^{n_1} A_i t_n) \| \\
= \mu \| t_n - P_C(t_n) \| > 0.
$$

This is a contradiction. Therefore, linesearch in Algorithm 3.1 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) $\liminf_{n \rightarrow \infty} \alpha_n^i > 0$ for all $i = 1, 2, \ldots, N$.

Then the sequence $\{x_n\}$ generated by Algorithm 2.1 converges strongly to $z \in \Xi.$
Proof. We split the proof into five steps.

**Step 1.** Show that \( \{x_n\} \) is well defined. From \( C_1 = C, C_1 \) is closed and convex. Assume that \( C_n \) is closed and convex. From the definition of \( C_{n+1} \) and Lemma 1.3 in [22], we get \( C_{n+1} \) is closed and convex. Let \( x^* \in \Upsilon \) and \( s_n^i = t_n - \lambda_n^i A_i y_n^i, \forall n \geq 1, i = 1, 2, \ldots, N, \) we have

\[
\| z_n^i - x^* \|^2 = \| P_{T_n^i}(s_n^i) - x^* \|^2 \\
= \| P_{T_n^i}(s_n^i) - s_n^i \|^2 + 2 \langle P_{T_n^i}(s_n^i) - s_n^i, s_n^i - x^* \rangle + \| s_n^i - x^* \|^2. \tag{2.5}
\]

Since \( x^* \in \Upsilon \subseteq C \subseteq T_n^i \) and by the characterization of the metric projection \( P_{T_n^i} \), we get

\[
2 \| s_n^i - P_{T_n^i}(s_n^i) \|^2 + 2 \langle P_{T_n^i}(s_n^i) - s_n^i, s_n^i - x^* \rangle \\
= 2 \langle s_n^i - P_{T_n^i}(s_n^i), x^* - P_{T_n^i}(s_n^i) \rangle \leq 0. \tag{2.6}
\]

This implies that

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

By the definition of Algorithm 3.1 the inequalities (2.5) and (2.6), we have

\[
\| z_n^i - x^* \|^2 \leq \| s_n^i - x^* \|^2 - \| s_n^i - z_n^i \|^2 \\
= \| t_n - x^* \|^2 - \| t_n - z_n^i \|^2 - \| t_n - z_n^i \|^2 - \| t_n - z_n^i \|^2 + 2 \lambda_n^i \langle - t_n + x^*, A_i y_n^i \rangle \\
+ 2 \lambda_n^i \langle t_n - z_n^i, A_i y_n^i \rangle \\
= \| t_n - x^* \|^2 - \| t_n - z_n^i \|^2 + 2 \lambda_n^i \langle x^* - z_n^i, A_i y_n^i \rangle. \tag{2.8}
\]

By the monotonicity of the operator \( A_i \), we have

\[
0 \leq \langle A_i y_n^i - A_i x^*, y_n^i - x^* \rangle \\
= \langle A_i y_n^i, y_n^i - x^* \rangle - \langle A_i x^*, y_n^i - x^* \rangle \\
\leq \langle A_i y_n^i, y_n^i - x^* \rangle \\
= \langle A_i y_n^i, y_n^i - z_n^i \rangle + \langle A_i y_n^i, z_n^i - x^* \rangle.
\]

Thus

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

Using (2.9) in (2.8), we obtain

\[
\| z_n^i - x^* \|^2 \leq \| t_n - x^* \|^2 - \| t_n - z_n^i \|^2 + 2 \lambda_n^i \langle A_i y_n^i, y_n^i - z_n^i \rangle \\
+ 2 \lambda_n^i \langle A_i y_n^i, y_n^i - z_n^i \rangle \\
= \| t_n - x^* \|^2 - \| t_n - y_n^i \|^2 - \| y_n^i - z_n^i \|^2 - 2 \langle t_n - y_n^i, y_n^i - z_n^i \rangle \\
+ 2 \lambda_n^i \langle A_i y_n^i, y_n^i - z_n^i \rangle \\
= \| t_n - x^* \|^2 - \| t_n - y_n^i \|^2 - \| y_n^i - z_n^i \|^2 \\
+ 2 \langle t_n - \lambda_n^i A_i y_n^i - y_n^i, y_n^i - y_n^i \rangle. \tag{2.10}
\]
Consider the following inequalities
\[
\langle t_n - \lambda_n^i A_i y_n^i - y_n^i, z_n^i - y_n^i \rangle = \langle t_n - \lambda_n^i A_i t_n - y_n^i, z_n^i - y_n^i \rangle + \langle \lambda_n^i A_i t_n - \lambda_n^i A_i y_n^i, z_n^i - y_n^i \rangle \\
\leq \langle \lambda_n^i A_i t_n - \lambda_n^i A_i y_n^i, z_n^i - y_n^i \rangle.
\]

Using the last inequality in (2.10), we have that
\[
\| z_n^i - x^* \|^2 \leq \| t_n - x^* \|^2 - \| t_n - y_n^i \|^2 - \| y_n^i - z_n^i \|^2 + 2\langle \lambda_n^i A_i t_n - \lambda_n^i A_i y_n^i, z_n^i - y_n^i \rangle \\
\leq \| t_n - x^* \|^2 - \| t_n - y_n^i \|^2 - \| y_n^i - z_n^i \|^2 + 2\lambda_n^i \| A_i t_n - A_i y_n^i \| \| z_n^i - y_n^i \| \\
\leq \| t_n - x^* \|^2 - \| t_n - y_n^i \|^2 - \| y_n^i - z_n^i \|^2 + 2\mu \| t_n - y_n^i \| \| z_n^i - y_n^i \| \\
\leq \| t_n - x^* \|^2 - \| t_n - y_n^i \|^2 - \| y_n^i - z_n^i \|^2 + \mu(\| t_n - y_n^i \|^2 + \| z_n^i - y_n^i \|^2) \\
= \| t_n - x^* \|^2 - (1 - \mu)(\| t_n - y_n^i \|^2 + \| y_n^i - z_n^i \|^2).
\]

This implies that
\[
\| \bar{u}_n - x^* \|^2 \leq \alpha_n^0 \| t_n - x^* \|^2 + \sum_{i=1}^{N} \| z_n^i - x^* \|^2 \\
\leq \| t_n - x^* \|^2.
\]
This shows that \( \| \bar{u}_n - x^* \| = \| t_n - x^* \| \), this mean that \( x^* \in C_n, \forall n \geq 1 \). This implies that \( \{x_n\} \) is well-defined.

**Step 2.** Show that \( \lim_{n \to \infty} \| x_n - x_1 \| \) exists. Since \( \Upsilon \) 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 get
\[
\| x_n - x_1 \| \leq \| x_{n+1} - x_1 \|; \forall n \geq 1.
\]
On the other hand, as \( \Upsilon \subset C_n \), we obtain
\[
\| x_n - x_1 \| \leq \| v - x_1 \|; \forall n \geq 1.
\]
It follows from (2.12) and (2.13) that the sequence \( \{x_n\} \) is bounded and nondecreasing. Therefore \( \lim_{n \to \infty} \| x_n - x_1 \| \) exists.

**Step 3.** Show that \( x_n \to \omega \in C \) as \( n \to \infty \). For \( k > j \), by the definition of \( C_j \), since \( x_k = P_{C_k} x_1 \in C_k \subset C_j \), so by the property of the metric projection \( P_{C_j} \) [6], we have
\[
\| x_k - x_j \|^2 \leq \| x_k - x_1 \|^2 - \| x_j - x_1 \|^2.
\]
Since \( \lim_{j \to \infty} \| x_j - x_1 \| \) exists, we have \( \| x_k - x_j \| \to 0 \), as \( k, j \to \infty \). This 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 4.** Show that \( \lim_{n \to \infty} \| x_n - y_n^i \| = \lim_{n \to \infty} \| y_n^i - z_n^i \| = 0 \) for all \( i = 1, 2, ..., N \). Let \( x^* \in \Upsilon \). Then,
we have from (2.2), (2.11) and Lemma 2.1 in [14] that
\[
\| \bar{u}_n - x^* \|^2 = \| \alpha_n^0(t_n) + \sum_{i=1}^{N} \alpha_n^i z_i^n - x^* \|^2 \\
\leq \alpha_n \| t_n - x^* \|^2 + \sum_{i=1}^{N} \alpha_n^i \| z_i^n - x^* \|^2 \\
= \| t_n - x^* \|^2 - (1 - \mu) \sum_{i=1}^{N} \alpha_n^i (\| t_n - y_i^n \|^2 + \| y_i^n - z_i^n \|^2) \\
= \| x_n - x^* \|^2 + \theta_n^2 \| x_n - x_{n-1} \|^2 + 2 \langle x_n - x^*, \theta_n(x_n - x_{n-1}) \rangle \\
-(1 - \mu) \sum_{i=1}^{N} \alpha_n^i (\| x_n - y_i^n \|^2 + \theta_n^2 \| x_n - x_{n-1} \|^2 \\
+ 2 \langle x_n - y_i^n, \theta_n(x_n - x_{n-1}) \rangle + \| y_i^n - z_i^n \|^2). \tag{2.14}
\]
Since \( x_{n+1} \in C_{n+1} \subset C_n \), we have
\[
\| \bar{u}_n - x_{n+1} \| \leq \| t_n - x_{n+1} \| \\
\leq \| t_n - x_n \| + \| x_n - x_{n+1} \| \\
= \theta_n \| x_n - x_{n-1} \| + \| x_n - x_{n+1} \| \to 0, \text{ as } n \to \infty.
\]
This implies that
\[
\| \bar{u}_n - x_n \| \leq \| \bar{u}_n - x_{n+1} \| + \| x_{n+1} - x_n \| \to 0, \text{ as } n \to \infty. \tag{2.15}
\]
It follows form (2.14) that
\[
(1 - \mu) \sum_{i=1}^{N} \alpha_n^i (\| x_n - y_i^n \|^2 + \| y_i^n - z_i^n \|^2) \leq \| x_n - x^* \|^2 - \| \bar{u}_n - x^* \|^2 \\
+ \theta_n^2 \| x_n - x_{n-1} \|^2 + 2 \langle x_n - x^*, \theta_n(x_n - x_{n-1}) \rangle \\
-(1 - \mu) \sum_{i=1}^{N} \alpha_n^i (\theta_n^2 \| x_n - x_{n-1} \|^2 \\
+ 2 \langle x_n - y_i^n, \theta_n(x_n - x_{n-1}) \rangle). \tag{2.15}
\]
By our assumptions (i), (ii) and (2.15), we obtain
\[
\lim_{n \to \infty} \| y_i^n - z_i^n \| = \lim_{n \to \infty} \| x_n - y_i^n \| = 0, \forall i = 1, 2, ..., N. \tag{2.16}
\]
**Step 5.** We show that \( \omega \in \mathcal{Y} \). Now, \( x_n - y_i^n \to 0 \) implies that \( y_i^n \to \omega \) and since \( y_i^n \in C \), we then obtain \( \omega \in C \). For all \( x \in C \) and using the property of the projection \( P_C \), we have (Since \( A_i \) is monotone)
\[
0 \leq \langle y_i^n - t_n + \lambda_n^i A_i t_n, x - y_i^n \rangle \\
= \langle y_i^n - t_n, x - y_i^n \rangle + \langle \lambda_n^i A_i t_n, x - x_n \rangle + \langle \lambda_n^i A_i t_n, x_n - y_i^n \rangle \\
\leq \langle y_i^n - x_n, x - x_n \rangle + \lambda_n^i \langle A_i x, x - x_n \rangle + \lambda_n^i \langle A_i x_n, x_n^i - y_i^n \rangle \\
+ \langle \theta_n(x_n - x_{n-1}), x - y_i^n \rangle + \lambda_n^i \langle A_i \theta_n(x_n - x_{n-1}), x - x_n \rangle \\
+ \lambda_n^i \langle A_i \theta_n(x_n - x_{n-1}), x - y_i^n \rangle. \tag{2.17}
\]
By Remark 3.2 in [29], we know that \( \inf_{n \geq 1} \lambda_n > 0 \). So by taking \( n \to \infty \) in (2.17), we obtain
\[
\langle A; x, x - \omega \rangle \geq 0, \quad \forall x \in C.
\]
This implies that \( \omega \in VI(C, A_i) \) for all \( i = 1, 2, \ldots, N \). This completes the proof.

Base on the choice of the inertial parameter \( \theta_n \) the relation between Algorithm 2.1 where \( A_i = A \) for all \( i = 1, 2, \ldots, N \), then Algorithm 2.1 reduces to the following hybrid inertial subgradient extragradient algorithm:

**Algorithm 2.4** Take \( \rho \in (0, 1) \), \( \mu \in (0, 1) \). 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
\[
t_n = x_n + \theta_n(x_n - x_{n-1}), \quad \forall n \geq 1.
\]

**Step 2** Compute \( y_n \) by
\[
y_n = P_C(t_n - \lambda_nAt_n), \quad \forall n \geq 1,
\]
where \( \lambda_n = \rho^n \) and \( l_n \) is the smallest nonnegative integer such that
\[
\lambda_n \| At_n - Ay_n \| \leq \mu \| t_n - y_n \|.
\] (2.18)

**Step 3** Compute
\[
z_n = P_{T_n}(t_n - \lambda_nAy_n),
\]
where \( T_n := \{z \in H : \langle t_n - \lambda_nAt_n - y_n, z - y_n \rangle \leq 0 \} \).

**Step 4** Compute
\[
\bar{u}_n = \alpha_n t_n + (1 - \alpha_n)z_n,
\]
where \( \alpha_n \in (0, 1) \).

**Step 5** Compute
\[
x_{n+1} = P_{C_{n+1}}x_1
\]
where \( C_{n+1} := \{z \in C_n : \| \bar{u}_n - z \| \leq \| t_n - z \| \} \).
Set \( n + 1 \to n \) and go to Step 1.

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

**Example 2.5** Let \( A_1, A_2 : \mathbb{R}^3 \to \mathbb{R}^3 \) be defined by \( A_1x = 4x \) and \( A_2x = \begin{pmatrix} 10 & -5 & 5 \\ -5 & 10 & -5 \\ 5 & -5 & 10 \end{pmatrix} \) 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 \( \theta = 0.15 \), \( \alpha_n^0 = \frac{n^2+1}{3n^2+n} \) 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 \( i = 1, 2, ..., N \) in the first case and the second \( A_i = A_2 \) for all \( i = 1, 2, ..., N \). Choose \( \alpha_n^0 = \frac{n^2+1}{100n^2+n} \), \( \alpha_n^1 = \frac{5n^2+2}{100n+1} \) and \( \alpha_n^2 = 1 - \left( \alpha_n^0 + \alpha_n^1 \right) \) for the third case that we put \( A_1, A_2 \) in our Algorithm 2.1.

(2) Choose \( \alpha_n^0 = \frac{1}{(n+1)(n+3)} \), \( \alpha_n^1 = \frac{1}{2n} \) and \( \alpha_n^2 = 1 - \left( \alpha_n^0 + \alpha_n^1 \right) \) for PVSEMG in Theorem 1 [28] to compare the convergence of our Algorithm 2.1.

Table 1: Comparison of the methods in Theorem 2.3 and Theorem 1 [28] of Example 2.5

<table>
<thead>
<tr>
<th>( A_1 )</th>
<th>( A_2 )</th>
<th>( \rho = 0.2, \mu = 0.3 )</th>
<th>302</th>
<th>0.0000226</th>
<th>263</th>
<th>0.0000392</th>
<th>229</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho = 0.4, \mu = 0.5 )</td>
<td>212</td>
<td>0.0000335</td>
<td>300</td>
<td>0.000029</td>
<td>202</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \rho = 0.4, \mu = 0.3 )</td>
<td>212</td>
<td>0.0000169</td>
<td>306</td>
<td>0.000026</td>
<td>215</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \rho = 0.3, \mu = 0.4 )</td>
<td>175</td>
<td>0.0000163</td>
<td>348</td>
<td>0.000024</td>
<td>187</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PVSEMG
| \( \rho = 0.2, \mu = 0.1 \) | 591 | 0.0000086 | 505 | 0.0000179 | 506 |

Figure 1-4: Error plots for Table 1 in Example 2.5.

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

(i) it is clearly seen that the common solution of CVIP (1.5) with \( N = 2 \) get 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 \( \rho \) and \( \mu \) are different, we get that the case \( N = 2 \) converges faster than \( N = 1 \);

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

$$b = Bx + v,$$

(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, $v$ 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 $x = B^{-1}(b - v)$ 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

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

(3.2)

where $\|\cdot\|$ is $\ell_2$-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 method [36, 37, 38, 39].

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

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|B_1 x - b_1\|_2^2, \min_{x \in \mathbb{R}^n} \frac{1}{2} \|B_2 x - b_2\|_2^2, ..., \min_{x \in \mathbb{R}^n} \frac{1}{2} \|B_N x - b_N\|_2^2,$$

(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 $i = 1, 2, ..., N$. For solving 3.3, we can apply our main Algorithm 2.1 by setting $A_i x = B_i^T (B_i x - b_i)$ for all $x \in \mathbb{R}^n$ in Algorithm 2.1 since $B_i^T (B_i x - b_i)$ is Lipschitz continuous for each $i = 1, 2, ..., N$. This algorithm is generated as follows:

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

(3.4)

where $T_n^i = \{ z \in H \mid \langle t_n - \lambda_n^i B_i t_n - y_n^i, z - y_n^i \rangle \leq 0 \}$, $C_{n+1} = \{ z \in C_n \mid \| \bar{u}_n - z \| \leq \| t_n - z \| \}$, $\rho, \mu, \alpha_n^i \in (0, 1)$ 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 $9 \times 9$ 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 ($\text{len} = 21$) and motion orientation $11^\circ$ ($\theta = 11$) (blur matrix $B_3$).

The original Grey and RGB images are show in figure 5-6.
Figure 5-6: The original Grey and RGB image of sizes $276 \times 490$ 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.

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

We apply the PVSEMG and our Algorithm 2.1 in getting the solution of deblurring problem with the three blurring matrices $B_1, B_2, B_3$. The results of the PVSEMG and our Algorithm 2.1 are considered in following seven cases:
Case I: Inputting $B_1$ on the PVSEMG and Algorithm 2.1,
Case II: Inputting $B_2$ on the PVSEMG and Algorithm 2.1,
Case III: Inputting $B_3$ on the PVSEMG and Algorithm 2.1,
Case IV: Inputting $B_1$ and $B_2$ on the PVSEMG and Algorithm 2.1,
Case V: Inputting $B_1$ and $B_3$ on the PVSEMG and Algorithm 2.1,
Case VI: Inputting $B_2$ and $B_3$ on the PVSEMG and Algorithm 2.1,
Case VII: Inputting $B_1, B_2$ and $B_3$ on the PVSEMG and Algorithm 2.1.
Table 2: Comparison of the number of iterations in Grey images.

<table>
<thead>
<tr>
<th>Inputting</th>
<th>PSNR of 10,000th</th>
<th>Number of Iterations</th>
<th>PSNR of 10,000th</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVSEMG</td>
<td>Our Algorithm</td>
<td>PVSEMG</td>
<td>Our Algorithm</td>
<td></td>
</tr>
<tr>
<td>$B_1$</td>
<td>24.70720</td>
<td>29.57263</td>
<td>4921th</td>
<td>50th</td>
</tr>
<tr>
<td>$B_2$</td>
<td>26.47867</td>
<td>34.15647</td>
<td>2775th</td>
<td>58th</td>
</tr>
<tr>
<td>$B_3$</td>
<td>29.50780</td>
<td>35.32024</td>
<td>801th</td>
<td>36th</td>
</tr>
<tr>
<td>$B_1, B_2$</td>
<td>28.59585</td>
<td>36.01784</td>
<td>975th</td>
<td>60th</td>
</tr>
<tr>
<td>$B_1, B_3$</td>
<td>32.37244</td>
<td>42.50473</td>
<td>446th</td>
<td>62th</td>
</tr>
<tr>
<td>$B_2, B_3$</td>
<td>33.47745</td>
<td>46.33505</td>
<td>538th</td>
<td>73th</td>
</tr>
<tr>
<td>$B_1, B_2, B_3$</td>
<td>34.41830</td>
<td>45.79034</td>
<td>411th</td>
<td>52th</td>
</tr>
</tbody>
</table>

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

![Figure 13-15](image_url)

Figure 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.

<table>
<thead>
<tr>
<th>Inputting</th>
<th>PSNR of 10,000th</th>
<th>Number of Iterations</th>
<th>PSNR of 10,000th</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVSEMG</td>
<td>Our Algorithm</td>
<td>PVSEMG</td>
<td>Our Algorithm</td>
<td></td>
</tr>
<tr>
<td>$B_1$</td>
<td>33.47997</td>
<td>38.31203</td>
<td>6816th</td>
<td>385th</td>
</tr>
<tr>
<td>$B_2$</td>
<td>34.13544</td>
<td>41.83745</td>
<td>5800th</td>
<td>364th</td>
</tr>
<tr>
<td>$B_3$</td>
<td>37.89834</td>
<td>45.57931</td>
<td>1014th</td>
<td>86th</td>
</tr>
<tr>
<td>$B_1, B_2$</td>
<td>37.46071</td>
<td>47.54648</td>
<td>1253th</td>
<td>190th</td>
</tr>
<tr>
<td>$B_1, B_3$</td>
<td>41.57133</td>
<td>54.15965</td>
<td>509th</td>
<td>86th</td>
</tr>
<tr>
<td>$B_2, B_3$</td>
<td>41.77308</td>
<td>53.88841</td>
<td>634th</td>
<td>87th</td>
</tr>
<tr>
<td>$B_1, B_2, B_3$</td>
<td>43.52842</td>
<td>60.59668</td>
<td>474th</td>
<td>122th</td>
</tr>
</tbody>
</table>
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\textsuperscript{th} iterations are shown in figures 16-18.

![Cauchy error and PSNR plots](image)

**Figure 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 10,000\textsuperscript{th} iterations is the stopping criterion are shown in figures 19-32 that be composed of the restored image and its PSNR.

![Deblurred images](image)

**Figure 19-24:** The reconstructed Grey and RGB images with their PSNR for Case I - Case III being used our Algorithm 2.1 presented in 10000\textsuperscript{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.

![Image](image1.png)

**Figure 25-30:** The reconstructed Grey and RGB images with their PSNR for Case IV - Case VI used 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).

![Image](image2.png)

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

Figure 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.
Figure 33-39: The reconstructed Grey images of all cases being used our Algorithm 2.1 with PSNR = 29.

Figure 33-39: The reconstructed Grey images of all cases being used our Algorithm 2.1 with PSNR = 29.
Figure 40-46: The reconstructed RGB images of all cases being used our Algorithm 2.1 with PSNR = 38.

4 Conclusions

In this paper, solving common variational inequality problem are studied by combining the hybrid inertial technique with a parallel subgradient extragradient-line method. Under some suitable conditions imposed on parameters, we have proved the strong convergence of the algorithm. Examples that demonstrate the effectiveness of the proposed algorithm by comparison with PVSEMG see in Table 1 and Figure 1-4. We apply our proposed algorithm to recover images compared to PVSEMG, when PSNR of 10,000th and number of iterations 33 PSNR are given, our algorithm is more efficient than PVSEMG see in Table 2 and 3. Moreover, our algorithm can solve image recovery under unknown situation of blur matrix type, to demonstrate the computational performance see in Figures 25-32 and Figures 33-46.

Acknowledgement

The authors would like to thank University of Phayao, Phayao, Thailand.

References


