SIAM J. CONTROL AND OPTIMIZATION
Vol. 34, No. 5, pp. 1814–1830, September 1996

MODIFIED PROJECTION-TYPE METHODS FOR MONOTONE VARIATIONAL INEQUALITIES*

MICHAEL V. SOLODOV† AND PAUL TSENG‡

Abstract. We propose new methods for solving the variational inequality problem where the underlying function F is monotone. These methods may be viewed as projection-type methods in which the projection direction is modified by a strongly monotone mapping of the form $I - \alpha F$ or, if F is affine with underlying matrix M, of the form $I + \alpha M^T$, with $\alpha \in (0, \infty)$. We show that these methods are globally convergent and, if in addition a certain error bound based on the natural residual holds locally, the convergence is linear. Computational experience with the new methods is also reported.

Key words. Monotone variational inequalities, projection-type methods, error bound, linear convergence.

AMS subject classifications. 49M45, 90C25, 90C33

1. Introduction.

We consider the monotone variational inequality problem of finding an $x^* \in X$ satisfying

$$F(x^*)^T(x - x^*) \geq 0 \quad \forall x \in X, \tag{1}$$

where X is a closed convex set in $\Re^n$ and F is a monotone and continuous function from $\Re^n$ to $\Re^n$. This problem, which we abbreviate as VI(X, F), is well known in optimization (see [1, 6, 15]) and, in the special case where F is affine and X is the nonnegative orthant, reduces to the classical monotone linear complementarity problem (see [7, 36]).

Many methods have been proposed to solve VI(X, F). The simplest of these is the projection method [46] (also see [1, 2, 3, 8, 27]) which, starting with any $x \in \Re^n$, iteratively updates x according to the formula

$$x^{new} := [x - \alpha F(x)]^+,$$

where $[\cdot]^+$ denotes the orthogonal projection map onto X and α is a judiciously chosen positive stepsize. However, the projection method requires the restrictive assumption that F or F^{-1} be strongly monotone for convergence. The extragradient method [22] (also see [47, 20, 21, 31] for extensions) overcomes this difficulty by the ingenious technique of updating x according to the double projection formula:

$$x^{new} := \left[x - \alpha F\left([x - \alpha F(x)]^+\right)\right]^+.$$

This method, by virtue of its using only function evaluations and projection onto X, is easy to implement, uses little storage, and can readily exploit any sparsity or separable structure in F or in X, such as those arising in the applications considered in [3, 9, 38, 45]. Moreover, its convergence requires only that a solution exists [20],

* Received by the editors xxxx, 1994; accepted for publication (in revised form) xxxx, 1995. The work of the first author was supported by the Air Force Office of Scientific Research, Grant No. F49620-94-1-0036, and by the National Science Foundation, Grant No. CCR-9101801. The work of the second author was supported by the National Science Foundation, Grant No. CCR-9311621.

† Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706 (solodov@cs.wisc.edu).

‡ Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

while its only drawback is its, at best, linear convergence. In contrast, the methods in [4, 8, 12, 27, 32, 33, 34, 38, 40, 50, 54] require restrictive assumptions on the problem (such as F or F^{-1} be strongly monotone or F be affine; for some of the methods, it is further required that F be continuously differentiable with nonsingular Jacobian or X be bounded and polyhedral), while the matrix-splitting methods in [10, 34, 49, 51] are applicable only when F is affine (and these methods also have, at best, linear convergence). And all these methods require more computation per iteration than the extragradient method. For the special case where X is the nonnegative orthant (the monotone nonlinear complementarity problem) or a box, many other solution methods exist, but these methods tend to be ill-suited for large sparse problems and are not practically extendable to more general X. Thus, it can be said that, unless F has a special structure (F or F^{-1} is strongly monotone or F is affine) and X has a special structure (X is polyhedral or, better still, just a box), the extragradient method is a very practical method (and sometimes the only practical method) for solving VI(X, F). And, even when F is affine, there are situations where the extragradient method may be practical. As a case in point, suppose X is the Cartesian product of simplices and ellipsoids and F is affine with an underlying matrix M that is asymmetric, positive semidefinite, sparse, and having no particular structure (so M^{-1} may be dense and impractical to compute). The extragradient method can be practically implemented to solve this special case of VI(X, F) since it requires only projection onto the simplices and ellipsoids (for which many efficient methods exist [42, 53]) and multiplication of x by the sparse matrix M. In contrast, the matrix-splitting methods in [10, 34, 49, 51] require solving a nontrivial strongly monotone variational inequality problem over X at each iteration. And even on structured problems such as the discrete-time deterministic optimal control problem [45], the extragradient method may yet be practical since it is linearly convergent like the methods in [5, 10, 49, 55] while its iterations are simpler.

In this paper, we propose a new class of methods for solving VI(X, F) that are as versatile and capable of exploiting problem structure as the extragradient method and, yet, are even simpler than the latter and have a scaling feature absent in the latter. And our preliminary computational experience suggests that the new methods are practical alternatives to the extragradient method. The idea of the new methods is to choose an $n \times n$ symmetric positive definite matrix P and, starting with any $x \in \Re^n$, to iteratively update x according to the formula

$$x^{new} := x - \gamma P^{-1}\left(T_\alpha(x) - T_\alpha\left([x - \alpha F(x)]^+\right)\right), \tag{2}$$

where γ is a positive stepsize and either $T_\alpha \equiv I - \alpha F$ or, if F is affine with underlying matrix M, $T_\alpha \equiv I + \alpha M^T$, with $\alpha \in (0, \infty)$ chosen so T_α is strongly monotone. These methods are like the projection method except the projection direction $[x - \alpha F(x)]^+ - x$ is modified by T_α and P^{-1}. Like the extragradient method, these methods use two function evaluations per iteration and, as we shall show (see Theorems 2.1 and 3.2), their convergence requires only that a solution exists. Unlike the extragradient method, these methods require only one projection per iteration, rather than two, and they have an additional parameter, the scaling matrix P, that can be chosen to help accelerate the convergence (see §2 and §4 for examples and further discussions). Thus, the new methods require less work per iteration than does the extragradient method (assuming P is chosen so P^{-1} is easily computed and stored), with the savings being the greatest when the projection is expensive. Our computational experience (§4) suggests that the new methods are practical alternatives to the extragradient method, especially when F is affine or when projection onto X is expensive.

Although we will also present computational results to illustrate the practical behavior of the new methods, the focus of our paper is on laying the theoretical foundations for these methods. In particular, we will present various convergence and rate of convergence results for the new methods. Central to our rate of convergence analysis is the following growth condition on the 2-norm of the projection residual function $r : \Re^n \mapsto \Re^n$, given by

$$r(x) = x - [x - F(x)]^+,$$

near the solution set S of VI(X, F) (i.e., S comprises all $x^* \in X$ satisfying (1)): There exist positive constants μ and δ (depending on F, X only) such that

$$d(x, S) \leq \mu \|r(x)\| \quad \forall x \text{ with } \|r(x)\| \leq \delta, \tag{3}$$

where $\|\cdot\|$ denotes the 2-norm and $d(\cdot, S)$ denotes the 2-norm distance to S. (It is well known that an $x^* \in \Re^n$ solves VI(X, F) if and only if $r(x^*) = 0$.) This growth condition on $\|r(\cdot)\|$ (also called error bound) has been used in the rate of convergence analysis of various methods [25, 26, 51] and is known to hold whenever X is polyhedral and either F is affine (see [26, 43]) or F has certain strong monotonicity structure (see [51, Theorem 2]). Moreover, under additional assumptions on F, this condition holds with $\delta = \infty$ (see [23, 24, 29, 39]). Our rate of convergence analysis, similar to that in [51], entails (roughly) showing that $d(x, S)^2$ decreases by an amount in the order of $\|r(x)\|^2$ per iteration, so $\|r(x)\|$ must eventually decrease below δ, at which time (3) yields that $d(x, S)^2$ decreases at a linear rate. The analysis is also similar in spirit to those for feasible descent methods (see [25, 26, 28]) but uses $d(\cdot, S)^2$, rather than the objective function, as the merit function.

Our main results are as follows: In §2, we consider the special case of VI(X, F) where F is affine. We show that, for suitable choices of the stepsize γ, the iterates generated by (2) with $T_\alpha \equiv I + \alpha M^T$ and $\alpha = 1$ converge to a solution of VI(X, F) and, under the assumption of (3) for some μ and δ, the convergence is linear (see Algorithm 2.1 and Theorem 2.1). We then extend this method by replacing the projection direction with a more general matrix-splitting direction (see Algorithm 2.2 and Theorem 2.2). Also, we consider a modification of this method whereby one of the "$[x - F(x)]^+$" terms is replaced with $x - F(x)$ and an extra projection step is taken (see Algorithm 2.3 and Theorem 2.3). In §3, we consider the general case of VI(X, F) and we analogously analyze the convergence of iterates generated by (2) with $T_\alpha \equiv I - \alpha F$ (see Algorithms 3.1, 3.2 and Theorems 3.1, 3.2). In §4, we report our preliminary computational experience with the new methods on sparse LPs, dense monotone LCPs, and linearly constrained variational inequality problems. In §5, we give some concluding remarks.

Subsequent to the writing of this paper, we learned of the recently proposed methods of He [18, 19] which may be viewed as special cases of Algorithm 2.1 in §2, with specific choices of the scaling matrix P. He's convergence and rate of convergence results for his methods are similar to ours for Algorithm 2.1 (Theorem 2.1), although He's rate of convergence results further require X to be an orthant. In an earlier work [17], He proposed a related method which may be viewed as a version of Algorithm 2.3 in §2 with $P = I$ and X an orthant, but using a different choice of the stepsize γ_i. During the finalization of this paper, we learned of a very recently proposed method of Sun [48] which may be viewed as a version of Algorithm 3.2 in §3 with $P = I$, but using a different choice of the stepsize γ_i (see Remark 4.3 therein). The convergence analysis in [48] applies to the more general problem where F is pseudomonotone and continuous, though no convergence rate result was given.

A few words about our notation. We denote by $\Re^n$ the space of n-dimensional real column-vectors and by superscript T the transpose (of vectors and matrices). We denote by $\|\cdot\|$ the 2-norm (i.e., $\|x\| = (x^T x)^{\frac{1}{2}}$ for all vectors x) and, for any $n \times n$ symmetric positive definite matrix P, by $\|\cdot\|_P$ the 2-norm in $\Re^n$ scaled by P (i.e., $\|x\|_P = (x^T P x)^{\frac{1}{2}}$ for all $x \in \Re$) and by $P^{-1/2}$ the (unique) $n \times n$ symmetric positive definite matrix whose product with itself is P^{-1}. We denote by I either the identity matrix or the identity map and, by R-linear convergence and Q-linear convergence, we mean linear convergence in the root sense and in the quotient sense, respectively, as defined in [37].

2. Algorithms for F Affine. In this section we consider the case of VI(X, F) where F is monotone and affine, i.e.,

$$F(x) = Mx + q$$

for some $n \times n$ positive semidefinite (not necessarily symmetric) matrix M and some $q \in \Re^n$. We present and analyze three methods for solving this special case of VI(X, F). The first method is our basic method (2) with $T_\alpha \equiv I + \alpha M^T$ and, for simplicity, $\alpha = 1$. The second method is an extension of the first method in which the projection direction is replaced with a matrix-splitting direction. The third method is a modification of the first method in which the projection operation is removed from one part and added to another part of the method.

We describe the first method formally below.

ALGORITHM 2.1. *Choose any $n \times n$ symmetric positive definite matrix P and any $x^0 \in \Re^n$. Also choose a $\theta \in (0, 2)$. For $i = 0, 1, \ldots$, compute x^{i+1} from x^i according to:*

$$x^{i+1} = x^i - \gamma_i P^{-1}(I + M^T) r(x^i), \tag{4}$$

where

$$\gamma_i = \theta \|P^{-1/2}(I + M^T) r(x^i)\|^{-2} \|r(x^i)\|^2. \tag{5}$$

The parameters P and θ are key to the performance of Algorithm 2.1. We can choose P so that P^{-1} is easily computed and stored (e.g., $P = I$) or that $\|P^{-1/2}(I + M^T)\|$ is small (e.g., $P = (I + M^T)(I + M)$) so γ_i is large. Below we show that this simple method is convergent and, when the error bound (3) holds, is linearly convergent. The proof is based on showing that $(I + M^T) r(x)$ makes an acute angle with $x - x^*$ for any solution x^*, so the distance from x to the solution set S, measured in the scaled 2-norm $\|\cdot\|_P$, decreases when x is moved opposite the direction $P^{-1}(I + M^T) r(x)$.

THEOREM 2.1. *Assume that $F(x) = Mx + q$ for some $n \times n$ positive semidefinite matrix M and some $q \in \Re^n$, and that the solution set S of VI(X, F) is nonempty. Then any sequence $\{x^i\}$ generated by Algorithm 2.1 converges to an element of S and, if (3) holds for some μ and δ, the convergence is R-linear.*

Proof. Let x^* be any element of S. For each $i \in \{0, 1, \ldots\}$, we have from (4) that

$$\begin{aligned} & \|x^{i+1} - x^*\|_P^2 \\ = \; & \|x^i - x^* - \gamma_i P^{-1}(I + M^T) r(x^i)\|_P^2 \\ = \; & \|x^i - x^*\|_P^2 - 2\gamma_i (x^i - x^*)^T (I + M^T) r(x^i) + \gamma_i^2 \|P^{-1/2}(I + M^T) r(x^i)\|^2. \end{aligned} \tag{6}$$

We bound below the next to last term in (6). Let $z^i = [x^i - Mx^i - q]^+$ (so $r(x^i) = x^i - z^i$). By properties of the projection operator, we have

$$0 \le (y - z^i)^T (Mx^i + q + z^i - x^i) \quad \forall y \in X.$$

Similarly, since x^* is a solution of VI(X, F), we have

$$0 \le (y - x^*)^T (Mx^* + q) \quad \forall y \in X.$$

Taking $y = x^*$ in the first inequality and taking $y = z^i$ in the second inequality and then adding the two resulting inequalities yields

$$\begin{aligned} 0 &\le (x^* - z^i)^T (M(x^i - x^*) + z^i - x^i) \\ &= (x^* - x^i)^T M(x^i - x^*) + (x^i - x^*)^T (I + M^T)(x^i - z^i) - \|x^i - z^i\|^2 \\ &\le (x^i - x^*)^T (I + M^T)(x^i - z^i) - \|x^i - z^i\|^2 \\ &= (x^i - x^*)^T (I + M^T) r(x^i) - \|r(x^i)\|^2, \end{aligned}$$

where the second inequality follows from the positive semidefinite property of M. Using this to bound the next to last term in (6) yields the key relation

$$\begin{aligned} \|x^{i+1} - x^*\|_P^2 &\le \|x^i - x^*\|_P^2 - 2\gamma_i \|r(x^i)\|^2 + \gamma_i^2 \|P^{-1/2}(I + M^T) r(x^i)\|^2 \\ &= \|x^i - x^*\|_P^2 - \theta(2-\theta)\|P^{-1/2}(I + M^T) r(x^i)\|^{-2} \|r(x^i)\|^4 \qquad (7) \\ &\le \|x^i - x^*\|_P^2 - \theta(2-\theta)\|P^{-1/2}(I + M^T)\|^{-2} \|r(x^i)\|^2, \qquad (8) \end{aligned}$$

where the equality follows from (5). The remaining argument is patterned after the proof of [44, Theorem 1] and of [51, Theorem 1].

Since (8) holds for all i, it follows that $\|x^i - x^*\|_P$ is nonincreasing with i and that $\|r(x^i)\| \to 0$ as $i \to \infty$. This shows that $\{x^i\}$ is bounded and, by continuity of $r(\cdot)$, each cluster point x^∞ satisfies $r(x^\infty) = 0$ and hence is in S. Then, we can choose x^* in (8) to be x^∞ and conclude that $\|x^i - x^\infty\|_P \to 0$ as $i \to \infty$, i.e., $\{x^i\}$ converges to x^∞.

Assume that (3) holds for some μ and δ. Let $\psi(x) = \min_{x^* \in S} \|x - x^*\|_P^2$ (so $\psi(x) \le \|P\| d(x, S)^2$). Since (7) holds for all i and all $x^* \in S$, by choosing (for each i) x^* to be the element of S closest to x^i in the norm $\|\cdot\|_P$, we obtain for all i,

$$\begin{aligned} \psi(x^{i+1}) &\le \|x^{i+1} - x^*\|_P^2 \\ &\le \|x^i - x^*\|_P^2 - \theta(2-\theta)\|P^{-1/2}(I + M^T) r(x^i)\|^{-2} \|r(x^i)\|^4 \\ &= \psi(x^i) - \theta(2-\theta)\|P^{-1/2}(I + M^T) r(x^i)\|^{-2} \|r(x^i)\|^4 \qquad (9) \\ &\le \psi(x^i) - \eta \|r(x^i)\|^2, \qquad (10) \end{aligned}$$

where we let $\eta = \theta(2-\theta)\|P^{-1/2}(I + M^T)\|^{-2}$. Since $\|r(x^i)\| \to 0$, we have $\|r(x^i)\| \le \delta$ for all i greater than some $\bar{i}$, in which case (3) yields $d(x^i, S) \le \mu \|r(x^i)\|$. Using this to bound the righthand side of the above inequality yields

$$\psi(x^{i+1}) \le \psi(x^i) - \frac{\eta}{\mu^2} d(x^i, S)^2 \le \psi(x^i) - \frac{\eta}{\mu^2 \|P\|} \psi(x^i)$$

for all $i > \bar{i}$, so $\{\psi(x^i)\}$ converges Q-linearly to zero and, by (10), $\{r(x^i)\}$ converges R-linearly to zero. Since by (4), (5) and (9) we have

$$\|x^{i+1} - x^i\|_P = \theta \|P^{-1/2}(I + M^T) r(x^i)\|^{-1} \|r(x^i)\|^2 \le \theta^{1/2}(2-\theta)^{-1/2}(\psi(x^i) - \psi(x^{i+1}))^{1/2}$$

for all i, it follows from $\{\psi(x^i)\}$ converging Q-linearly to zero that $\{\|x^{i+1} - x^i\|_P\}$ converges R-linearly to zero and hence $\{x^i\}$ converges R-linearly. $\Box$

The above proof shows that we can alternatively choose $\gamma_i = \bar{\gamma}$ for all i in Algorithm 2.1, where $\bar{\gamma}$ is any scalar satisfying

$$0 < \bar{\gamma} < 2\|P^{-1/2}(I + M^T)\|^{-2}.$$

However, this constant stepsize choice is impractical since it is conservative and difficult to compute.

Algorithm 2.1 can be further extended by replacing the projection term $[x - (Mx + q)]^+$ in the definition of $r(x)$ with a more general matrix splitting term. In particular, consider the following method:

ALGORITHM 2.2. *Choose any $n \times n$ symmetric positive definite matrix P and any $x^0 \in \Re^n$. Also choose an $n \times n$ positive definite matrix B and a $\theta \in (0, 2)$. For $i = 0, 1, \ldots$, compute x^{i+1} from x^i according to:*

$$x^{i+1} = x^i - \gamma_i P^{-1}(B + M^T)(x^i - z^i), \tag{11}$$

where z^i is the unique solution of the nonlinear equations

$$z^i = [z^i - (B(z^i - x^i) + Mx^i + q)]^+, \tag{12}$$

and γ_i is given by

$$\gamma_i = \theta\|P^{-1/2}(B + M^T)(x^i - z^i)\|^{-2}(x^i - z^i)^T B(x^i - z^i). \tag{13}$$

Notice if we choose $B = I$, then Algorithm 2.2 reduces to Algorithm 2.1. In general, we should choose B to be close to M (so that z^i is close to S for fast convergence) and yet to have enough structure (e.g., lower/upper triangular or tridiagonal or block diagonal) so that z^i is easily computable. We have the following result whose proof is similar to that of Theorem 2.1 and thus is omitted.

THEOREM 2.2. *Assume that $F(x) = Mx + q$ for some $n \times n$ positive semidefinite matrix M and some $q \in \Re^n$, and that the solution set S of VI(X, F) is nonempty. Then any sequence $\{x^i\}$ generated by Algorithm 2.2 converges to an element of S and, if (3) holds for some μ and δ, the convergence is R-linear.*

We note that Algorithm 2.2 is closely related to the following iterative method proposed in [11]

$$x^{i+1} = \arg\min_{x \in X} \left\{ \psi_i(x) := (x - x^i)^T(Mx + q) + \frac{\nu}{2}\|x - x^i\|^2 \right\}, \tag{14}$$

where ν is a positive scalar. For the specific choice of $B = M + M^T + \nu I$, we have from (12) that

$$\begin{aligned} z^i &= [z^i - (B(z^i - x^i) + Mx^i + q)]^+ \\ &= [z^i - \big((M + M^T + \nu I)(z^i - x^i) + Mx^i + q\big)]^+ \\ &= [z^i - \big((M + M^T)z^i - M^T x^i + q + \nu(z^i - x^i)\big)]^+ \\ &= [z^i - \nabla\psi_i(z^i)]^+, \end{aligned}$$

so that

$$z^i = \arg\min_{x \in X} \psi_i(x).$$

Thus (12) generalizes (14). We note that in [11] no convergence result is given for (14). Theorem 2.2 shows that if the step (11) is added, the resulting method (11)-(13) converges to a solution of VI(X, F) and, if (3) holds (as in the case where X is also polyhedral), the convergence is R-linear.

Additional modifications of the preceding methods are possible. For example, we can pass each iterate through a nearest-point projection (with respect to the norm $\|\cdot\|_P$) on to X. For Algorithm 2.1, this modification would entail replacing (4) with

$$x^{i+1} = [x^i - \gamma_i P^{-1}(I + M^T)r(x^i)]_P^+,$$

where $[y]_P^+$ denotes the point in X whose distance to y (measured in the norm $\|\cdot\|_P$) is minimal. To see that this does not affect the convergence (and, in fact, accelerates convergence) of the methods, we use the following fact about nearest-point projection:

$$\|[y]_P^+ - x^*\|_P^2 \le \|y - x^*\|_P^2 - \|y - [y]_P^+\|_P^2 \tag{15}$$

for all $y \in \Re^n$ and all $x^* \in X$ (see, e.g., [31, Appendix]).

We next present a modification, rather than an extension, of Algorithm 2.1, in which we expand out $(I + M^T)r(x^i) = x^i - [x^i - (Mx^i + q)]^+ + M^T r(x^i)$ and replace the "$[x^i - (Mx^i + q)]^+$" term with $x^i - (Mx^i + q)$. In contrast to Algorithm 2.1, an extra projection on to X is needed.

Algorithm 2.3. *Choose any $n \times n$ symmetric positive definite matrix P and any $x^0 \in X$. Also choose a $\theta \in (0, 2)$. For $i = 0, 1, \ldots$, compute x^{i+1} from x^i according to:*

$$x^{i+1} = \left[x^i - \gamma_i P^{-1}(Mx^i + q + M^T r(x^i))\right]_P^+, \tag{16}$$

where γ_i is given by

$$\gamma_i = \theta \|P^{-1/2}(Mx^i + q + M^T r(x^i))\|^{-2} \|r(x^i)\|^2. \tag{17}$$

The convergence properties of Algorithm 2.3 are stated in the following theorem, whose proof is similar to that of Theorem 2.1.

Theorem 2.3. *Assume that $F(x) = Mx + q$ for some $n \times n$ positive semidefinite matrix M and some $q \in \Re^n$, and that the solution set S of VI(X, F) is nonempty. Then any sequence $\{x^i\}$ generated by Algorithm 2.3 converges to an element of S.*

Proof. Let x^* be any element of S. For each $i \in \{0, 1, \ldots\}$, we have from (16) and (15) (with $y = x^i - \gamma_i P^{-1}(Mx^i + q + M^T r(x^i))$) that

$$\begin{aligned} \|x^{i+1} - x^*\|_P^2 &\le \|x^i - x^* - \gamma_i P^{-1}(Mx^i + q + M^T r(x^i))\|_P^2 \\ &= \|x^i - x^*\|_P^2 - 2\gamma_i (x^i - x^*)^T (Mx^i + q + M^T r(x^i)) \\ &+ \gamma_i^2 \|P^{-1/2}(Mx^i + q + M^T r(x^i))\|^2. \end{aligned} \tag{18}$$

We bound below the next to last term in (18). Let $z^i = [x^i - Mx^i - q]^+$ (so $r(x^i) = x^i - z^i$). By properties of the projection operator, we have

$$0 \le (y - z^i)^T (Mx^i + q + z^i - x^i) \quad \forall y \in X.$$

Similarly, since x^* is a solution of VI(X, F), we have

$$0 \le (y - x^*)^T (Mx^* + q) \quad \forall y \in X.$$

Taking $y = x^i$ in the first inequality and taking $y = z^i$ in the second inequality and then adding the two resulting inequalities yields

$$\begin{aligned} 0 &\le (x^i - z^i)^T(Mx^i + q + z^i - x^i) + (z^i - x^*)^T(Mx^* + q) \\ &= (x^i - x^*)^T(Mx^i + q + M^T(x^i - z^i)) + (x^i - x^*)^T M(x^* - x^i) - \|x^i - z^i\|^2 \\ &\le (x^i - x^*)^T(Mx^i + q + M^T(x^i - z^i)) - \|x^i - z^i\|^2 \\ &= (x^i - x^*)^T(Mx^i + q + M^T r(x^i)) - \|r(x^i)\|^2, \end{aligned}$$

where the second inequality follows from the positive semidefinite property of M. Using this to bound the next to last term in (6) yields

$$\begin{aligned} \|x^{i+1} - x^*\|_P^2 &\le \|x^i - x^*\|_P^2 - 2\gamma_i\|r(x^i)\|^2 + \gamma_i^2\|P^{-1/2}(Mx^i + q + M^T r(x^i))\|^2 \\ &= \|x^i - x^*\|_P^2 - \theta(2-\theta)\|P^{-1/2}(Mx^i + q + M^T r(x^i))\|^{-2}\|r(x^i)\|^4, \end{aligned}$$

where the equality follows from (17). The remainder of the proof is similar to that of Theorem 2.1, but using the above relation instead of (8). □

Notice that Algorithm 2.3 requires two projections per iteration, the same as the extragradient method. However, unlike the extragradient method, Algorithm 2.3 does not appear to have linear convergence, even if (3) holds for some μ and δ.[1]

3. Algorithms for F Non-Affine. In this section we consider the general case of VI(X, F) where F is monotone and continuous. We present and analyze two versions of our basic method (2) with $T_\alpha \equiv I - \alpha F$ and with α chosen so T_α is strongly monotone. The first version, which uses a fixed α, is simpler but requires F furthermore to be Lipschitz continuous. The second version, which chooses α dynamically, is more intricate but is more practical and solves the general problem.

We describe the first method formally below. For this method to be applicable, we require F furthermore to be Lipschitz continuous.

ALGORITHM 3.1. *Choose any $n \times n$ symmetric positive definite matrix P and any $x^0 \in \Re^n$. Also choose any $\theta \in (0, 2)$ and any $\alpha \in (0, 1/\lambda)$, where λ is a constant satisfying*

$$(x - z)^T(F(x) - F(z)) \le \lambda\|x - z\|^2 \quad \forall x, z \in \Re^n. \tag{19}$$

(We can, for example, take λ to be the Lipschitz constant of F.) For $i = 0, 1, \ldots$, compute x^{i+1} from x^i according to:

$$x^{i+1} = x^i - \gamma_i P^{-1}(x^i - z^i - \alpha F(x^i) + \alpha F(z^i)), \tag{20}$$

where z^i and γ_i are given by, respectively,

$$z^i = [x^i - \alpha F(x^i)]^+, \tag{21}$$

$$\gamma_i = \theta(1 - \alpha\lambda)\|P^{-1/2}(x^i - z^i - \alpha F(x^i) + \alpha F(z^i))\|^{-2}\|x^i - z^i\|^2. \tag{22}$$

Algorithm 3.1 requires less computation per iteration than the extragradient method (in particular, it avoids performing an extra projection step). Also, unlike

[1] Subsequent to the writing of the original paper, we were informed by B. He that, for the method he proposed in [17] (which may be viewed as Algorithm 2.3 with $P = I$ and X an orthant, but using a different choice of γ_i), he could show a linear convergence result analogous to that shown in [18] (see [19, Eq. (8)]).

the extragradient method, Algorithm 3.1 allows scaling of direction by P^{-1} without having to accordingly scale the norm with respect to which projection is taken. In the case where F is affine, i.e., $F(x) = Mx + q$ for some $n \times n$ positive semidefinite matrix M and some $q \in \Re^n$, the formula (20) reduces to

$$x^{i+1} = x^i - \gamma_i P^{-1}(I - \alpha M)(x^i - z^i),$$

which is reminiscent of (4). If in addition M is skew symmetric (i.e., $M^T = -M$) so that (19) holds with $\lambda = 0$, we can choose α arbitrarily large and can reasonably choose P to be $P = (I - \alpha M)(I - \alpha M^T)$. In fact, for the choice of $\alpha = 1$ (and using $M^T = -M$), the formula (20) reduces precisely to (4).

We show in the following theorem that Algorithm 3.1 has convergence properties similar to that of Algorithm 2.1. The proof of this theorem is patterned after that of Theorem 2.1.

THEOREM 3.1. *Assume that F is monotone and Lipschitz continuous and that the solution set S of VI(X, F) is nonempty. Then any sequence $\{x^i\}$ generated by Algorithm 3.1 converges to an element of S and, if (3) holds for some μ and δ, the convergence is R-linear.*

Proof. Let x^* be any element of S. For each $i \in \{0, 1, ...\}$, we have from (20) that

$$\begin{aligned} \|x^{i+1} - x^*\|_P^2 &= \|x^i - x^* - \gamma_i P^{-1}(x^i - z^i - \alpha F(x^i) + \alpha F(z^i))\|_P^2 \\ &= \|x^i - x^*\|_P^2 - 2\gamma_i (x^i - x^*)^T (x^i - z^i - \alpha F(x^i) + \alpha F(z^i)) \\ &+ \gamma_i^2 \|P^{-1/2}(x^i - z^i - \alpha F(x^i) + \alpha F(z^i))\|^2. \end{aligned} \tag{23}$$

We bound below the next to last term in (23). By (21) and properties of the projection operator, we have

$$0 \le (y - z^i)^T (\alpha F(x^i) + z^i - x^i) \quad \forall y \in X.$$

Similarly, since x^* is a solution of VI(X, F), we have

$$0 \le (y - x^*)^T F(x^*) \quad \forall y \in X.$$

Taking $y = x^*$ in the first inequality and taking $y = z^i$ in the second inequality and then adding the two resulting inequalities yields

$$\begin{aligned} 0 &\le (x^* - z^i)^T (\alpha F(x^i) + z^i - x^i) + \alpha (z^i - x^*)^T F(x^*) \\ &= \alpha (x^* - z^i)^T (F(z^i) - F(x^*)) + (x^* - x^i)^T (\alpha F(x^i) - \alpha F(z^i) + z^i - x^i) \\ &+ \alpha (x^i - z^i)^T (F(x^i) - F(z^i)) - \|x^i - z^i\|^2 \\ &\le (x^* - x^i)^T (\alpha F(x^i) - \alpha F(z^i) + z^i - x^i) + \alpha (x^i - z^i)^T (F(x^i) - F(z^i)) - \|x^i - z^i\|^2 \\ &\le (x^* - x^i)^T (\alpha F(x^i) - \alpha F(z^i) + z^i - x^i) - (1 - \alpha\lambda)\|x^i - z^i\|^2, \end{aligned}$$

where the second inequality follows from the monotone property of F and the last inequality follows from (19). Using this to bound the next to last term in (23) yields

$$\begin{aligned} &\|x^{i+1} - x^*\|_P^2 \\ \le\ &\|x^i - x^*\|_P^2 - 2\gamma_i (1 - \alpha\lambda)\|x^i - z^i\|^2 + \gamma_i^2 \|P^{-1/2}(x^i - z^i - \alpha F(x^i) + \alpha F(z^i))\|^2 \\ =\ &\|x^i - x^*\|_P^2 - \theta(2 - \theta)(1 - \alpha\lambda)^2 \|P^{-1/2}(x^i - z^i - \alpha F(x^i) + \alpha F(z^i))\|^{-2} \|x^i - z^i\|^4, \end{aligned} \tag{24}$$

where the equality follows from (22).

The remainder of the proof is similar to that of Theorem 2.1, but using (24) instead of (8). For the R-linear convergence result, we also use the observations (see (21) and (22)) that

$$\begin{aligned}\|x^{i+1} - x^i\|_P &= \theta(1-\alpha\lambda)\|P^{-1/2}(x^i - z^i - \alpha F(x^i) + \alpha F(z^i))\|^{-1}\|x^i - z^i\|^2 \\ &\geq \theta(1-\alpha\lambda)\|P^{-1/2}\|^{-1}(1+\alpha L)^{-1}\|x^i - z^i\| \\ &\geq \theta(1-\alpha\lambda)\|P^{-1/2}\|^{-1}(1+\alpha L)^{-1}\min\{1,\alpha\}\|r(x^i)\|\end{aligned}$$

for all i, where L denotes the Lipschitz constant of F and the last inequality follows from [13, Lemma 1]. Thus, the rightmost term in (24) is bounded above by a positive constant times $-\|r(x^i)\|^2$ and, whenever this term converges R-linearly to zero as $i \to \infty$, so does $\|x^{i+1} - x^i\|_P^2$; hence $\{x^i\}$ converges R-linearly. □

Algorithm 3.1 is a conceptual method since in practice F need not be Lipschitz continuous or the constant λ may be difficult to estimate or a stepsize of less than $1/\lambda$ may be too conservative. Below we present a practical version of Algorithm 3.1 that chooses α dynamically according to a novel Armijo-Goldstein-type rule. This practical version has all the convergence properties of Algorithm 3.1 and requires F to be only monotone and continuous for convergence (see Theorem 3.2).

ALGORITHM 3.2. *Choose any $n \times n$ symmetric positive definite matrix P and any $x^0 \in \Re^n$ and $\alpha_{-1} \in (0,\infty)$. Also choose any $\theta \in (0,2)$, $\rho \in (0,1)$, and $\beta \in (0,1)$. For $i = 0, 1, \ldots$, compute (x^{i+1}, α_i) from (x^i, α_{i-1}) according to: Choose α_i to be the largest $\alpha \in \{\alpha_{i-1}, \alpha_{i-1}\beta, \alpha_{i-1}\beta^2, \ldots\}$ satisfying*

$$\alpha(x^i - z^i(\alpha))^T(F(x^i) - F(z^i(\alpha))) \leq (1-\rho)\|x^i - z^i(\alpha)\|^2, \tag{25}$$

and let

$$x^{i+1} = x^i - \gamma_i P^{-1}(x^i - z^i(\alpha_i) - \alpha_i F(x^i) + \alpha_i F(z^i(\alpha_i))), \tag{26}$$

where $z^i(\alpha)$ and γ_i are given by, respectively,

$$z^i(\alpha) = [x^i - \alpha F(x^i)]^+ \quad \forall \alpha \in (0,\infty), \tag{27}$$

$$\gamma_i = \theta\rho\|P^{-1/2}(x^i - z^i(\alpha_i) - \alpha_i F(x^i) + \alpha_i F(z^i(\alpha_i)))\|^{-2}\|x^i - z^i(\alpha_i)\|^2. \tag{28}$$

The motivation for taking trial values of α starting at α_{i-1} comes from our empirical experience that, for $i > 0$, $\alpha = \alpha_{i-1}$ either satisfies or comes close to satisfying (25), so in general only a few trial values of α are needed to find α_i. The condition (25) may be viewed as a local approximation to the condition $\alpha < 1/\lambda$ used in Algorithm 3.1. (If we let $\lambda_i = (x^i - z^i(\alpha))^T(F(x^i) - F(z^i(\alpha)))/\|x^i - z^i(\alpha)\|^2$, then (25) reduces to $\alpha \leq (1-\rho)/\lambda_i$.) We had also considered choosing α_i to be the largest $\alpha \in \{\sigma, \sigma\beta, \sigma\beta^2, \ldots\}$ satisfying (25), where $\sigma \in (0,\infty)$. It can be checked that the convergence results below still hold for this alternative stepsize rule, but this rule is not as practical since it typically needs many more trial values of α to find α_i. Lastly, as with Algorithm 2.1 for the affine case, we can pass each iterate generated by Algorithm 3.2 through a nearest-point projection (with respect to the norm $\|\cdot\|_P$) on to X and the convergence results below would still hold.

Below we present the convergence results for Algorithm 3.2. The proof is patterned after that for Theorem 3.1 and, for simplicity, we supply only the key steps.

THEOREM 3.2. *Assume that F is monotone and continuous and that the solution set S of VI(X,F) is nonempty. Then any sequence $\{x^i\}$ generated by Algorithm 3.2 converges to an element of S and, if (3) holds for some μ and δ and F is Lipschitz continuous on $S+\epsilon B$ for some $\epsilon>0$ (where $B=\{x \mid \|x\|\leq 1\}$), the convergence is R-linear.*

Proof. First, we claim that, for each i, (25) holds for all α sufficiently small, so α_i is well defined. To see this, note that $z^i(\alpha)\to[x^i]^+$ as $\alpha\to 0$, so if $x^i\notin X$, then the righthand side of (25) would tend to a positive limit while the lefthand side of (25) would tend to zero as $\alpha\to 0$, implying the claim. If $x^i\in X$ (so $x^i=[x^i]^+$), then since F is continuous and $z^i(\alpha)\to[x^i]^+=x^i$ as $\alpha\to 0$, we have

$$\|F(x^i)\|\|F(x^i)-F(z^i(\alpha))\|\leq(1-\rho)\|r(x^i)\|^2$$

for all $\alpha\in(0,1]$ sufficiently small. For any such α, we have

$$\begin{aligned}
\alpha(x^i-z^i(\alpha))^T(F(x^i)-F(z^i(\alpha))) &= \alpha([x^i]^+-[x^i-\alpha F(x^i)]^+)^T(F(x^i)-F(z^i(\alpha)))\\
&\leq \alpha^2\|F(x^i)\|\|F(x^i)-F(z^i(\alpha))\|\\
&\leq \alpha^2(1-\rho)\|r(x^i)\|^2\\
&\leq (1-\rho)\|x^i-z^i(\alpha)\|^2,
\end{aligned}$$

where the first inequality uses the Cauchy-Schwartz inequality and the nonexpansive property of $[\cdot]^+$; the last inequality uses $\alpha\in(0,1]$ and [13, Lemma 1]. Thus, the claim holds.

To show that $\{x^i\}$ converges to an element of S, let x^* be any element of S. For each $i\in\{0,1,...\}$, we have by an argument analogous to the proof of Theorem 3.1, but with (19)–(22) replaced by (25)–(28) (and taking $\alpha=\alpha_i$ in (25)), that (cf. (24))

$$\begin{aligned}
\|x^{i+1}-x^*\|_P^2 &\leq \|x^i-x^*\|_P^2-\theta(2-\theta)\rho^2\|x^i-z^i(\alpha_i)\|^4\\
&\quad\cdot \|P^{-1/2}(x^i-z^i(\alpha_i)-\alpha_i F(x^i)+\alpha_i F(z^i(\alpha_i)))\|^{-2}.
\end{aligned}\tag{29}$$

Thus $\{x^i\}$ is bounded and $\{\|x^i-z^i(\alpha_i)\|\}\to 0$. Also, $\{\alpha_i\}$ is nonincreasing, so it has a limit α_∞. We claim that $\{x^i\}$ has at least one cluster point in S. In the case where $\alpha_\infty>0$, this follows from $\{\|x^i-z^i(\alpha_i)\|\}\to 0$ and the the continuity of F and the projection operator, which imply that every cluster point x^∞ of $\{x^i\}$ satisfies

$$x^\infty=[x^\infty-\alpha_\infty F(x^\infty)]^+,$$

and hence is in S. In the case where $\alpha_\infty=0$, we argue by contradiction by supposing that every cluster point of $\{x^i\}$ is not in S. Since $\alpha_\infty=0$, there must exist a subsequence K of $\{0,1,...,\}$ satisfying $\alpha_i<\alpha_{i-1}$ for all $i\in K$, and, by passing to a subsequence if necessary, we can assume that $\{x^i\}_{i\in K}$ converges to some $x^\infty\notin S$. Since $x^\infty\notin S$, it follows from the continuity of F and our earlier argument showing that (25) holds for all α sufficiently small that, for all $i\in K$ sufficiently large (so that x^i is near x^∞ and α_{i-1} is sufficiently small), $\alpha=\alpha_{i-1}$ satisfies (25). This implies we would choose $\alpha_i=\alpha_{i-1}$ for all $i\in K$ sufficient large, contradicting our hypothesis on K. Thus, $\{x^i\}$ has at least one cluster point, say x^∞, that is in S. Letting $x^*=x^\infty$ in (29), we obtain that the sequence $\{\|x^i-x^\infty\|\}$ is nonincreasing. Since this sequence has a subsequence converging to zero, the entire sequence must converge to zero.

In the case where (3) holds for some μ and δ and F is Lipschitz continuous (with constant L) on $S+\epsilon B$ for some $\epsilon>0$, we note that since $\{x^i\}$ converges to an element of S, we have x^i and $z^i(\alpha)$ inside $S+\epsilon B$ for all $\alpha \in (0, \alpha_{-1}]$ and all i exceeding some $\bar{i}$. For all such i, (25) holds for all $\alpha \in (0,(1-\rho)/L)$, so our choice of α_i implies $\alpha_i \geq \min\{\alpha_{\bar{i}}, \beta(1-\rho)/L\}$. Thus, $\{\alpha_i\}$ is bounded away from zero. The R-linear convergence of $\{x^i\}$ then follows from an argument analogous to the the proof of Theorem 3.1. □

4. Computational Experience. To better understand the behavior of the new methods in practice, we implemented Algorithm 2.1 in Fortran to solve sparse LPs and dense monotone LCPs, and implemented Algorithms 2.1 and 3.2 in Matlab to solve linearly constrained variational inequality problems (using the quadratic-program solver qp.m from the Matlab optimization toolbox to perform the projection). For benchmark, we compared the performance of these implementations with analogous implementations of the extragradient method as described in [31]. (We have included LPs and dense monotone LCPs in our tests not because they are problems for which the new methods are designed to solve, but because these problems are well known special cases of VI(X,F) and tests on them give us a better overall understanding of the new methods.) Though our results are preliminary, they suggest that the new methods are practical alternatives to the extragradient method, especially when F is affine or when projection onto X is expensive. We describe the test details below.

All Fortran codes were compiled by the DEC Fortran-77 compiler Version 4.2 using the default optimization option and were run on a Decstation 5000 under the operating system Ultrix Version 4.2A. All Matlab codes were ran on the same Decstation 5000 under Matlab version 4.2a.

Our first set of tests was conducted on sparse LP of the form: $\min\{\ c^Ty \mid Ay = b,\ y \geq 0\ \}$, where A is an $m \times l$ matrix, $b \in \Re^m$, and $c \in \Re^l$. We reformulated the LP as a VI(X,F) with

$$X = \Re^l_+ \times \{0\}^m, \quad F(x) = Mx + q, \quad M = \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix}, \quad q = \begin{bmatrix} c \\ -b \end{bmatrix}.$$

Then we applied Algorithm 2.1 and the extragradient method to this VI(X,F). The first six test problems were randomly generated, with the entries of c uniformly generated from $[1,100]$, with the number of nonzeros per column of A fixed at 5% and the nonzeros uniformly generated from $[-5,5]$, and with $b = A\bar{x}$, where $\bar{x} = (10/l, ..., 10/l)$. The seventh to ninth test problems were taken from the Netlib library (see [14]). The performance of Algorithm 2.1 is sensitive to the choice of P and θ and, in our implementation of Algorithm 2.1, we chose P to be the diagonal part of $(I+M^T)(I+M)$ (which made P^{-1} easy to compute and still yielded fast convergence) and chose $\theta = .7$ (which yielded much faster convergence than with $\theta = 1$). The parameters in the extragradient method were similarly tuned to optimize the method's performance. The test results are summarized in Table 1 below. In general, Algorithm 2.1 required fewer iterations and less time than the extragradient method, with the improvement most pronounced when $l \leq 2m$. However, both methods did very poorly on the Netlib problems, which suggests that these methods are not well suited for solving small to medium-sized LP. For large-sized LP, these methods may yet be practical since they have low storage requirement and can exploit sparsity structure in the problem.

Our second set of tests was conducted on dense monotone LCP, corresponding to VI(X,F) with

$$X = \Re^n_+, \qquad F(x) = Mx + q,$$

Problem			Algorithm 2.1[1]				Extragradient[2]			
			($\epsilon = 10^{-2}$)		($\epsilon = 10^{-3}$)		($\epsilon = 10^{-2}$)		($\epsilon = 10^{-3}$)	
Name	m	l	**iter.**[3]	**CPU**[4]	**iter.**[3]	**CPU**[4]	**iter.**[3]	**CPU**[4]	**iter.**[3]	**CPU**[4]
RanLP1	100	200	738	2.6	2776	12.3	1009	5.0	5056	31.5
RanLP2	100	300	599	4.0	2811	15.1	867	6.3	8380	65.2
RanLP3	100	400	697	5.1	3326	25.0	762	8.0	3058	31.8
RanLP4	200	400	790	22.6	3174	81.9	759	28.4	3005	107.6
RanLP5	200	600	691	14.7	2301	85.0	748	23.4	2980	82.8
RanLP6	200	800	875	25.3	3215	97.3	861	38.0	4496	177.2
Adlittle	56	138	56219	123.9	73804	163.9	–[5]	–	–	–
Scorpion	388	466	1609	13.01	6058	56.8	3372	36.7	14277	159.3
Bandm	305	472	1202607	12837	–	–	–	–	–	–
hline										

TABLE 1
Results for Algorithm 2.1 and extragradient method on LP.

[1] Algorithm 2.1 with P being the diagonal part of $(I + M^T)(I + M)$ and $\theta = .7$.
[2] The extragradient method as described in [31], with $\beta = .7$ and initial $\alpha = 1$.
[3] For all methods, $x^0 = 0$ and the termination criterion is $\|r(x)\| \le \epsilon$.
[4] Time (in seconds) obtained using the intrinsic function SECNDS and with the codes compiled by the DEC Fortran-77 compiler and ran on a Decstation 5000; does not include time to read problem data.
[5] $\|r(x)\| \approx 2 \cdot 10^{-2}$ after 50955000 iterations.

for some $n \times n$ positive semidefinite matrix M and some $q \in \Re^n$. The first three (respectively, fourth to sixth) test problems were randomly generated with

$$M = \omega E E^T + E - E^T,$$

where $\omega = 0$ (respectively, $\omega = 1$) and every entry of the $n \times n$ matrix E was uniformly generated from $[-5, 5]$, and with $q = -M\bar{x} + \bar{y}$, where each entry of $\bar{x}$ has equal probability of being 0 or being uniformly generated from $[5, 10]$ and each entry of $\bar{y}$ is 0 if the corresponding entry of $\bar{x}$ is 0 and otherwise has equal probability of being 0 or being uniformly generated from $[5, 10]$ (so $\bar{x}$ is a solution). The seventh to ninth test problems were deterministically generated with

$$M = E E^T,$$

where the (i, j)th entry of the $n \times n$ matrix E is $5(i - j)/n$ for all i and j, and with $q = -M\bar{x} + \bar{y}$, where the first $n/2$ entries of $\bar{x}$ are 0 and the rest are 7.5 and the first $n/4$ entries of $\bar{y}$ are 5 and the rest are 0 (so $\bar{x}$ is a solution). The remaining test problems were borrowed from [16, §5]. In particular, the tenth (respectively, eleventh) test problem was randomly generated with

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

where every entry of the $n \times n$ matrix A and of the $n \times n$ skew-symmetric matrix B is uniformly generated from $(-5, 5)$ and every diagonal entry of the $n \times n$ diagonal B is uniformly generated from $(0, 0.3)$ (so M is positive definite), and with every entry of q uniformly generated from $(-500, 500)$ (respectively, $(-500, 0)$). The twelveth test problem is one for which Lemke's method is known to run in exponential time, with the (i, j)th entry of M equal to 2 (respectively, 1 and 0) if $j > i$ (respectively, $j = i$ and $j < i$) for all i and j (so M is positive semidefinite), and with every entry of q equal

Problem		Algorithm 2.1[1]				Extragradient[2]			
		($\epsilon = 10^{-2}$)		($\epsilon = 10^{-3}$)		($\epsilon = 10^{-2}$)		($\epsilon = 10^{-3}$)	
Name	n	**iter.**[3]	**CPU**[4]	**iter.**[3]	**CPU**[4]	**iter.**[3]	**CPU**[4]	**iter.**[3]	**CPU**[4]
RanLCP1	100	5721	113.0	11600	233.7	36611	739.5	71491	1462.3
RanLCP2	200	59744	5474.5	144157	15028.2	48013	7456.4	198282	18831.7
RanLCP3	300	37769	8415.2	171963	46096.1	316489	13201.0	–	–
RanLCP4	100	2378	45.7	3149	60.4	7802	148.2	10369	195.0
RanLCP5	200	1133	112.6	1412	138.1	3425	341.6	4276	444.3
RanLCP6	300	748	200.4	944	246.2	2394	517.9	3033	713.4
DetLCP1	100	32	2.1	36	2.2	136	2.9	157	2.9
DetLCP2	200	37	16.5	42	16.7	156	18.1	178	19.6
DetLCP3	300	40	50.9	45	52.8	167	36.4	189	43.6
HPEasy	100	79	2.9	109	3.5	423	8.0	531	9.7
HPHard	100	64	2.7	85	3.8	855	16.2	1115	20.9
Lemke	100	1057	21.3	1107	22.0	1508	27.5	2261	44.3

TABLE 2

Results for Algorithm 2.1 and extragradient method on LCP.

[1] Algorithm 2.1 with $P = (I + M^T)(I + M)$ and $\theta = 1$.
[2] The extragradient method as described in [31], with $\beta = .7$ and initial $\alpha = 1$.
[3] For all methods, $x^0 = 0$ and the termination criterion is $\|r(x)\| \le \epsilon$.
[4] Time (in seconds) obtained using the intrinsic function SECNDS and with the codes compiled by the DEC Fortran-77 compiler and ran on a Decstation 5000; does not include time to read problem data.

to -1. In our implementation of Algorithm 2.1, we chose P to be $(I + M^T)(I + M)$ and chose $\theta = 1$ (so $\gamma_i = 1$ for all i). The performance of Algorithm 2.1 also benefited substantially from a priori scaling of M and q and, in our test, we scaled M and q by multiplying both with $10 \cdot$ (maximum magnitude of entries of M and $q)^{-1}$. (We did not need to scale M and q for the extragradient method since the scaling is done automatically via its stepsize parameter α.) The test results are summarized in Table 2 below. In general, Algorithm 2.1 required fewer iterations and less time than the extragradient method, though both had difficulty on skew symmetric problems (the first three test problems). On the other hand, we caution that the performance of Algorithm 2.1 strongly depends on the scaling of M and q and finding a suitable choice of scaling can be difficult in general.

Our third set of tests was conducted on VI(X, F) where X is not an orthant or a box. The first test problem, used first by Mathiesen [35], and later in [41, 54], has

$$\begin{aligned} F(x_1, x_2, x_3) &= \begin{bmatrix} .9(5x_2 + 3x_3)/x_1 \\ .1(5x_2 + 3x_3)/x_2 - 5 \\ -3 \end{bmatrix}, \\ X &= \left\{(x_1, x_2, x_3) \in \Re^3_+ \mid x_1 + x_2 + x_3 = 1, x_1 - x_2 - x_3 \le 0\right\}. \end{aligned}$$

We had trouble finding more test problems from the literature, so we created five additional test problems of our own, in which $X = \{x \in \Re^n_+ \mid x_1 + \cdots + x_n = n\}$ and F and n are specified as follows: For the first three problems, F is the function from, respectively, the Kojima-Shindo NCP (with $n = 4$) and the Nash-Cournot NCP (with $n = 5$ and $n = 10$) [41, pp. 321-322]; for the fourth problem, F is affine and is generated as in the problem HPHard of Table 2, but with $n = 20$; for the fifth problem, we took the F from the fourth problem and added to its ith component the linear/quadratic term $\max\{0, x_i\}^2$ for $i = 1, ..., \lfloor n/2 \rfloor$. In our implementation

		Algorithm 2.1[1]		Algorithm 3.2[2]		Extragradient[3]	
Name	n	**iter.**$(nf/np)^4$	CPU[5]	**iter.**$(nf/np)^4$	CPU[5]	**iter.**$(nf/np)^4$	CPU[5]
Mathiesen	3	–	–	25(56/31)	3.9	260(524/524)	66.1
		–	–	18(40/22)	2.7	13(30/30)	3.2
KojimaSh	4	–	–	38(85/47)	3.9	16(36/36)	2.4
Nash5	5	–	–	74(155/81)	6.6	43(89/89)	5.5
Nash10	10	–	–	93(192/99)	10.6	84(172/172)	13.4
HPHard	20	38(38/38)	31.5	286(579/293)	264.3	248(499/499)	395.2
qHPHard	20	–	–	274(555/281)	251.6	239(481/481)	380.4

TABLE 3

Results for Algorithms 2.1, 3.2 and extragradient method on linearly constrained variational inequality problems.

[1] Algorithm 2.1 with $P = (I + M^T)(I + M)$ and $\theta = 1.5$.
[2] Algorithm 3.2 with $P = I$, $\alpha_{-1} = 1$, $\theta = 1.5$, $\rho = .1$ and $\beta = .3$.
[3] The extragradient method as described in [31], with $\beta = .7$ and initial $\alpha = 1$.
[4] For all methods, the termination criterion is $\|r(x)\| \leq 10^{-4}$. (nf denotes the total number of times F is evaluated and np denotes the total number of times a projection onto X is performed.) On the Mathiesen problem, we ran each method twice with $x^0 = (.1, .8, .1)$ and $x^0 = (.4, .3, .3)$ respectively; on the other problems, we used $x^0 = (1, ..., 1)$.
[5] Time (in seconds) obtained using the intrinsic Matlab function etime and with the codes ran on a Decstation 5000; does not include time to read problem data.

of Algorithm 3.2, we chose $P = I$, $\alpha_{-1} = 1$, $\theta = 1.5$, $\rho = .1$ and $\beta = .3$. On the Mathiesen problem, we used the same x^0 as in [54]; on the other problems, we used $x^0 = (1, ..., 1)$. (The F from the Mathiesen problem and from the Nash-Cournot NCP are defined on the positive orthant only.) The test results are summarized in Table 3. In general, Algorithm 3.2 requires more iterations and function evaluations, but fewer projections, than the extragradient method. (The performance of Algorithm 3.2 is also less sensitive to the starting point than the extragradient method. Surprisingly, both methods solved problems, such as the Kojima-Shindo problem, for which F is not monotone.) Thus, on problems where projection onto X is expensive, Algorithm 3.2 may be more practical than the extragradient method, as is reflected in its lower CPU times on all problems except Nash5. But if F is affine, Algorithm 2.1 may be more practical than either method (compare their CPU times on HPHard). In general, the performance of Algorithm 3.2 is insensitive to x^0 or ρ or α_{-1}, as long these parameters are reasonably chosen. We had also tried alternative choices for P and more conservative choices for θ and β (e.g., $\theta = 1$ and $\beta = .7$), but the results were typically worse.

5. Concluding Remarks. We have presented new iterative methods for solving monotone variational inequality problems and have established their convergence and rate of convergence under mild assumptions on the problem. Preliminary computational experience with the new methods suggest the new methods are practical alternatives to the extragradient method.

We mention in passing that Algorithms 2.1 and 3.1 may be generated by the following general approach: We set y in the inequality

$$0 \leq (y - z)^T(\alpha F(x) + z - x) \quad \forall y \in X,$$

where $z = [x - \alpha F(x)]^+$, to x^*; and we set y in the inequality

$$0 \leq \alpha(y - x^*)F(x^*) \quad \forall y \in X,$$

where $x^* \in S$, to z. Then we add the two inequalities and, by using the monotone property of F and, if necessary, the affine property of F, we reduce the resulting inequality to the form:

$$0 \leq (x - x^*)^T T(x) + (\text{an expression involving } \alpha,\ F,\ x \text{ and } z \text{ only})$$

for some mapping T (depending on F and α) from $\Re^n$ to $\Re^n$. Provided that the rightmost term is negative, the method then updates x according to the formula

$$x^{new} := x - \gamma T(x).$$

Algorithm 2.3, as well as the extragradient method, may be similarly generated except we set y in the first inequality to x instead. Then, we need x to be in X which is why an extra projection on to X is needed. (We can also set y in the second inequality to x, but this does not appear to yield anything useful.)

REFERENCES

[1] A. A. Auslender, *Optimisation: Méthodes Numériques*, Masson, Paris, 1976.

[2] A. B. Bakusinskii and B. T. Polyak, *On the solution of variational inequalities*, Soviet Math. Doklady, 15 (1974), pp. 1705–1710.

[3] D. P. Bertsekas and E. M. Gafni, *Projection methods for variational inequalities with application to the traffic assignment problem*, Math. Programming Study, 17 (1982), pp. 139–159.

[4] J. F. Bonnans, *Local analysis of Newton-type methods for variational inequalities and nonlinear programming*, Appl. Math. Optim., 29 (1994), pp. 161–186.

[5] H.-G. Chen, *Forward-backward splitting techniques: theory and applications*, Ph.D. Thesis, Department of Applied Mathematics, University of Washington, Seattle, WA, December 1994.

[6] R. W. Cottle, F. Giannessi, and J.-L. Lions, Eds., *Variational Inequalities and Complementarity Problems: Theory and Applications*, Wiley, New York, NY, 1980.

[7] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, New York, NY, 1992.

[8] S. C. Dafermos, *An iterative scheme for variational inequalities*, Math. Programming, 26 (1983), pp. 40–47.

[9] S. C. Dafermos and S. C. McKelvey, *Partitionable variational inequalities with applications to network and economic equilibria*, J. Optim. Theory Appl., 73 (1992), pp. 243–268.

[10] J. Eckstein and M. C. Ferris, *Operator splitting methods for monotone affine variational inequalities, with a parallel application to optimal control*, Technical Report, Thinking Machines Corporation, Cambridge, MA, and Computer Sciences Department, University of Wisconsin, Madison, WI, December 1994.

[11] M. C. Ferris and O. L. Mangasarian, *Error bounds and strong upper semicontinuity for monotone affine variational inequalities*, Ann. Oper. Res., 47 (1993), pp. 293–305.

[12] M. Fukushima, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Programming, 53 (1992), pp. 99–110.

[13] E. M. Gafni and D. P. Bertsekas, *Two-metric projection methods for constrained optimization*, SIAM J. Contr. Optim., 22 (1984), pp. 936–964.

[14] D. Gay, *Electronic mail distribution of linear programming test problems*, Math. Programming Soc. COAL Newslett., December, 1985.

[15] R. Glowinski, J.-L. Lions, and R. Trémolières, *Numerical Analysis of Variational Inequalities*, North-Holland, Amsterdam, 1981.

[16] P. Harker and J.-S. Pang, *A damped-Newton method for the linear complementarity problem*, in Computational Solution of Nonlinear Systems of Equations, Lectures in Applied Mathematics 26, G. Allgower and K. Georg, eds., American Mathematical Society, Providence, RI, 1990, pp. 265–284.

[17] B. He, *A projection and contraction method for a class of linear complementarity problems and its application in convex quadratic programming*, Appl. Math. Optim., 25 (1992), pp. 247–262.

[18] ———, *A new method for a class of linear variational inequalities*, Math. Programming, 66 (1994), pp. 137–144.

[19] ———, *Solving a class of linear projection equations*, Numer. Math., 68 (1994), pp. 71–80.

[20] A. N. Iusem, *An iterative algorithm for the variational inequality problem*, Mat. Apl. Comput., 13 (1994), pp 103–114.

[21] E. N. Khobotov, *A modification of the extragradient method for the solution of variational inequalities and some optimization problems*, Zh. Vychisl. Mat. i Mat. Fiz., 27 (1987), pp. 1462–1473.

[22] G. M. Korpelevich, *The extragradient method for finding saddle points and other problems*, Matecon, 12 (1976), pp. 747–756.

[23] X.-D. Luo and P. Tseng, *On global projection-type error bound for the linear complementarity problem*, Linear Algebra Appl., to appear.

[24] Z.-Q. Luo, O. L. Mangasarian, J. Ren, and M. V. Solodov, *New error bounds for the linear complementarity problem*, Math. Oper. Res., 19 (1994), pp. 880–892.

[25] Z.-Q. Luo and P. Tseng, *On the linear convergence of descent methods for convex essentially smooth minimization*, SIAM J. Control Optim., 30 (1992), pp. 408–425.

[26] ———, *Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem*, SIAM J. Optim., 2 (1992), pp. 43–54.

[27] T. L. Magnanti and G. Perakis, *On the convergence of classical variational inequality algorithms*, Working Paper, Operations Research Center, Masachusetts Institute of Technology, Cambridge, MA, May 1993.

[28] O. L. Mangasarian, *Convergence of iterates of an inexact matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Optim., 1 (1991), pp. 114–122.

[29] O. L. Mangasarian and J. Ren, *New improved error bounds for the linear complementarity problem*, Math. Programming, 66 (1994), pp. 241–255.

[30] O. L. Mangasarian and M. V. Solodov, *Nonlinear complementarity as unconstrained and constrained minimization*, Math. Programming, 62 (1993), pp. 277–297.

[31] P. Marcotte, *Application of Khobotov's algorithm to variational inequalities and network equilibrium problems*, Inform. Systems Oper. Res., 29 (1991), pp. 258–270.

[32] P. Marcotte and J.-P. Dussault, *A note on a globally convergent Newton method for solving monotone variational inequalities*, Oper. Res. Lett., 6 (1987), pp. 35–42.

[33] ———, *A sequential linear programming algorithm for solving monotone variational inequalities*, SIAM J. Control Optim., 27 (1989), pp. 1260–1278.

[34] P. Marcotte and J.-H. Wu, *On the convergence of projection methods: application to the decomposition of affine variational inequalities*, J. Optim. Theory Appl., 85 (1995), pp. 347-362.

[35] L. Mathiesen, *An algorithm based on a sequence of linear complementarity problems applied to a Walras' equilibrium model: An example*, Math. Programming, 37 (1987), pp. 1-18.

[36] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Helderman–Verlag, Berlin, 1988.

[37] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, 1970.

[38] J.-S. Pang, *Asymmetric variational inequality problems over product sets: applications and iterative methods*, Math. Programming, 31 (1985), pp. 206–219.

[39] ———, *A posteriori error bounds for the linearly–constrained variational inequality problem*, Math. Oper. Res., 12 (1987), pp. 474–484.

[40] J.-S. Pang and D. Chan, *Iterative methods for variational and complementarity problems*, Math. Programming, 24 (1982), pp. 284–313.

[41] J.-S. Pang and S. A. Gabriel, *NE/SQP: A robust algorithm for the nonlinear complementarity problem*, Math. Programming, 60 (1993), pp. 295–337.

[42] P. M. Pardalos and N. Kovoor, *An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds*, Math. Programming, 46 (1986), pp. 235–238.

[43] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, Math. Programming Study, 14 (1981), pp. 206–214.

[44] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[45] R. T. Rockafellar and R. J.-B. Wets, *Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time*, SIAM J. Control Optim., 28 (1990), pp. 810–

822.

[46] M. Sibony, *Méthodes itératives pour les équations et inéquations aux dérivées partielles non-linéares de type monotone*, Calcolo, 7 (1970), pp. 65–183.

[47] D. Sun, *A new step-size skill for solving a class of nonlinear projection equations*, J. Comput. Math., 13 (1995), to appear.

[48] ———, *A class of iterative methods for solving nonlinear projection equations*, J. Optim. Theory Appl., 91 (1996), to appear.

[49] P. Tseng, *Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming*, Math. Programming, 48 (1990), pp. 249–263.

[50] ———, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Control Optim., 29 (1991), pp. 119–138.

[51] ———, *On linear convergence of iterative methods for the variational inequality problem*, J. Comput. Appl. Math., 60 (1995), pp. 237–252.

[52] N. Yamashita and M. Fukushima, *On dtationary points of the implicit Lagrangian for nonlinear complementarity problems*, J. Optim. Theory Appl., 84 (1995), pp. 653–663.

[53] Y. Ye, *A new complexity result on minimization of a quadratic function with a sphere constraint*, in Recent Advances in Global Optimization, C. Floudas and P. M. Pardalos, eds., Princeton University Press, Princeton, NJ, 1992.

[54] L. Zhao and S. Dafermos, *General economic equilibrium and variational inequalities*, Oper. Res. Lett., 10 (1991), pp. 369–376.

[55] C. Zhu and R. T. Rockafellar, *Primal-dual projected gradient algorithms for extended linear-quadratic programming*, SIAM J. Optim., 3 (1993), pp. 751–783.