

PROJECTED REFLECTED GRADIENT METHODS FOR MONOTONE VARIATIONAL INEQUALITIES

YU. MALITSKY[†]

Abstract. This paper is concerned with some new projection methods for solving variational inequality problems with monotone and Lipschitz-continuous mapping in Hilbert space. First, we propose the projected reflected gradient algorithm with a constant stepsize. It is similar to the projected gradient method, namely, the method requires only one projection onto the feasible set and only one value of the mapping per iteration. This distinguishes our method from most other projection-type methods for variational inequalities with monotone mapping. Also we prove that it has R-linear rate of convergence under the strong monotonicity assumption. The usual drawback of algorithms with constant stepsize is the requirement to know the Lipschitz constant of the mapping. To avoid this, we modify our first algorithm so that the algorithm needs at most two projections per iteration. In fact, our computational experience shows that such cases with two projections are very rare. This scheme, at least theoretically, seems to be very effective. All methods are shown to be globally convergent to a solution of the variational inequality. Preliminary results from numerical experiments are quite promising.

Key words. variational inequality, projection method, monotone mapping, extragradient method

AMS subject classifications. 47J20, 90C25, 90C30, 90C52

1. Introduction. We consider the classical variational inequality problem (VIP) which is to find a point $x^* \in C$ such that

$$\langle F(x^*), x - x^* \rangle \geq 0 \quad \forall x \in C, \quad (1.1)$$

where C is a closed convex set in Hilbert space H , $\langle \cdot, \cdot \rangle$ denotes the inner product in H , and $F: H \rightarrow H$ is some mapping. We assume that the following conditions hold

- (C1) The solution set of (1.1), denoted by S , is nonempty.
- (C2) The mapping F is monotone, i.e.,

$$\langle F(x) - F(y), x - y \rangle \geq 0 \quad \forall x, y \in H.$$

- (C3) The mapping F is Lipschitz-continuous with constant $L > 0$, i.e., there exists $L > 0$ such that

$$\|F(x) - F(y)\| \leq L \|x - y\| \quad \forall x, y \in H.$$

The variational inequality problem is one of the central problems in nonlinear analysis (see [2,5,21]). Also monotone operators have turned out to be an important tool in the study of various problems arising in the domain of optimization, nonlinear analysis, differential equations and other related fields (see [3,22]). Therefore, numerical methods for VIP with monotone operator have been extensively studied in the literature, see [5,10] and references therein. In this section we briefly consider the development of projection methods for monotone variational inequality that provide weak convergence to a solution of (1.1).

The simplest iterative procedure is the well-known projected gradient method

$$x_{n+1} = P_C(x_n - \lambda F(x_n)),$$

where P_C denotes the metric projection onto the set C and λ is some positive number. In order to converge, however, this method requires the restrictive assumption that F be strongly (or inverse strongly) monotone. The extragradient

[†]Department of Cybernetics, Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska Str., Kyiv, 01601, Ukraine (y.malitsky@gmail.com).

method proposed by Korpelevich and Antipin [1, 11], according to the following formula, overcomes this difficulty

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

where $\lambda \in (0, \frac{1}{L})$. The extragradient method has received a great deal of attention by many authors, who improved it in various ways; see, e.g., [4, 7, 9, 17] and references therein. We restrict our attention only to one extension of the extragradient method. It was proposed in [4] by Censor, Gibali and Reich

$$\begin{cases} y_n = P_C(x_n - \lambda F(x_n)) \\ T_n = \{w \in H \mid \langle x_n - \lambda F(x_n) - y_n, w - y_n \rangle \leq 0\} \\ x_{n+1} = P_{T_n}(x_n - \lambda F(y_n)), \end{cases} \quad (1.3)$$

where $\lambda \in (0, \frac{1}{L})$. Since the second projection in (1.3) can be found in a closed form, this method is more applicable when a projection onto the closed convex set C is a nontrivial problem.

As an alternative to the extragradient method or its modifications is the following remarkable scheme proposed by Tseng in [20]

$$\begin{cases} y_n = P_C(x_n - \lambda F(x_n)) \\ x_{n+1} = y_n + \lambda(F(x_n) - F(y_n)), \end{cases} \quad (1.4)$$

where $\lambda \in (0, \frac{1}{L})$. Both algorithms (1.3) and (1.4) have the same complexity per iteration: we need to compute one projection onto the set C and two values of F .

Popov in his work [16] proposed an ingenious method, which is similar to the extragradient method, but uses on every iteration only one value of the mapping F . Using the idea from [4, 12], Malitsky and Semenov improved Popov algorithm. They presented in [13] the following algorithm

$$\begin{cases} T_n = \{w \in H \mid \langle x_n - \lambda F(y_{n-1}) - y_n, w - y_n \rangle \leq 0\} \\ x_{n+1} = P_{T_n}(x_n - \lambda F(y_n)) \\ y_{n+1} = P_C(x_{n+1} - \lambda F(y_n)), \end{cases} \quad (1.5)$$

where $\lambda \in (0, \frac{1}{3L}]$. It is easy to see that this method needs only one projection onto the set C (as in (1.3) or (1.4)) and only one value of F per iteration. The latter makes algorithm (1.5) very attractive for cases when a computation of operator F is expensive. This often happens, for example, in a huge-scale VIP or VIP that arises from optimal control.

In this work we propose the following scheme

$$x_{n+1} = P_C(x_n - \lambda F(2x_n - x_{n-1})), \quad (1.6)$$

where $\lambda \in (0, \frac{\sqrt{2}-1}{L})$. Again we see that both algorithms (1.5) and (1.6) have the same computational complexity per iteration, but the latter has more simple and elegant structure. Algorithm (1.6) reminds the projected gradient method, however, a value of a gradient is taken in the point that is a reflection of x_{n-1} in x_n . Preliminary results from numerical experiments, comparing algorithm (1.6) to others, are promising. Also note that a simple structure of (1.6) allows us to obtain a nonstationary variant of (1.6) with variable stepsize in a very convenient way.

The paper is organized as follows. Section 2 presents some basic useful facts which we use throughout the paper. In Section 3 we prove the convergence of the

method (1.6). Under some more restrictive assumption we also establish its rate of convergence. In Section 4 we present the nonstationary Algorithm 4.2 which is more flexible, since it does not use the Lipschitz constant of the mapping F . This makes it much more convenient in practical applications than Algorithm 1.6. Section 5 contains the results of numerical experiments.

2. Preliminaries. The next three statements are classical. For the proof we refer the reader to [2, 3].

LEMMA 2.1. *Let M be nonempty closed convex set in H , $x \in H$. Then*

- (i) $\langle P_M x - x, y - P_M x \rangle \geq 0 \quad \forall y \in M$;
- (ii) $\|P_M x - y\|^2 \leq \|x - y\|^2 - \|x - P_M x\|^2 \quad \forall y \in M$.

LEMMA 2.2 (Minty). *Assume that $F : C \rightarrow H$ is a continuous and monotone mapping. Then x^* is a solution of (1.1) iff x^* is a solution of the following problem*

$$\text{find } x \in C \quad \text{such that} \quad \langle F(y), y - x \rangle \geq 0 \quad \forall y \in C.$$

REMARK 2.1. *The solution set S of variational inequality (1.1) is closed and convex.*

As usual, the symbol \rightharpoonup denotes a weak convergence in H .

LEMMA 2.3 (Opial). *Let (x_n) be a sequence in H such that $x_n \rightharpoonup x$. Then for all $y \neq x$*

$$\liminf_{n \rightarrow \infty} \|x_n - x\| < \liminf_{n \rightarrow \infty} \|x_n - y\|.$$

The proofs of the next four statements are omitted by their simplicity.

LEMMA 2.4. *Let M be nonempty closed convex set in H , $x \in H$ and $\bar{x} = P_M x$. Then for all $\lambda > 0$*

$$P_M(\bar{x} + \lambda(x - \bar{x})) = \bar{x}.$$

LEMMA 2.5. *Let u, v be in H . Then*

$$\|2u - v\|^2 = 2\|u\|^2 - \|v\|^2 + 2\|u - v\|^2.$$

LEMMA 2.6. *Let (a_n) and (b_n) be two real sequences. Then*

$$\liminf_{n \rightarrow \infty} a_n + \liminf_{n \rightarrow \infty} b_n \leq \liminf_{n \rightarrow \infty} (a_n + b_n).$$

LEMMA 2.7. *Let $(a_n), (b_n)$ be two nonnegative real sequences such that*

$$a_{n+1} \leq a_n - b_n.$$

Then (a_n) is bounded and $\lim_{n \rightarrow \infty} b_n = 0$.

In order to establish the rate of convergence, we need the following

LEMMA 2.8. *Let $(a_n), (b_n)$ be two nonnegative real sequences and $\alpha_0, \beta \in (0, 1)$ such that for all $\alpha \in (0, \alpha_0)$ the following holds*

$$a_{n+1} + b_{n+1} \leq (1 - 2\alpha)a_n + \alpha a_{n-1} + \beta b_n. \quad (2.1)$$

Then there exist $\gamma \in (0, 1)$ and $M > 0$ such that $a_n \leq \gamma^n M$ for any $n > 0$.

Proof. First, note that a simple calculus ensures that $f(x) = \frac{1-2x+\sqrt{1+4x^2}}{2}$ is decreasing on $(0, 1)$. Since f is continuous and $f(0) = 1$, we can choose $\alpha \in (0, \alpha_0)$

such that $f(\alpha) \geq \beta$. Our next task is to find some $\gamma \in (0, 1)$ and $\delta > 0$ such that (2.1) with α defined as above will be equivalent to the inequality

$$a_{n+1} + \delta a_n + b_{n+1} \leq \gamma(a_n + \delta a_{n-1}) + \beta b_n.$$

It is easy to check that such numbers are

$$\begin{aligned} \gamma &= \frac{1 - 2\alpha + \sqrt{1 + 4\alpha^2}}{2} \in (0, 1), \\ \delta &= \frac{-1 + 2\alpha + \sqrt{1 + 4\alpha^2}}{2} > 0. \end{aligned}$$

Then by $\gamma = f(\alpha) \geq \beta$ we can conclude that

$$a_{n+1} + \delta a_n + b_{n+1} \leq \gamma(a_n + \delta a_{n-1} + b_n).$$

Iterating the last inequality simply leads to the desired result

$$a_{n+1} + \delta a_n + b_{n+1} \leq \gamma(a_n + \delta a_{n-1} + b_n) \leq \dots \leq \gamma^n(a_1 + \delta a_0 + b_1) = \gamma^n M,$$

where $M = a_1 + \delta a_0 + b_1 > 0$. \square

3. Algorithm and its convergence. We first note that solutions of (1.1) coincide with zeros of the following projected residual function:

$$r(x, y) := \|y - P_C(x - \lambda F(y))\| + \|x - y\|,$$

where λ is some positive number. Now we formally state our algorithm.

ALGORITHM 3.1.

1. Choose $x_0 = y_0 \in H$ and $\lambda \in (0, \frac{\sqrt{2}-1}{L})$.

2. Given the current iterate x_n and y_n , compute

$$x_{n+1} = P_C(x_n - \lambda F(y_n))$$

3. If $r(x_n, y_n) = 0$ then stop: $x_n = y_n = x_{n+1}$ is a solution. Otherwise compute

$$y_{n+1} = 2x_{n+1} - x_n,$$

set $n := n + 1$ and return to step 2.

Next lemma is central to our proof of the convergence theorem.

LEMMA 3.1. Let (x_n) and (y_n) be two sequences generated by Algorithm 3.1 and let $z \in S$. Then

$$\begin{aligned} \|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - (1 - \lambda L(1 + \sqrt{2})) \|x_n - x_{n-1}\|^2 + \lambda L \|x_n - y_{n-1}\|^2 \\ &\quad - (1 - \sqrt{2}\lambda L) \|x_{n+1} - y_n\|^2 - 2\lambda \langle F(z), y_n - z \rangle. \end{aligned} \quad (3.1)$$

Proof. By Lemma 2.1 we have

$$\begin{aligned} \|x_{n+1} - z\|^2 &\leq \|x_n - \lambda F(y_n) - z\|^2 - \|x_n - \lambda F(y_n) - x_{n+1}\|^2 \\ &= \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 - 2\lambda \langle F(y_n), x_{n+1} - z \rangle. \end{aligned} \quad (3.2)$$

Since F is monotone, $2\lambda \langle F(y_n) - F(z), y_n - z \rangle \geq 0$. Thus, adding this item to the right side of (3.2), we get

$$\begin{aligned} \|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 + 2\lambda \langle F(y_n), y_n - x_{n+1} \rangle - 2\lambda \langle F(z), y_n - z \rangle \\ &= \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 + 2\lambda \langle F(y_n) - F(y_{n-1}), y_n - x_{n+1} \rangle \\ &\quad + 2\lambda \langle F(y_{n-1}), y_n - x_{n+1} \rangle - 2\lambda \langle F(z), y_n - z \rangle. \end{aligned} \quad (3.3)$$

As $x_{n+1}, x_{n-1} \in C$, we have by Lemma 2.1

$$\begin{aligned}\langle x_n - x_{n-1} + \lambda F(y_{n-1}), x_n - x_{n+1} \rangle &\leq 0, \\ \langle x_n - x_{n-1} + \lambda F(y_{n-1}), x_n - x_{n-1} \rangle &\leq 0.\end{aligned}$$

Adding these two inequalities yields

$$\langle x_n - x_{n-1} + \lambda F(y_{n-1}), y_n - x_{n+1} \rangle \leq 0,$$

from which we conclude

$$\begin{aligned}2\lambda \langle F(y_{n-1}), y_n - x_{n+1} \rangle &\leq 2 \langle x_n - x_{n-1}, x_{n+1} - y_n \rangle = 2 \langle y_n - x_n, x_{n+1} - y_n \rangle \\ &= \|x_{n+1} - x_n\|^2 - \|x_n - y_n\|^2 - \|x_{n+1} - y_n\|^2, \quad (3.4)\end{aligned}$$

since $x_n - x_{n-1} = y_n - x_n$.

We next turn to estimating $\langle F(y_n) - F(y_{n-1}), x_{n+1} - y_n \rangle$. It follows that

$$\begin{aligned}2\lambda \langle F(y_n) - F(y_{n-1}), y_n - x_{n+1} \rangle &\leq 2\lambda L \|y_n - y_{n-1}\| \|x_{n+1} - y_n\| \\ &\leq \lambda L \left(\frac{1}{\sqrt{2}} \|y_n - y_{n-1}\|^2 + \sqrt{2} \|x_{n+1} - y_n\|^2 \right) \\ &\leq \lambda L \frac{1}{\sqrt{2}} \left((2 + \sqrt{2}) \|y_n - x_n\|^2 + \sqrt{2} \|x_n - y_{n-1}\|^2 \right) + \sqrt{2}\lambda L \|x_{n+1} - y_n\|^2 \\ &= \lambda L (1 + \sqrt{2}) \|y_n - x_n\|^2 + \lambda L \|x_n - y_{n-1}\|^2 + \sqrt{2}\lambda L \|x_{n+1} - y_n\|^2. \quad (3.5)\end{aligned}$$

Using (3.4) and (3.5), we deduce in (3.3) that

$$\begin{aligned}\|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - (1 - \lambda L (1 + \sqrt{2})) \|x_n - y_n\|^2 + \lambda L \|x_n - y_{n-1}\|^2 \\ &\quad - (1 - \sqrt{2}\lambda L) \|x_{n+1} - y_n\|^2 - 2\lambda \langle F(z), y_n - z \rangle.\end{aligned}$$

which completes the proof. \square

Now we can state and prove our main convergence result.

THEOREM 3.2. *Assume that (C1)–(C3) hold. Then any sequence (x_n) generated by Algorithm 3.1 weakly converges to a solution of (1.1).*

Proof. Let us show that the sequence (x_n) is bounded. Fix any $z \in S$. Since $1 - \sqrt{2}\lambda L > \lambda L$ and

$$\begin{aligned}\langle F(z), y_n - z \rangle &= 2 \langle F(z), x_n - z \rangle - \langle F(z), x_{n-1} - z \rangle \\ &\geq \langle F(z), x_n - z \rangle - \langle F(z), x_{n-1} - z \rangle,\end{aligned}$$

we can deduce from inequality (3.1) that

$$\begin{aligned}\|x_{n+1} - z\|^2 + \lambda L \|x_{n+1} - y_n\|^2 + 2\lambda \langle F(z), x_n - z \rangle \\ \leq \|x_n - z\|^2 + \lambda L \|x_n - y_{n-1}\|^2 + 2\lambda \langle F(z), x_{n-1} - z \rangle \\ - (1 - \lambda L (1 + \sqrt{2})) \|x_n - x_{n-1}\|^2.\end{aligned} \quad (3.6)$$

For $n \geq 2$ let

$$\begin{aligned}a_n &= \|x_n - z\|^2 + \lambda L \|x_n - y_{n-1}\|^2 + 2\lambda \langle F(z), x_{n-1} - z \rangle, \\ b_n &= (1 - \lambda L (1 + \sqrt{2})) \|x_n - x_{n-1}\|^2.\end{aligned}$$

Then we can rewrite inequality (3.6) as $a_{n+1} \leq a_n - b_n$. By Lemma 2.7, we conclude that (a_n) is bounded and $\lim_{n \rightarrow \infty} \|x_n - x_{n-1}\| = 0$. Therefore, the sequences $(\|x_n - z\|)$ and hence (x_n) are also bounded. From the inequality

$$\|x_{n+1} - y_n\| \leq \|x_{n+1} - x_n\| + \|x_n - y_n\| = \|x_{n+1} - x_n\| + \|x_n - x_{n-1}\|$$

we also have that $\lim_{n \rightarrow \infty} \|x_{n+1} - y_n\| = 0$.

As (x_n) is bounded, there exist a subsequence (x_{n_i}) of (x_n) such that (x_{n_i}) converges weakly to some $x^* \in H$. It is clear that (y_{n_i}) is also convergent to that x^* . We show $x^* \in S$. From Lemma 2.1 it follows that

$$\langle x_{n_i+1} - x_{n_i} + \lambda F(y_{n_i}), y - x_{n_i+1} \rangle \geq 0 \quad \forall y \in C.$$

From this we conclude that, for all $y \in C$,

$$\begin{aligned} 0 &\leq \langle x_{n_i+1} - x_{n_i}, y - x_{n_i+1} \rangle + \lambda \langle F(y_{n_i}), y - y_{n_i} \rangle + \lambda \langle F(y_{n_i}), y_{n_i} - x_{n_i+1} \rangle \\ &\leq \langle x_{n_i+1} - x_{n_i}, y - x_{n_i+1} \rangle + \lambda \langle F(y), y - y_{n_i} \rangle + \lambda \langle F(y_{n_i}), y_{n_i} - x_{n_i+1} \rangle. \end{aligned} \quad (3.7)$$

In the last inequality we used condition (C2). Taking the limit as $i \rightarrow \infty$ in (3.7) and using that $\lim_{i \rightarrow \infty} \|x_{n_i+1} - x_{n_i}\| = \lim_{i \rightarrow \infty} \|y_{n_i+1} - y_{n_i}\| = 0$, we obtain

$$0 \leq \langle F(y), y - x^* \rangle \quad \forall y \in C,$$

which implies by Lemma 2.2 that $x^* \in S$.

Let us show $x_n \rightarrow x^*$. From (3.6) it follows that the sequence (a_n) is monotone for any $z \in S$. Taking into account its boundedness we deduce that it is convergent. At last, the sequence $(\|x_n - y_{n-1}\|^2)$ is also convergent, therefore, $(\|x_n - z\|^2 + 2\lambda \langle F(z), x_{n-1} - z \rangle)$ is convergent.

We want to prove that (x_n) is weakly convergent. On the contrary, assume that the sequence (x_n) has at least two weak cluster points $\bar{x} \in S$ and $\tilde{x} \in S$ such that $\bar{x} \neq \tilde{x}$. Let (x_{n_k}) be a sequence such that $x_{n_k} \rightarrow \bar{x}$ as $k \rightarrow \infty$. Then by Lemma 2.6 and 2.3 we have

$$\begin{aligned} &\lim_{n \rightarrow \infty} (\|x_n - \bar{x}\|^2 + 2\lambda \langle F(\bar{x}), x_n - \bar{x} \rangle) \\ &= \lim_{k \rightarrow \infty} (\|x_{n_k} - \bar{x}\|^2 + 2\lambda \langle F(\bar{x}), x_{n_k} - \bar{x} \rangle) = \lim_{k \rightarrow \infty} \|x_{n_k} - \bar{x}\|^2 \\ &= \liminf_{k \rightarrow \infty} \|x_{n_k} - \bar{x}\|^2 < \liminf_{k \rightarrow \infty} \|x_{n_k} - \tilde{x}\|^2 \\ &\leq \liminf_{k \rightarrow \infty} \|x_{n_k} - \tilde{x}\|^2 + 2\lambda \liminf_{k \rightarrow \infty} \langle F(\tilde{x}), x_{n_k} - \tilde{x} \rangle \\ &\leq \liminf_{k \rightarrow \infty} (\|x_{n_k} - \tilde{x}\|^2 + 2\lambda \langle F(\tilde{x}), x_{n_k} - \tilde{x} \rangle) \\ &= \lim_{n \rightarrow \infty} (\|x_n - \tilde{x}\|^2 + 2\lambda \langle F(\tilde{x}), x_n - \tilde{x} \rangle). \end{aligned}$$

We can now proceed analogously to the proof that

$$\lim_{n \rightarrow \infty} (\|x_n - \tilde{x}\|^2 + 2\lambda \langle F(\tilde{x}), x_n - \tilde{x} \rangle) < \lim_{n \rightarrow \infty} (\|x_n - \bar{x}\|^2 + 2\lambda \langle F(\bar{x}), x_n - \bar{x} \rangle),$$

which is impossible. Hence we can conclude that x_n weakly converges to some $x^* \in S$. \square

It is well-known (see [19]) that under some suitable conditions the extragradient method has R-linear rate of convergence. In the following theorem we show that our method has the same rate of convergence under a strongly monotonicity assumption of the mapping F , i.e.,

$$(C2^*) \quad \langle F(x) - F(y), x - y \rangle \geq m \|x - y\|^2 \quad \forall x, y \in H$$

for some $m > 0$.

THEOREM 3.3. *Assume that (C1), (C2*), (C3) hold. Then any sequence (x_n) generated by Algorithm 3.1 converges to the solution of (1.1) at least R-linearly.*

Proof. Since F is strongly monotone, (1.1) has a unique solution, which we denote by z . Note that by Lemma 2.5

$$\|y_n - z\|^2 = 2 \|x_n - z\|^2 - \|x_{n-1} - z\|^2 + 2 \|x_n - x_{n-1}\|^2 \geq 2 \|x_n - z\|^2 - \|x_{n-1} - z\|^2$$

From this and from (C2*) we conclude that, for all $m_1 \in (0, m]$,

$$\begin{aligned} & 2\lambda(\langle F(y_n) - F(z), y_n - z \rangle - m_1(2\|x_n - z\|^2 - \|x_{n-1} - z\|^2)) \\ & \geq 2\lambda(\langle F(y_n) - F(z), y_n - z \rangle - m\|y_n - z\|^2) \geq 0. \end{aligned} \quad (3.8)$$

From now on, let m_1 be any number in $(0, m]$. Then adding the left part of (3.8) to the right side of (3.2), we get

$$\begin{aligned} \|x_{n+1} - z\|^2 & \leq \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 + 2\lambda \langle F(y_n), y_n - x_{n+1} \rangle \\ & \quad - 2\lambda \langle F(z), y_n - z \rangle - 2\lambda m_1(2\|x_n - z\|^2 - \|x_{n-1} - z\|^2) \\ & = (1 - 4\lambda m_1) \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 + 2\lambda m_1 \|x_{n-1} - z\|^2 \\ & \quad + 2\lambda \langle F(y_n) - F(y_{n-1}), y_n - x_{n+1} \rangle + 2\lambda \langle F(y_{n-1}), y_n - x_{n+1} \rangle \\ & \quad - 2\lambda \langle F(z), y_n - z \rangle. \end{aligned}$$

For items $\langle F(y_{n-1}), y_n - x_{n+1} \rangle$ and $\langle F(y_n) - F(y_{n-1}), y_n - x_{n+1} \rangle$ we use estimations (3.4) and (3.5) from Lemma 3.1. Therefore, we obtain

$$\begin{aligned} \|x_{n+1} - z\|^2 & \leq (1 - 4\lambda m_1) \|x_n - z\|^2 + 2\lambda m_1 \|x_{n-1} - z\|^2 \\ & \quad - (1 - \lambda L(1 + \sqrt{2})) \|x_n - x_{n-1}\|^2 - (1 - \sqrt{2}\lambda L) \|x_{n+1} - y_n\|^2 \\ & \quad + \lambda L \|x_n - y_{n-1}\|^2 - 2\lambda \langle F(z), y_n - z \rangle. \end{aligned}$$

From the last inequality it follows that

$$\begin{aligned} & \|x_{n+1} - z\|^2 + (1 - \sqrt{2}\lambda L) \|x_{n+1} - y_n\|^2 + 4\lambda \langle F(z), x_n - z \rangle \leq (1 - 4\lambda m_1) \|x_n - z\|^2 \\ & \quad + 2\lambda m_1 \|x_{n-1} - z\|^2 + \lambda L \|x_n - y_{n-1}\|^2 + 2\lambda \langle F(z), x_{n-1} - z \rangle \\ & \leq (1 - 4\lambda m_1) \|x_n - z\|^2 + 2\lambda m_1 \|x_{n-1} - z\|^2 \\ & + \max\left\{\frac{\lambda L}{1 - \sqrt{2}\lambda L}, \frac{1}{2}\right\} ((1 - \sqrt{2}\lambda L) \|x_n - y_{n-1}\|^2 + 4\lambda \langle F(z), x_{n-1} - z \rangle). \end{aligned} \quad (3.9)$$

In the first inequality we used that $\langle F(z), y_n - z \rangle = 2\langle F(z), x_n - z \rangle - \langle F(z), x_{n-1} - z \rangle$, and in the second we used that

$$\begin{aligned} \lambda L & \leq (1 - \sqrt{2}\lambda L) \max\left\{\frac{\lambda L}{1 - \sqrt{2}\lambda L}, \frac{1}{2}\right\}, \\ 2 & \leq 4 \max\left\{\frac{\lambda L}{1 - \sqrt{2}\lambda L}, \frac{1}{2}\right\}. \end{aligned}$$

Set

$$\begin{aligned} a_n & = \|x_n - z\|^2, \\ b_n & = (1 - \sqrt{2}\lambda L) \|x_n - y_{n-1}\|^2 + 4\lambda \langle F(z), x_{n-1} - z \rangle, \\ \beta & = \max\left\{\frac{\lambda L}{1 - \sqrt{2}\lambda L}, \frac{1}{2}\right\}, \\ \alpha & = 2\lambda m_1 \\ \alpha_0 & = 2\lambda m. \end{aligned}$$

As m_1 is arbitrary in $(0, m]$, we can rewrite (3.9) in the new notation as

$$a_{n+1} + b_{n+1} \leq (1 - 2\alpha)a_n + \alpha a_{n-1} + \beta b_n \quad \forall \alpha \in (0, \alpha_0].$$

Since $\beta < 1$, we can conclude by Lemma 2.1 that $a_n \leq \gamma^n M$ for some $\gamma \in (0, 1)$ and $M > 0$. This means that (x_n) converges to z at least R-linearly. \square

4. Modified Algorithm. The main shortcoming of all algorithms mentioned in §1 is a requirement to know the Lipschitz constant or at least to know some estimation of it. Usually it is difficult to estimate the Lipschitz constant more or less precisely, thus stepsizes will be quite tiny and, of course, this is not practical. For this reason, algorithms with constant stepsize are not applicable in most cases of interest. The usual approaches to overcome this difficulty consist in some prediction of a stepsize with its further correction (see [9, 14, 20]) or in a usage of Armijo-type linesearch procedure along a feasible direction (see [7, 17]). Usually the latter approach is more effective, since very often the former approach requires too many projections onto the feasible set per iteration.

Nevertheless, our modified method uses the prediction-correction strategy. However, in contrast to algorithms in [9, 14, 20], we need at most two projections per iteration. This is explained by the fact that for direction y_n in Algorithm 3.1 we use a very simple and cheap formula: $y_n = 2x_n - x_{n-1}$. Although we can not explain this theoretically, but numerical experiments show that cases with two projections per iteration are quite rare, so usually we have only one projection per iteration that is a drastic contrast to other existing methods.

Looking on the proof of Theorem 3.2, we can conclude that inequality (3.5) is the only place where we use Lipschitz constant L . Therefore, choosing λ_n such that the inequality $\lambda_n \|F(y_n) - F(y_{n-1})\| \leq \alpha \|y_n - y_{n-1}\|$ holds for some fixed $\alpha < 1$ we can obtain the similar estimation as in (3.5). All this leads us to the following

ALGORITHM 4.1.

$$\left\{ \begin{array}{l} \text{Choose } x_0 = y_0 \in H, \lambda_0 > 0, \alpha \in (0, \sqrt{2} - 1). \\ \text{Choose } \lambda_n \text{ s.t. } \lambda_n \|F(y_n) - F(y_{n-1})\| \leq \alpha \|y_n - y_{n-1}\|, n > 0. \\ x_{n+1} = P_C(x_n - \lambda_n F(y_n)), \\ y_{n+1} = 2x_{n+1} - x_n. \end{array} \right.$$

Although numerical results showed us effectiveness of Algorithm 4.1, we can not prove its convergence for all cases. Nevertheless, we want to notice that we did not find any example where Algorithm 4.1 did not work. Thus, even Algorithm 4.1 seems to be very reliable for many problems.

Now our task is to modify Algorithm 4.1 in a such way that we will be able to prove convergence of the obtained algorithm. For this we need to distinguish the good cases, where Algorithm 4.1 works well, from the bad ones, where it possibly does not.

From now on we adopt the convention that $0/0 = +\infty$. Clearly, it follows that $1/0 = +\infty$ as well. The following algorithm gets round the difficulty of bad cases of Algorithm 4.1.

ALGORITHM 4.2.

1. Choose $x_0 \in C$, $\lambda_{-1} > 0$, $\tau_0 = 1$, $\alpha \in (0, \sqrt{2} - 1)$ and some large $\bar{\lambda} > 0$. Compute

$$\begin{aligned} y_0 &= P_C(x_0 - \lambda_{-1} F(x_0)), \\ \lambda_0 &= \min \left\{ \frac{\alpha \|x_0 - y_0\|}{\|F(x_0) - F(y_0)\|}, \bar{\lambda} \right\}, \\ x_1 &= P_C(x_0 - \lambda_0 F(y_0)). \end{aligned}$$

2. Given x_n and x_{n-1} , set $\tau_n = 1$ and let $\lambda(y, \tau)$ be defined by

$$\lambda(y, \tau) := \min \left\{ \frac{\alpha \|y - y_{n-1}\|}{\|F(y) - F(y_{n-1})\|}, \frac{1 + \tau_{n-1}}{\tau} \lambda_{n-1}, \bar{\lambda} \right\}. \quad (4.1)$$

Compute

$$\begin{aligned} y_n &= 2x_n - x_{n-1}, \\ \lambda_n &= \lambda(y_n, \tau_n), \\ x_{n+1} &= P_C(x_n - \lambda_n F(y_n)). \end{aligned}$$

3. If $r(x_n, y_n) = 0$ then stop: $x_n = y_n = x_{n+1}$ is a solution. Otherwise compute

$$\begin{aligned} t_n &= -\|x_{n+1} - x_n\|^2 + 2\lambda_n \langle F(y_n), y_n - x_{n+1} \rangle + (1 - \alpha(1 + \sqrt{2})) \|x_n - y_n\|^2 \\ &\quad - \alpha \|x_n - y_{n-1}\|^2 + (1 - \sqrt{2}\alpha) \|x_{n+1} - y_n\|^2. \end{aligned}$$

4. If $t_n \leq 0$ then set $n := n + 1$ and go to step 2. Otherwise we have two cases $\lambda_n \geq \lambda_{n-1}$ and $\lambda_n < \lambda_{n-1}$.

(i) If $\lambda_n \geq \lambda_{n-1}$ then choose $\lambda'_n \in [\lambda_{n-1}, \lambda_n]$ such that

$$\|\lambda'_n F(y_n) - \lambda_{n-1} F(y_{n-1})\| \leq \alpha \|y_n - y_{n-1}\|. \quad (4.2)$$

Compute

$$x_{n+1} = P_C(x_n - \lambda'_n F(y_n)).$$

Set $\lambda_n := \lambda'_n$, $n := n + 1$ and go to step 2.

(ii) If $\lambda_n < \lambda_{n-1}$ then find $\tau'_n \in (0, 1]$ such that

$$\begin{aligned} y'_n &= x_n + \tau'_n(x_n - x_{n-1}) \\ \lambda(y'_n, \tau'_n) &\geq \tau'_n \lambda_{n-1}. \end{aligned} \quad (4.3)$$

Then choose $\lambda'_n \in [\tau'_n \lambda_{n-1}, \lambda(y'_n, \tau'_n)]$ such that

$$\|\lambda'_n F(y'_n) - \tau'_n \lambda_{n-1} F(y_{n-1})\| \leq \alpha \|y'_n - y_{n-1}\|. \quad (4.4)$$

Compute

$$x_{n+1} = P_C(x_n - \lambda'_n F(y'_n))$$

Set $\lambda_n := \lambda'_n$, $\tau_n := \tau'_n$, $y_n := y'_n$, $n := n + 1$ and go to step 2.

It is clear that on every iteration in Algorithm 4.2 we need to use a residual function with different λ , namely, $r(x_n, y_n) = \|y_n - P_C(x_n - \lambda_n F(y_n))\| + \|x_n - y_n\|$.

First, let us show that Algorithm 4.2 is correct, i.e., it is always possible to choose λ'_n and τ'_n on steps (4.i) and (4.ii). For this we need two simple lemmas.

LEMMA 4.1. *Step (4.i) in Algorithm 4.2 is well-defined.*

Proof. From the inequality

$$\|\lambda_{n-1} F(y_n) - \lambda_{n-1} F(y_{n-1})\| \leq \lambda_n \|F(y_n) - F(y_{n-1})\| \leq \alpha \|y_n - y_{n-1}\|$$

we can see that it is sufficient to take $\lambda'_n = \lambda_{n-1}$. (However, it seems better for practical reasons to choose λ'_n as great as possible). \square

LEMMA 4.2. *Step (4.ii) in Algorithm 4.2 is well-defined.*

Proof. First, let us show that $\lambda(y, \tau) \geq \alpha/L$ for all $y, y_n \in H$ and $\tau \in (0, 1]$. It is clear that for every $u, v \in H$

$$\frac{\alpha \|u - v\|}{\|F(u) - F(v)\|} \geq \frac{\alpha \|u - v\|}{L \|u - v\|} = \frac{\alpha}{L}.$$

Then $\lambda_0 \geq \alpha/L$ and hence by induction

$$\lambda(y, \tau) := \min \left\{ \frac{\alpha \|y - y_{n-1}\|}{\|F(y) - F(y_{n-1})\|}, \frac{1 + \tau_{n-1}}{\tau} \lambda_{n-1}, \bar{\lambda} \right\} \geq \frac{\alpha}{L}.$$

Therefore, it is sufficient to take $\tau'_n = \frac{\alpha}{\lambda_{n-1}L} \in (0, 1]$. (But, as above, it seems better to choose τ'_n as great as possible).

At last, we can prove the existence of $\lambda'_n \in [\tau'_n \lambda_{n-1}, \lambda(y'_n, \tau'_n)]$ such that (4.4) will hold by the same arguments as in Lemma 4.1. \square

The following lemma yields an analogous inequality to (3.1).

LEMMA 4.3. *Let (x_n) and (y_n) be two sequences generated by Algorithm 4.2 and let $z \in S$, $\alpha \in (0, \sqrt{2} - 1)$. Then*

$$\begin{aligned} \|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - (1 - \alpha(1 + \sqrt{2})) \|x_n - y_n\|^2 - (1 - \sqrt{2}\alpha) \|x_{n+1} - y_n\|^2 \\ &\quad + \alpha \|x_n - y_{n-1}\|^2 - 2\lambda_n \langle F(z), y_n - z \rangle. \end{aligned} \quad (4.5)$$

Proof. Proceeding analogously as in (3.2) and in (3.3), we get

$$\begin{aligned} \|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 + 2\lambda_n \langle F(y_n) - F(y_{n-1}), y_n - x_{n+1} \rangle \\ &\quad + 2\lambda_n \langle F(y_{n-1}), y_n - x_{n+1} \rangle - 2\lambda_n \langle F(z), y_n - z \rangle. \end{aligned} \quad (4.6)$$

The same arguments as in (3.4) yield

$$\begin{aligned} 2\lambda_{n-1} \langle F(y_{n-1}), y_n - x_{n+1} \rangle &\leq 2 \langle x_n - x_{n-1}, x_{n+1} - y_n \rangle = 2 \langle y_n - x_n, x_{n+1} - y_n \rangle \\ &= \|x_{n+1} - x_n\|^2 - \|x_n - y_n\|^2 - \|x_{n+1} - y_n\|^2. \end{aligned} \quad (4.7)$$

Using (3.5) and the inequality $\lambda_n \|F(y_n) - F(y_{n-1})\| \leq \alpha \|y_n - y_{n-1}\|$, we obtain

$$\begin{aligned} 2\lambda_n \langle F(y_n) - F(y_{n-1}), y_n - x_{n+1} \rangle &\leq 2\alpha \|y_n - y_{n-1}\| \|x_{n+1} - y_n\| \\ &\leq \alpha(1 + \sqrt{2}) \|y_n - x_n\|^2 + \alpha \|x_n - y_{n-1}\|^2 + \sqrt{2}\alpha \|x_{n+1} - y_n\|^2. \end{aligned} \quad (4.8)$$

Applying (4.7) and (4.8) in (4.6), we conclude that

$$\begin{aligned} \|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 \\ &\quad + \alpha((1 + \sqrt{2}) \|x_n - y_n\|^2 + \|x_n - y_{n-1}\|^2 + \sqrt{2} \|x_{n+1} - y_n\|^2) \\ &\quad + \frac{\lambda_n}{\lambda_{n-1}} (\|x_{n+1} - x_n\|^2 - \|x_n - y_n\|^2 - \|x_{n+1} - y_n\|^2) \\ &\quad - 2\lambda_n \langle F(z), y_n - z \rangle. \end{aligned} \quad (4.9)$$

Now we have a lot of cases

- (a) $\lambda_n = \lambda_{n-1}$;
- (b) $\lambda_n < \lambda_{n-1}$ and $\|x_{n+1} - x_n\|^2 > \|x_n - y_n\|^2 - \|x_{n+1} - y_n\|^2$;
- (c) $\lambda_n > \lambda_{n-1}$ and $\|x_{n+1} - x_n\|^2 \geq \|x_n - y_n\|^2 - \|x_{n+1} - y_n\|^2$;
- (d) $\lambda_n < \lambda_{n-1}$ and $\|x_{n+1} - x_n\|^2 \leq \|x_n - y_n\|^2 - \|x_{n+1} - y_n\|^2$;
- (e) $\lambda_n > \lambda_{n-1}$ and $\|x_{n+1} - x_n\|^2 < \|x_n - y_n\|^2 - \|x_{n+1} - y_n\|^2$.

Let us consider all them.

Case (a). It is quite obvious that when $\lambda_n = \lambda_{n-1}$ we obtain inequality (4.5).

Cases (b) and (e). In both cases we notice that

$$\left(1 - \frac{\lambda_n}{\lambda_{n-1}}\right) \|x_{n+1} - x_n\|^2 > \left(1 - \frac{\lambda_n}{\lambda_{n-1}}\right) (\|x_n - y_n\|^2 + \|x_{n+1} - y_n\|^2).$$

Thus, from (4.9) we conclude that

$$\begin{aligned}
\|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - \left(1 - \frac{\lambda_n}{\lambda_{n-1}}\right) \|x_{n+1} - x_n\|^2 \\
&\quad + \alpha \left((1 + \sqrt{2}) \|x_n - y_n\|^2 + \|x_n - y_{n-1}\|^2 + \sqrt{2} \|x_{n+1} - y_n\|^2 \right) \\
&\quad - \frac{\lambda_n}{\lambda_{n-1}} \left(\|x_n - y_n\|^2 + \|x_{n+1} - y_n\|^2 \right) - 2\lambda_n \langle F(z), y_n - z \rangle \\
&< \|x_n - z\|^2 - \|x_n - y_n\|^2 - \|x_{n+1} - y_n\|^2 - 2\lambda_n \langle F(z), y_n - z \rangle \\
&\quad + \alpha \left((1 + \sqrt{2}) \|x_n - y_n\|^2 + \|x_n - y_{n-1}\|^2 + \sqrt{2} \|x_{n+1} - y_n\|^2 \right) \\
&= \|x_n - z\|^2 - (1 - \alpha(1 + \sqrt{2})) \|x_n - y_n\|^2 - (1 - \sqrt{2}\alpha) \|x_{n+1} - y_n\|^2 \\
&\quad + \alpha \|x_n - y_{n-1}\|^2 - 2\lambda_n \langle F(z), y_n - z \rangle,
\end{aligned}$$

which is exactly inequality (4.5).

Observe that inequality (4.9) with condition $t_n \leq 0$ is equivalent to (4.5). So, from now on we assume that $t_n > 0$.

Case (c) and $t_n > 0$. It is clear that we can rewrite (4.6) (with λ'_n instead of λ_n) as

$$\begin{aligned}
\|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 + 2 \langle \lambda'_n F(y_n) - \lambda_{n-1} F(y_{n-1}), y_n - x_{n+1} \rangle \\
&\quad + 2\lambda_{n-1} \langle F(y_{n-1}), y_n - x_{n+1} \rangle - 2\lambda'_n \langle F(z), y_n - z \rangle \\
&\leq \|x_n - z\|^2 - \|x_{n+1} - x_n\|^2 + 2\alpha \|y_n - y_{n-1}\| \|y_n - x_{n+1}\| \\
&\quad + 2\lambda_{n-1} \langle F(y_{n-1}), y_n - x_{n+1} \rangle - 2\lambda'_n \langle F(z), y_n - z \rangle.
\end{aligned}$$

Now using (4.7) and the second inequality in (4.8), we get easily (4.5).

Case (d) and $t_n > 0$. Let $\lambda'_{n-1} = \tau'_n \lambda_{n-1}$ and $x'_{n-1} = x_n - \tau'_n (x_n - x_{n-1})$. Since $\lambda'_{n-1} \leq \lambda_{n-1}$, the point x'_{n-1} lies on the segment $[x_{n-1}, x_n]$ and the point $x'_{n-1} - \lambda'_{n-1} F(y_{n-1})$ lies on the segment $[x_n, x_{n-1} - \lambda_{n-1} F(y_{n-1})]$. Thus by Lemma 2.4

$$x_n = P_C(x'_{n-1} - \lambda'_{n-1} F(y_{n-1})).$$

Also we have that $x_{n+1} = P_C(x_n - \lambda'_n F(y'_n))$ and $y'_n = 2x_n - x'_{n-1}$. Applying the same arguments as above for this new x_{n+1} , we do not get into the Case (d), since $\lambda'_n \geq \lambda'_{n-1}$, and do not get into the Case (c), since

$$\|\lambda'_n F(y'_n) - \lambda'_{n-1} F(y_{n-1})\| \leq \alpha \|y'_n - y_{n-1}\|.$$

This means that we get

$$\begin{aligned}
\|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - (1 - \alpha(1 + \sqrt{2})) \|x_n - y'_n\|^2 - (1 - \sqrt{2}\alpha) \|x_{n+1} - y'_n\|^2 \\
&\quad + \alpha \|x_n - y_{n-1}\|^2 - 2\lambda'_n \langle F(z), y'_n - z \rangle.
\end{aligned}$$

Hence, for redefined values we obtain the desired inequality (4.5) and the proof is complete. \square

Before proceeding, we want to give some intuition to Algorithm 4.2. First, note that $\bar{\lambda}$ is the upper bound of (λ_n) . It guarantees that our stepsize does not go to infinity. The item $\frac{1+\tau_{n-1}}{\tau} \lambda_{n-1}$ in (4.1) provides us necessary condition to obtain the good estimation of $2\lambda_n \langle F(z), y_n - z \rangle$. This seems not very important, since in many cases $\langle F(z), y_n - z \rangle \geq 0$ and we can just not take into account this item in (4.5). But this condition also ensures that λ_n is not too large in comparison with λ_{n-1} . This may be useful if we get into the Case (c) or (d) in Lemma 4.3, since we will be able to perform (4.2) or (4.4) faster. Also note that usually we have $\lambda_n \leq 2\lambda_{n-1}$, since $\tau_n \neq 1$ only in the Case (d).

The inequality $t_n \leq 0$ ensures at once that on the n -th iteration inequality (4.5) holds. Therefore, we do not need to immerse into a branch of the cases in Algorithm 4.2 or Lemma 4.3. Also note that a former and a latter use different classifications of possible cases. In Lemma 4.3 the inequality $t_n \leq 0$ holds certainly for cases (a), (b), and (e). But it happens very often that even in such bad cases as (c) or (d) $t_n \leq 0$ also. Thus, by checking whether $t_n \leq 0$, we can avoid to get into the undesirable steps (4.i) or (4.ii).

If $t_n > 0$ and $\lambda_n \geq \lambda_{n-1}$, then we get into step (4.i). Actually, this case is not so bad. We have two possibilities here. The first one is to decrease λ_n until it satisfies (4.2). And the second one is just to set $\lambda'_n = \lambda_{n-1}$. The first one seems to be better, since we suppose, as usual, that a larger stepsize provides a smaller number of iterations (however, it needs some extra work to check condition (4.2)).

It only remains the most tedious case when $t_n > 0$ and $\lambda_n < \lambda_{n-1}$. In this case we show that it is always possible to change $y_n = 2x_n - x_{n-1}$ to some other y_n , defined by $y_n = (1 + \tau_n)x_n - \tau_n x_{n-1}$ with some $\tau_n \in (0, 1)$, which we choose to ensure that (4.3) and (4.4) hold. Next, with new y_n and λ_n we compute x_{n+1} as in step 2. The conditions (4.3) and (4.4) guarantee that for new t_n we have $t_n \leq 0$. However, when we go to the next iteration, we must not forget that y_n and λ_n have been changed.

Now we prove the convergence of Algorithm 4.2 in a quite similar way as in § 3.

THEOREM 4.4. *Assume that (C1)–(C3) hold. Then any sequences (x_n) and (y_n) generated by Algorithm 4.2 weakly converge to a solution of (1.1).*

Proof. Let us show that the sequence (x_n) is bounded. Fix any $z \in S$. Since $\tau_n \lambda_n \leq (1 + \tau_{n-1})\lambda_{n-1}$, we have that

$$\begin{aligned} \lambda_n \langle F(z), y_n - z \rangle &= \lambda_n \langle F(z), (1 + \tau_n)x_n - \tau_n x_{n-1} - z \rangle \\ &= \lambda_n(1 + \tau_n) \langle F(z), x_n - z \rangle - \lambda_n \tau_n \langle F(z), x_{n-1} - z \rangle \\ &\geq \lambda_n(1 + \tau_n) \langle F(z), x_n - z \rangle - \lambda_{n-1}(1 + \tau_{n-1}) \langle F(z), x_{n-1} - z \rangle. \end{aligned}$$

And hence we can deduce from inequality (4.5) that

$$\begin{aligned} \|x_{n+1} - z\|^2 + \alpha \|x_{n+1} - y_n\|^2 + 2\lambda_n(1 + \tau_n) \langle F(z), x_n - z \rangle \\ \leq \|x_n - z\|^2 + \alpha \|x_n - y_{n-1}\|^2 + 2\lambda_{n-1}(1 + \tau_{n-1}) \langle F(z), x_{n-1} - z \rangle \\ - (1 - \alpha(1 + \sqrt{2}))(\|x_n - y_n\|^2 + \|x_{n+1} - y_n\|^2). \end{aligned} \quad (4.10)$$

For $n \geq 1$ set

$$\begin{aligned} a_{n+1} &= \|x_{n+1} - z\|^2 + \alpha \|x_{n+1} - y_n\|^2 + 2\lambda_n(1 + \tau_n) \langle F(z), x_n - z \rangle, \\ b_n &= (1 - \alpha(1 + \sqrt{2}))(\|x_n - y_n\|^2 + \|x_{n+1} - y_n\|^2). \end{aligned}$$

With this notation we can rewrite inequality (4.10) as $a_{n+1} \leq a_n - b_n$. Thus by Lemma 2.7 we can conclude that (a_n) , and hence (x_n) , are bounded and $\lim_{n \rightarrow \infty} b_n = 0$. From the latter it follows immediately that

$$\lim_{n \rightarrow \infty} \|x_{n+1} - y_n\| = 0 \quad \text{and} \quad \lim_{n \rightarrow \infty} \|x_{n+1} - x_n\| = 0.$$

From this point we need to proceed in much the same way as in Theorem 3.2. Thus, to avoid the repetition, we restrict our attention to only the place where arguments differ. This is inequality (3.7) where we want to go to the limit as $i \rightarrow \infty$. Recall that $\bar{\lambda}$ and $\frac{\alpha}{L}$ are the upper and the lower bounds of the sequence (λ_n) , respectively (the former follows from the definition of λ_n and the latter from Lemma 4.2). Consequently, we can conclude that the sequence (λ_n) has a cluster

point, and moreover, all its cluster points are not equal to zero. This means that when we go to the limit in (3.7) we obtain the same result as in Theorem 3.2. This is the desired conclusion. \square

Algorithm 4.2 allows several modifications. In particular, instead of checking whether $t_n \leq 0$, one can require at once that λ_n , which we choose in step 2, satisfies the following inequality

$$\|\lambda_n F(y_n) - \lambda_{n-1} F(y_{n-1})\| \leq \alpha \|y_n - y_{n-1}\|.$$

The arguments, used in steps (4.i) and (4.ii), show that is always possible to do (perhaps, with changed y_n). Although this algorithm have only one projection per iteration, our computational experience shows that it is less effective than Algorithm 4.2. This is explained again by the fact that very often $t_n \leq 0$ even for not suitable λ_n .

5. Numerical illustration. In this section we examine the benchmark of Alg. 3.1 and 4.2. For this, first, we present the performance of stationary algorithms on some affine problems. And then we compare nonstationary Alg. 4.2 with two well-known methods. For abbreviation we use the following.

Stationary methods:

- EGM for algorithm described by (1.2);
- SubEGM for algorithm described by (1.3);
- TBFM for algorithm described by (1.4);
- SubPM for algorithm described by (1.5).

Nonstationary methods:

- Alg. Solod.-Svait. for the algorithm of Solodov and Svaiter [17];
- TBFM-nst for the nonstationary backward-forward algorithm of Tseng [20].

Note that in [17] there were considered two algorithms, so we chose the second one with the better performance.

For a benchmark of all algorithms we included the number of iterations (`iter.`) and the time of execution (`time`). For the nonstationary algorithms we also added the total number (`np`) of projections and the total number (`nf`) of all values F that were evaluated.

We describe the test details below. Computations were performed using Wolfram Mathematica 8.0 on an Intel Core i3-2348M 2.3GHz running 64-bit Linux Mint 13. The time is measured in seconds using the intrinsic Mathematica function `Timing`. The projection onto the feasible set C was performed using the intrinsic Mathematica function `ArgMin` (for Problems 1, 4, and 5 the projection is explicit). The termination criteria are the following

$$\text{EGM, SubEGM, TBFM, Solod-Svait., TBFM-nst.} \quad \|x_n - y_n\| \leq \varepsilon, \quad (5.1)$$

$$\text{SubPM} \quad r(x_{n+1}, y_n) \leq \varepsilon, \quad (5.2)$$

$$\text{Alg. 3.1, 4.2} \quad r(x_n, y_n) \leq \varepsilon. \quad (5.3)$$

We do not use the simple criterion $\|x - P_C(x - \lambda F(x))\| \leq \varepsilon$ for Alg. 3.1, 4.2, and SubPM because we do not want to compute $P_C(x - \lambda F(x))$ extra. The better way is to verify that

$$\begin{aligned} \|y_n - P_C(y_n - \lambda F(y_n))\| &\leq \|x_{n+1} - y_n\| + \|x_{n+1} - P_C(y_n - \lambda F(y_n))\| \\ &\leq \|x_{n+1} - y_n\| + \|x_n - y_n\| = r(x_n, y_n). \end{aligned}$$

The last inequality holds, since P_C is nonexpansive. The similar argument works for SubPM. Moreover, we notice that termination criteria (5.2), (5.3) are stronger than (5.1). This means that the solutions generated by Alg. 3.1, 4.2, and SubPM are not worse (or even better) than the solutions generated by other algorithms if we measure the quality of the solution by the projected error bound.

m	EGM		SubEGM		SubPM		Alg. 3.1	
	iter.	time	iter.	time	iter.	time	iter.	time
500	129	1.3	129	1.3	109	0.6	92	0.5
1000	133	5.6	133	5.7	120	2.6	95	2
2000	138	22	138	22	121	10	98	8
4000	143	98	143	101	122	43	101	33

TABLE 5.1

Results for EGM, SubEGM, SubPM, and Alg 3.1 for Problem 1. $L = 1$, $\lambda = 0.4$, $\varepsilon = 10^{-3}$

m	EGM		SubEGM		TBFM		SubPM		Alg. 3.1	
	iter.	time	iter.	time	iter.	time	iter.	time	iter.	time
10	82	7.7	82	3.9	82	3.9	83	4.0	83	4.0
10	146	15.2	148	7.8	148	7.8	152	8.0	147	7.8
10	63	6.3	63	3.2	63	3.2	63	3.2	63	3.2
20	78	10.2	79	5.2	79	5.3	79	5.3	79	5.3
20	101	14.5	102	7.3	102	7.4	101	7.3	102	7.4
20	110	17.6	110	8.9	110	8.9	110	8.9	111	9.0
50	224	243	226	126	226	128	231	134	225	134
50	236	277	237	144	238	143	241	151	237	144
50	444	507	447	265	447	266	455	263	445	267

TABLE 5.2

Results for EGM, SubEGM, TBFM, SubPM, and Alg 3.1 for Problem 2. $L = \|M\|$, $\lambda = 0.4/L$, $\varepsilon = 10^{-3}$

Problem 1. Our first example is classical. The feasible set is $C = \mathbb{R}^m$ and $F(x) = Ax$, where A is a square matrix $m \times m$ defined by condition

$$a_{ij} = \begin{cases} -1, & j = m + 1 - i > i \\ 1, & j = m + 1 - i < i \\ 0, & \text{otherwise.} \end{cases}$$

This corresponds to a matrix whose secondary diagonal consists half of -1 s and half of 1 s, the rest of the elements being zero. For even m the zero vector is the problem solution of (1.1). This is a classical example of a problem where usual gradient method does not converge. The results of the numerical experiments are compiled in Table 5.1. We did not include the results of TBFM, since for this problem it is identical with EGM. For all tests we took $x^0 = (1, \dots, 1)$, $\varepsilon = 10^{-3}$ and $\lambda = 0.4$.

Problem 2. We take $F(x) = Mx + q$ with the matrix M randomly generated as suggested in [6]:

$$M = AA^T + B + D,$$

where every entry of the $m \times m$ matrix A and of the $m \times m$ skew-symmetric matrix B is uniformly generated from $(-5, 5)$, and every diagonal entry of the $m \times m$ diagonal D is uniformly generated from $(0, 0.3)$ (so M is positive definite), with every entry of q uniformly generated from $(-500, 0)$. The feasible set is

$$C = \{x \in \mathbb{R}_+^m \mid x_1 + x_2 + \dots + x_m = m\},$$

and the starting point is $x_0 = (1, \dots, 1)$. For this problem we took $L = \|M\|$. For every m , as shown in Table 5.2, we have generated three random samples with different choice of M and q . The error is $\varepsilon = 10^{-3}$.

As one can see from Table 5.2, EGM has the worst performance. This is explained by the presence of two projections per iteration, which played the key role, since for small m the computation of F is cheap. All others algorithms behave similarly. Also it is very likely that with an increasing of m they will continue to have approximately the same number of iterations. However, for large m the computation

x^0	ε	Alg. 4.2		Alg. Solod.-Svait.		TBFM-nst	
		iter. (np/nf)	time	iter. (np/nf)	time	iter. (np/nf)	time
(1, 1, 1, 1)	10^{-3}	36 (36/36)	1.0	6 (11/11)	0.4	62 (66/128)	2.1
	10^{-6}	72 (82/86)	2.2	6 (11/11)	0.4	156 (160/316)	4.9
(0.5, 0.5, 2, 1)	10^{-3}	41 (41/41)	1.1	18 (35/58)	1.4	69 (73/142)	2.2
	10^{-6}	75 (87/86)	2.4	—	> 50	163 (167/330)	5.0
random	10^{-3}	57 (57/58)	1.6	13 (25/49)	1.2	72 (77/149)	2.3
	10^{-6}	63 (67/65)	1.9	—	> 50	79 (84/163)	2.6
random	10^{-3}	40 (40/40)	1.0	20 (39/73)	1.5	66 (70/136)	2.0
	10^{-6}	71 (82/74)	2.0	—	> 50	160 (164/324)	4.9
random	10^{-3}	59 (59/59)	1.5	15 (29/40)	1.1	87 (91/178)	2.6
	10^{-6}	93 (99/105)	2.5	—	> 50	181 (185/366)	5.7

TABLE 5.3

Results for Alg. 4.2, Alg. Solod.-Svait., and TBFM-nst for Problem 3

of F will also matter, so for those cases we can expect that SubPM and Alg. 3.1 will have substantially better performance, compared with the others.

From now on, we present out test results for nonstationary methods. The choice of test problems for nonstationary algorithms is explained by a simple structure of the feasible set C . This is important for the better performance of TBFM-nst, since one must compute many projections per iteration, and Alg. Solod.-Svait., since on every iteration one must compute a projection onto the intersection of C with some hyperspace. (Of course, instead of a projection onto this intersection, we can project onto the set C . But in that case the stepsize will be smaller).

In the implementation of Alg. Solod.-Svait. we chose $\sigma = 0.3$, $\gamma = 0.5$, $\theta = 4$; in the implementation of TBFM-nst. we chose $\beta = 0.5$, $\theta = 0.9$ where the notation is taken from [17] and [20] respectively. For Alg. 4.2 we used $\alpha = 0.4$ and $\lambda_{-1} = 0.01$. Problem 3. Kojima-Shindo Nonlinear Complementarity Problem (NCP) was considered in [15] and in [8], where $m = 4$ and the mapping F is defined by

$$F(x_1, x_2, x_3, x_4) = \begin{bmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 10x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 9x_4 - 9 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{bmatrix}.$$

The feasible set is $C = \{x \in \mathbb{R}_+^4 \mid x_1 + x_2 + x_3 + x_4 = 4\}$. First, we chose some particular starting points: $x^0 = (1, 1, 1, 1)$ and $x^0 = (0.5, 0.5, 2, 1)$. And then we used different random points from C . For all starting points we had two tests: with $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-6}$. The results are summarized in Table 5.3. Here we aborted the evaluation of algorithm after 50 seconds.

As one can see, the ratio $nf/iter.$ for Alg. Solod.-Svait. is much higher than for Alg. 4.2. For this reason even with larger number of iterations Alg. 4.2 sometimes outperforms Alg. Solod.-Svait. Note that TBFM-nst has the worst performance, since on every iteration it must compute many nontrivial projections onto the simplex (this is exactly what we meant at the beginning of §4). The solution of this problem is not unique, but most of the times all algorithms converge to the $x^* = (1.225, 0, 0, 2.775)$. But for example for $x^0 = (1, 1, 1, 1)$ algorithms converge to the different solutions, this is the reason of such distinction in their performance. Also for $\varepsilon = 10^{-6}$ Alg. Solod.-Svait. often converges very slowly, that is why it does not execute in 50 seconds. In that time, Alg. 4.2 and TBFM-nst have not this drawback, since their stepsizes λ_n are separated from zero.

x^0	ε	Alg. 4.2		Alg. Solod.-Svait.		TBFM-nst	
		iter. (np/nf)	time	iter. (np/nf)	time	iter. (np/nf)	time
5	10^{-3}	20 (20/20)	0.008	18 (35/70)	0.012	22 (24/46)	0.007
	10^{-6}	43 (43/43)	0.014	34 (67/134)	0.018	47 (49/96)	0.019
50	10^{-3}	23 (24/26)	0.02	20 (39/78)	0.06	28 (30/58)	0.04
	10^{-6}	46 (47/49)	0.04	34 (67/134)	0.1	53 (55/108)	0.07
500	10^{-3}	27 (28/30)	0.6	22 (43/86)	1.6	32 (34/66)	1.0
	10^{-6}	50 (51/53)	1.4	36 (71/142)	2.7	58 (60/118)	1.8
1000	10^{-3}	28 (29/31)	1.4	22 (43/86)	4.8	33 (35/68)	2.8
	10^{-6}	51 (52/54)	2.6	36 (71/142)	7.4	59 (61/120)	5.1

TABLE 5.4

Results for Alg. 4.2 and Alg. Solod.-Svait., and TBFM-nst for Problem 4

Problem 4. This example was considered by Sun in [18], where

$$\begin{aligned}
 F(x) &= F_1(x) + F_2(x), \\
 F_1(x) &= (f_1(x), f_2(x), \dots, f_m(x)), \\
 F_2(x) &= Dx + c, \\
 f_i(x) &= x_{i-1}^2 + x_i^2 + x_{i-1}x_i + x_i x_{i+1}, \quad i = 1, 2, \dots, m \\
 x_0 &= x_{m+1} = 0,
 \end{aligned}$$

Here D is a square matrix $m \times m$ defined by condition

$$d_{ij} = \begin{cases} 4, & i = j, \\ 1, & i - j = 1, \\ -2, & i - j = -1, \\ 0, & \text{otherwise,} \end{cases}$$

and $c = (-1, -1, \dots, -1)$. The feasible set is $C = \mathbb{R}_+^m$ and the starting point is $x^0 = (0, 0, \dots, 0)$. The test results are listed in Table 5.4 for different m and ε .

Problem 5. This problem was used by Kanzow in [8] with $m = 5$. The mapping F is defined by

$$\begin{aligned}
 F(x) &= (f_1(x), \dots, f_5(x)), \\
 f_i(x) &= 2(x_i - i + 2) \exp\left\{\sum_{i=1}^5 (x_i - i + 2)^2\right\}, \quad i = 1, \dots, 5
 \end{aligned}$$

and the feasible set is $C = \mathbb{R}^5$. The solution for this problem is $x^* = (-1, 0, 1, 2, 3)$. The results of the numerical experiments for different starting points and ε are compiled in Table 5.5.

We want to note that for this problem only the number of iterations matter, since both projection and mapping F are cheap. For this reason, the time is quite small and hence, can not be a good illustration of the algorithm's performance, since Mathematica function `Timing` measures time only with some error of order 10^{-3} .

To summarize our numerical experiments, we want to make some observations. First, the disadvantage of Alg. 3.1 in comparison with other stationary algorithms is a smaller interval for possible stepsize λ . For simple problems it really can matter, however, for huge-scale problems evaluation of F is much more important. To illustrate this, let us consider Problem 1 again. The best performance of EGM and SubEGM achieves approximately for $\lambda = 0.7$ (SubPM and Alg. 3.1 do not converge for such λ). Nevertheless, the time execution of EGM and SubEGM for $m = 2000$ is 12 seconds for both, and for $m = 4000$ is 52 and 53 seconds, respectively. Hence,

x^0	ε	Alg. 4.2		Alg. Solod.-Svait.		TBFM-nst	
		iter. (np/nf)	time	iter. (np/nf)	time	iter. (np/nf)	time
(1, 1, 1, 1, 1)	10^{-3}	26 (26/26)	0.009	14 (27/53)	0.009	28 (29/57)	0.007
	10^{-6}	49 (49/49)	0.017	24 (47/93)	0.015	52 (53/105)	0.014
(0, 0, 0, 0, 0)	10^{-3}	15 (18/35)	0.009	32 (63/82)	0.015	43 (44/87)	0.012
	10^{-6}	34 (37/54)	0.016	42 (83/122)	0.025	67 (68/135)	0.017
random	10^{-3}	25 (28/41)	0.012	79 (157/175)	0.029	90 (91/181)	0.024
	10^{-6}	59 (59/59)	0.02	89 (179/215)	0.050	114 (115/229)	0.036
random	10^{-3}	20 (24/44)	0.015	23 (45/67)	0.011	36 (37/73)	0.017
	10^{-6}	43 (47/67)	0.022	33 (65/107)	0.017	60 (61/121)	0.016

TABLE 5.5

Results for Alg. 4.2 and Alg. Solod.-Svait., and TBFM-nst for Problem 5

for Problem 1 with large m EGM and SubEGM, even with their best stepsize, are slower.

Second, our numerical experiments confirm that “bad” cases (4.i) and (4.ii) in Alg 4.2 are really very rare. Notice that the number of evaluation of F is approximately the same as the number of iteration, that is a drastic contrast with Alg. Solod.Svait and TBFM-nst. Also if even we do not take into account the good performance of Alg. 4.2, it seems to be more robust than Alg. Solod.-Svait., since it does not require any additional assumptions on the feasible set and its stepsizes are separated from zero. On the other hand, the proof of Alg. Solod-Svait., in contrast to Alg. 4.2, does not require the Lipschitz-continuity of the mapping F .

Acknowledgments. The author wishes to express his gratitude to Mikhail Solodov and to the two anonymous referees for their constructive suggestions that led to improvements in the paper.

REFERENCES

- [1] A. S. ANTIPIN, *On a method for convex programs using a symmetrical modification of the Lagrange function*, *Ekonomika i Matematicheskie Metody*, 12 (1976), pp. 1164–1173.
- [2] C. BAIocchi AND A. CAPELO, *Variational and Quasivariational Inequalities. Applications to Free Boundary Problems*, Wiley, New York, 1984.
- [3] H. H. BAUSCHKE AND P. L. COMBETTES, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.
- [4] Y. CENSOR, A. GIBALI, AND S. REICH, *The subgradient extragradient method for solving variational inequalities in Hilbert space*, *Journal of Optimization Theory and Applications*, 148 (2011), pp. 318–335.
- [5] F. FACCHINEI AND J.-S. PANG, *Finite-Dimensional Variational Inequalities and Complementarity Problems, Volume I and Volume II*, Springer-Verlag, New York, USA, 2003.
- [6] P. T. HARKER AND J.-S. PANG, *A damped-newton method for the linear complementarity problem*, *Lectures in Applied Mathematics*, 26 (1990), pp. 265–284.
- [7] A. N. IUSEM AND B. F. SVAITER, *A variant of Korpelevich’s method for variational inequalities with a new search strategy*, *Optimization*, 42 (1997), pp. 309–321.
- [8] C. KANZOW, *Some equation-based methods for the nonlinear complementarity problem*, *Optimization Methods and Software*, 3 (1994), pp. 327–340.
- [9] E. N. KHOBOTOV, *Modification of the extragradient method for solving variational inequalities and certain optimization problems*, *USSR Computational Mathematics and Mathematical Physics*, 27 (1989), pp. 120–127.
- [10] I. V. KONNOV, *Equilibrium models and variational inequalities*, Elsevier, Amsterdam, 2007.
- [11] G. M. KORPELEVICH, *The extragradient method for finding saddle points and other problems*, *Ekonomika i Matematicheskie Metody*, 12 (1976), pp. 747–756.
- [12] S. I. LYASHKO, V. V. SEMENOV, AND T. A. VOITOVA, *Low-cost modification of Korpelevich’s method for monotone equilibrium problems*, *Cybernetics and Systems Analysis*, 47 (2011), pp. 631–639.
- [13] YU. V. MALITSKY AND V. V. SEMENOV, *An extragradient algorithm for monotone variational inequalities*, *Cybernetics and Systems Analysis*, 50 (2014), pp. 271–277.
- [14] P. MARCOTTE, *Application of Khobotov’s algorithm to variational inequalities and network equilibrium problems*, *Information Systems and Operational Research*, 29 (1991), pp. 258–270.

- [15] J.-S. PANG AND S. A. GABRIEL, *NE/SQP: A robust algorithm for the nonlinear complementarity problem*, *Mathematical Programming*, 60 (1993), pp. 295–337.
- [16] L. D. POPOV, *A modification of the Arrow-Hurwicz method for finding saddle points*, *Mathematical Notes*, 28 (1980), pp. 845–848.
- [17] M. V. SOLODOV AND B. F. SVAITER, *A new projection method for variational inequality problems*, *SIAM Journal on Control and Optimization*, 37 (1999), pp. 765–776.
- [18] D. SUN, *A projection and contraction method for the nonlinear complementarity problems and its extensions*, *Mathematica Numerica Sinica*, 16 (1994), pp. 183–194.
- [19] P. TSENG, *On linear convergence of iterative methods for the variational inequality problem*, *Journal of Computational and Applied Mathematics*, 60 (1995), pp. 237–252.
- [20] ———, *A modified forward-backward splitting method for maximal monotone mappings*, *SIAM Journal on Control and Optimization*, 38 (2000), pp. 431–446.
- [21] E. ZEIDLER, *Nonlinear Functional Analysis and its Applications III: Variational Methods and Optimization*, Springer-Verlag, New York, 1985.
- [22] ———, *Nonlinear Functional Analysis and its Applications II: Linear Monotone Operators, Nonlinear Monotone Operators*, Springer-Verlag, New York, 1990.