SIAM J. CONTROL OPTIM.
Vol. 37, No. 3, pp. 765–776

A NEW PROJECTION METHOD FOR VARIATIONAL INEQUALITY PROBLEMS*

M. V. SOLODOV† AND B. F. SVAITER†

Abstract. We propose a new projection algorithm for solving the variational inequality problem, where the underlying function is continuous and satisfies a certain generalized monotonicity assumption (e.g., it can be pseudomonotone). The method is simple and admits a nice geometric interpretation. It consists of two steps. First, we construct an appropriate hyperplane which strictly separates the current iterate from the solutions of the problem. This procedure requires a single projection onto the feasible set and employs an Armijo-type linesearch along a feasible direction. Then the next iterate is obtained as the projection of the current iterate onto the intersection of the feasible set with the halfspace containing the solution set. Thus, in contrast with most other projection-type methods, only two projection operations per iteration are needed. The method is shown to be globally convergent to a solution of the variational inequality problem under minimal assumptions. Preliminary computational experience is also reported.

Key words. variational inequalities, projection methods, pseudomonotone maps

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

PII. S0363012997317475

1. Introduction.

We consider the classical variational inequality problem [1, 3, 7] VI(F,C), which is to find a point x^* such that

$$x^* \in C, \quad \langle F(x^*), x - x^* \rangle \geq 0 \quad \text{for all} \quad x \in C, \tag{1.1}$$

where C is a closed convex subset of $\Re^n$, $\langle \cdot, \cdot \rangle$ denotes the usual inner product in $\Re^n$, and $F : \Re^n \to \Re^n$ is a continuous function. Let S be the solution set of VI(F,C), which we assume to be nonempty. Let x^* be any element of the solution set S. We further assume that

$$\langle F(x), x - x^* \rangle \geq 0 \quad \text{for all} \quad x \in C. \tag{1.2}$$

It is clear that (1.2) is satisfied if $F(\cdot)$ is monotone; i.e.,

$$\langle F(x) - F(y), x - y \rangle \geq 0 \quad \text{for all} \quad x, y \in \Re^n.$$

More generally, (1.2) also holds if $F(\cdot)$ is pseudomonotone (as defined in [11]); i.e., for all $x, y \in \Re^n$

$$\langle F(y), x - y \rangle \geq 0 \quad \Longrightarrow \quad \langle F(x), x - y \rangle \geq 0.$$

Moreover, it is not difficult to construct examples where (1.2) is satisfied but $F(\cdot)$ is not monotone or pseudomonotone everywhere. Typically, condition (1.2) holds under some kind of generalized monotonicity assumptions on $F(\cdot)$, some of which are not difficult to check (see [26, 25]).

*Received by the editors February 28, 1997; accepted for publication (in revised form) June 3, 1998; published electronically March 2, 1999.

http://www.siam.org/journals/sicon/37-3/31747.html

†Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brazil (solodov@impa.br, benar@impa.br). The research of the first author was supported by CNPq grant 300734/95-6 and by PRONEX–Optimization. The research of the second author was supported by CNPq grant 301200/93-9(RN) and by PRONEX–Optimization.

In the case when $F(\cdot)$ is strongly monotone and/or the feasible set C has some special structure (e.g., C is the nonnegative orthant or, more generally, a box), there exist many efficient methods that can be used to solve those special cases of $\mathrm{VI}(F, C)$ (see [4, 5, 14, 18, 19, 20, 22, 23, 34, 36, 2, 16, 29, 15, 28, 27, 31]). In some of those methods, $F(\cdot)$ is further assumed to be differentiable, or Lipschitz continuous, or affine. Sometimes it is also assumed that the method starts close enough to the solution set (i.e., only local convergence is guaranteed). In the general case when $F(\cdot)$ and C do not possess any special structure, relatively few methods are applicable. In that case, projection-type algorithms are of particular relevance (we refer the reader to [32] for a more detailed discussion). The oldest algorithm of this class is the extragradient method proposed in [13] and later refined and extended in [10, 12, 17, 33, 9]. Some new projection-type algorithms that appear to be more efficient than the extragradient method were recently introduced in [32] (see also references therein).

In this paper, we are mainly concerned with the general case when the projection operator

$$P_C[x] := \arg\min_{y \in C} \|y - x\|$$

is computationally expensive (i.e., one has to solve an optimization problem to find a projection). Furthermore, we make no assumptions on the problem other than continuity of $F(\cdot)$ and condition (1.2). In this setting, one of the important tasks in devising efficient algorithms is to minimize the number of projection operations performed at each iteration. We note that in the case when $F(\cdot)$ is not Lipschitz continuous or the Lipschitz constant is not known, the extragradient method, as described in [12, 17, 10, 33], requires a linesearch procedure to compute the stepsize, with a new projection needed for each trial point. The same holds for the modified projection-type method in [32]. Clearly, this can be very computationally expensive. A novel idea to get around this inefficiency was proposed in [9] for the extragradient method. Here we will use this idea to devise a new projection algorithm that has even better properties, both theoretically and in our computational experience.

The algorithm proposed here allows a nice geometric interpretation, which is given in Figure 1.1 for its simplest version. Suppose we have x^i, a current approximation to the solution of $\mathrm{VI}(F, C)$. First, we compute the point $P_C[x^i - F(x^i)]$. Next, we search the line segment between x^i and $P_C[x^i - F(x^i)]$ for a point z^i such that the hyperplane $\partial H_i := \{x \in \Re^n \mid \langle F(z^i), x - z^i \rangle = 0\}$ strictly separates x^i from any solution x^* of the problem. A computationally inexpensive Armijo-type procedure is used to find such z^i. Once the hyperplane is constructed, the next iterate x^{i+1} is computed by projecting x^i onto the intersection of the feasible set C with the halfspace $H_i := \{x \in \Re^n \mid \langle F(z^i), x - z^i \rangle \le 0\}$, which contains the solution set S. It can be seen that x^{i+1} thus computed is closer to any solution $x^* \in S$ than x^i. At each iteration, our algorithm uses one projection onto the set C (to construct the separating hyperplane H_i), and one projection onto the intersection $C \cap H_i$, which gives the next iterate.

Before proceeding, we emphasize the differences between the method of [9] and our Algorithms 2.1 and 2.2. First, the second projection step in our method is onto the intersection $C \cap H_i$. In [9], x^i is projected first onto the separating hyperplane ∂H_i (this point is denoted by $\bar{x}^i$ in Figure 1.1) and then onto C (in Figure 1.1, the resulting point is denoted by $P_C[\bar{x}^i]$). It can be verified that our iterate x^{i+1} is closer to the solution set S than the iterate computed by the method of [9]. We also avoid the extra work of computing the point $\bar{x}^i$. (Even though it can be carried out

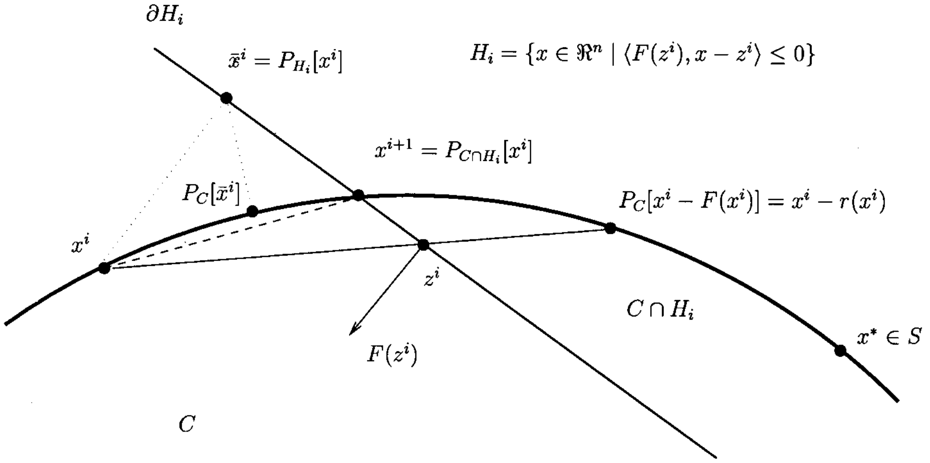
∂H_i
$\bar{x}^i = P_{H_i}[x^i]$
$H_i = \{x \in \Re^n \mid \langle F(z^i), x - z^i \rangle \leq 0\}$
$x^{i+1} = P_{C \cap H_i}[x^i]$
$P_C[\bar{x}^i]$
$P_C[x^i - F(x^i)] = x^i - r(x^i)$
x^i
z^i
$C \cap H_i$
$x^* \in S$
$F(z^i)$
C

FIG. 1.1. *The new projection method.*

explicitly, this is still some extra work.) Furthermore, the search direction in our method is not the same as in [9] for the following reasons. The search directions we use here are $P_C[x^i - \mu_i F(x^i)] - x^i$, where the stepsizes μ_i are chosen so that they are of the same order as the stepsizes η_i generating separating hyperplanes H_i (see Algorithm 2.2). The important point is that we allow *both* of them to go to zero if needed (but at the same rate). Coordination of the two stepsizes proves to be very significant in our computational experience (see section 3). We point out that [9] does not permit the stepsizes to go to zero in the first projection step even if the stepsizes generating the hyperplanes go to zero, and the proof there does not handle this case. The above-mentioned modifications seem to make a drastic difference in the numerical performance when our algorithm is compared to that of [9]. Our preliminary computational experience with the new algorithm is quite encouraging and is reported in section 3. However, we emphasize that comprehensive numerical study is not the primary focus of this paper.

Finally, our convergence results are stated under the assumption (1.2), which is considerably weaker than monotonicity of $F(\cdot)$ used in [9].

2. The algorithm and its convergence.

We first note that solutions of VI(F, C) coincide with zeros of the following projected residual function:

$$r(x) := x - P_C[x - F(x)];$$

i.e., $x \in S$ if and only if $r(x) = 0$.

We now formally state our algorithm described in section 1.

ALGORITHM 2.1. *Choose $x^0 \in C$ and two parameters $\gamma \in (0,1)$ and $\sigma \in (0,1)$. Having x^i, compute $r(x^i)$. If $r(x^i) = 0$, stop. Otherwise, compute*

$$z^i = x^i - \eta_i r(x^i),$$

where $\eta_i = \gamma^{k_i}$, with k_i being the smallest nonnegative integer k satisfying

$$\langle F\left(x^i - \gamma^k r(x^i)\right), r(x^i)\rangle \geq \sigma \|r(x^i)\|^2. \tag{2.1}$$

Compute

$$x^{i+1} = P_{C \cap H_i}[x^i],$$

where

$$H_i = \{x \in \Re^n \mid \langle F(z^i), x - z^i \rangle \le 0\}.$$

The following well-known properties of the projection operator will be used below.

LEMMA 2.1. (see [35]). *Let B be any nonempty closed convex set in $\Re^n$. For any $x, y \in \Re^n$ and any $z \in B$ the following properties hold.*

1. $\langle x - P_B[x], z - P_B[x] \rangle \le 0.$
2. $\|P_B[x] - P_B[y]\|^2 \le \|x - y\|^2 - \|P_B[x] - x + y - P_B[y]\|^2.$

We start with a preliminary result. For now, we assume that the linesearch procedure in Algorithm 2.1 is well defined. This fact will be formally established in Theorem 2.1.

LEMMA 2.2. *Suppose that the linesearch procedure* (2.1) *of Algorithm* 2.1 *is well defined. Then it holds that*

$$x^{i+1} = P_{C \cap H_i}[\bar{x}^i],$$

where

$$\bar{x}^i = P_{H_i}[x^i].$$

Proof. Assuming that the point z^i is well defined, by (2.1) we have that

$$\langle F(z^i), x^i - z^i \rangle > 0.$$

It immediately follows that $x^i \notin H_i$. Also, by (1.2), $\langle F(z^i), x^* - z^i \rangle \le 0$ for any $x^* \in S$ because $z^i = (1 - \eta_i)x^i + \eta_i P_C[x^i - F(x^i)] \in C$ by the convexity of C. Therefore, $x^* \in H_i$. Since also $x^* \in C$, it follows that $C \cap H_i \ne \emptyset$. Because $C \cap H_i$ is a closed convex set that is nonempty, $x^{i+1} = P_{C \cap H_i}[x^i]$ is well defined.

It can be further verified that

$$\begin{aligned} \bar{x}^i = P_{H_i}[x^i] &= x^i - \frac{\langle F(z^i), x^i - z^i \rangle}{\|F(z^i)\|^2} F(z^i) \\ &= x^i - \frac{\eta_i \langle F(z^i), r(x^i) \rangle}{\|F(z^i)\|^2} F(z^i). \end{aligned}$$

Take any $y \in C \cap H_i$. Since $x^i \in C$ but $x^i \notin H_i$, there exist $\beta \in [0, 1]$ such that $\tilde{x} = \beta x^i + (1 - \beta)y \in C \cap \partial H_i$, where $\partial H_i := \{x \in \Re^n \mid \langle F(z^i), x - z^i \rangle = 0\}$. We have

$$\begin{aligned} \|y - \bar{x}^i\|^2 &\ge (1 - \beta)\|y - \bar{x}^i\|^2 \\ &= \|\tilde{x} - \beta x^i - (1 - \beta)\bar{x}^i\|^2 \\ &= \|\tilde{x} - \bar{x}^i\|^2 + \beta^2\|x^i - \bar{x}^i\|^2 - 2\beta\langle \tilde{x} - \bar{x}^i, x^i - \bar{x}^i \rangle \\ &\ge \|\tilde{x} - \bar{x}^i\|^2, \end{aligned} \tag{2.2}$$

where the last inequality follows from Lemma 2.1 applied with $B = H_i$, $x = x^i$, and $z = \tilde{x} \in H_i$. Furthermore, we have

$$\begin{aligned} \|\tilde{x} - \bar{x}^i\|^2 &= \|\tilde{x} - x^i\|^2 - \|x^i - \bar{x}^i\|^2 \\ &\ge \|x^{i+1} - x^i\|^2 - \|x^i - \bar{x}^i\|^2 \\ &= \|x^{i+1} - \bar{x}^i\|^2, \end{aligned} \tag{2.3}$$

where the first equality is by $\bar{x}^i = P_{\partial H_i}[x^i]$, $\tilde{x} \in \partial H_i$, and Pythagoras's theorem; the inequality is by the fact that $\tilde{x} \in C \cap H_i$ and $x^{i+1} = P_{C \cap H_i}[x^i]$; and the last equality is again by Pythagoras's theorem. Combining (2.2) and (2.3), we obtain

$$\|y - \bar{x}^i\| \geq \|x^{i+1} - \bar{x}^i\| \quad \text{for all } y \in C \cap H_i.$$

Hence, $x^{i+1} = P_{C \cap H_i}[\bar{x}^i]$. □

We next prove our main convergence result.

THEOREM 2.1. *Let $F(\cdot)$ be continuous. Suppose the solution set S of VI(F, C) is nonempty and condition* (1.2) *is satisfied.*

Then any sequence $\{x^i\}$ generated by Algorithm 2.1 *converges to a solution of VI(F, C).*

Proof. We first show that the linesearch procedure in Algorithm 2.1 is well defined. If $r(x^i) = 0$, then the method terminates at a solution of the problem. Therefore, from now on, we assume that $\|r(x^i)\| > 0$. Also note that $x^i \in C$ for all i. Suppose that, for some i, (2.1) is not satisfied for any integer k, i.e., that

$$\langle F\left(x^i - \gamma^k r(x^i)\right), r(x^i)\rangle < \sigma \|r(x^i)\|^2 \quad \text{for all } k. \tag{2.4}$$

Applying Lemma 2.1 with $B = C,\ x = x^i - F(x^i),\ z = x^i \in C$, we obtain

$$\begin{aligned} 0 &\geq \langle x^i - F(x^i) - P_C[x^i - F(x^i)], x^i - P_C[x^i - F(x^i)]\rangle \\ &= \|r(x^i)\|^2 - \langle F(x^i), r(x^i)\rangle. \end{aligned}$$

Hence,

$$\langle F(x^i), r(x^i)\rangle \geq \|r(x^i)\|^2. \tag{2.5}$$

Since $x^i - \gamma^k r(x^i) \to x^i$ as $k \to \infty$, and $F(\cdot)$ is continuous, passing onto the limit as $k \to \infty$ in (2.4), we obtain

$$\langle F(x^i), r(x^i)\rangle \leq \sigma \|r(x^i)\|^2.$$

But the latter relation contradicts (2.5) because $\sigma < 1$ and $\|r(x^i)\| > 0$. Hence (2.1) is satisfied for some integer k_i.

Thus the linesearch step is well defined, and by Lemma 2.2 we know that the rest of the method is as well. In particular, $x^{i+1} = P_{C \cap H_i}[\bar{x}^i]$, where $\bar{x}^i = P_{H_i}[x^i]$. By Lemma 2.1 applied with $B = C \cap H_i$, $x = \bar{x}^i$, and $z = x^* \in S \subset C \cap H_i$, we obtain

$$\begin{aligned} 0 &\geq \langle \bar{x}^i - x^{i+1}, x^* - x^{i+1}\rangle \\ &= \|x^{i+1} - \bar{x}^i\|^2 + \langle \bar{x}^i - x^{i+1}, x^* - \bar{x}^i\rangle. \end{aligned}$$

Hence,

$$\langle x^* - \bar{x}^i, x^{i+1} - \bar{x}^i\rangle \geq \|x^{i+1} - \bar{x}^i\|^2.$$

Therefore,

$$\begin{aligned} \|x^{i+1} - x^*\|^2 &= \|\bar{x}^i - x^*\|^2 + \|x^{i+1} - \bar{x}^i\|^2 + 2\langle \bar{x}^i - x^*, x^{i+1} - \bar{x}^i\rangle \\ &\leq \|\bar{x}^i - x^*\|^2 - \|x^{i+1} - \bar{x}^i\|^2 \\ &= \|x^i - x^*\|^2 - \|x^{i+1} - \bar{x}^i\|^2 \end{aligned}$$

$$
\begin{aligned}
&+\left(\frac{\eta_i\langle F(z^i),r(x^i)\rangle}{\|F(z^i)\|}\right)^2-\frac{2\eta_i\langle F(z^i),r(x^i)\rangle}{\|F(z^i)\|^2}\langle F(z^i),x^i-x^*\rangle\\
&=\|x^i-x^*\|^2-\|x^{i+1}-\bar{x}^i\|^2-\left(\frac{\eta_i\langle F(z^i),r(x^i)\rangle}{\|F(z^i)\|}\right)^2\\
&\quad-\frac{2\eta_i\langle F(z^i),r(x^i)\rangle}{\|F(z^i)\|^2}\langle F(z^i),z^i-x^*\rangle\\
&\le\|x^i-x^*\|^2-\|x^{i+1}-\bar{x}^i\|^2-\left(\frac{\eta_i\sigma}{\|F(z^i)\|}\right)^2\|r(x^i)\|^4, \qquad (2.6)
\end{aligned}
$$

where the last inequality follows from (2.1) and (1.2).

Now (2.6) implies that the sequence $\{\|x^i-x^*\|\}$ is nonincreasing. Therefore, it converges. We further deduce that the sequence $\{x^i\}$ is bounded, and so is $\{z^i\}$. Thus there exists a constant $M>0$ such that $\|F(z^i)\|\le M$ for all i. Hence, from (2.6),

$$\|x^{i+1}-x^*\|^2\le\|x^i-x^*\|^2-\|x^{i+1}-\bar{x}^i\|^2-(\sigma/M)^2\eta_i^2\|r(x^i)\|^4. \qquad (2.7)$$

From convergence of $\{\|x^i-x^*\|\}$, it follows from (2.7) that

$$\lim_{i\to\infty}\eta_i\|r(x^i)\|=0. \qquad (2.8)$$

We consider the two possible cases. Suppose first that $\limsup_{i\to\infty}\eta_i>0$. For (2.8) to hold it must then be the case that $\liminf_{i\to\infty}\|r(x^i)\|=0$. Since $r(\cdot)$ is continuous and $\{x^i\}$ is bounded, there exists $\hat{x}$, an accumulation point of $\{x^i\}$, such that $r(\hat{x})=0$. It follows that $\hat{x}\in S$ and we can take $x^*=\hat{x}$ in the preceding arguments and, in particular, in (2.7). Thus the sequence $\{\|x^i-\hat{x}\|\}$ converges. Since $\hat{x}$ is an accumulation point of $\{x^i\}$, it easily follows that $\{\|x^i-\hat{x}\|\}$ converges to zero, i.e., that $\{x^i\}$ converges to $\hat{x}\in S$.

Suppose now that $\lim_{i\to\infty}\eta_i=0$. By the choice of η_i we know that (2.1) was not satisfied for k_i-1 (at least for i large enough, so that $\eta_i<1$); i.e.,

$$\langle F\left(x^i-\gamma^{-1}\eta_i r(x^i)\right),r(x^i)\rangle<\sigma\|r(x^i)\|^2 \quad \text{for all} \quad i\ge i_0. \qquad (2.9)$$

Let $\hat{x}$ be any accumulation point of $\{x^i\}$ and $\{x^{i_j}\}$ be the corresponding subsequence converging to $\hat{x}$. Passing onto the limit in (2.9) along this subsequence, and using (2.5), we obtain

$$\sigma\|r(\hat{x})\|^2\ge\langle F(\hat{x}),r(\hat{x})\rangle\ge\|r(\hat{x})\|^2,$$

implying that $r(\hat{x})=0$, i.e., that $\hat{x}\in S$. Setting $x^*=\hat{x}$ in (2.7) and repeating the previous arguments, we conclude that the whole sequence $\{x^i\}$ converges to $\hat{x}\in S$. This completes the proof. □

We next propose a modification of Algorithm 2.1 that is motivated by our computational experience and appears to be more practical. For $\mu>0$, define

$$r(x,\mu):=x-P_C[x-\mu F(x)].$$

With this definition, we have $r(x,1)=r(x)$.

The idea is to use, for the first projection step at the current iteration, the stepsize that is not too different from the stepsize computed at the previous iteration (a similar technique was also used in [32, Algorithm 3.2]). Note that Algorithm 2.2 has a certain coordination between the stepsizes μ_i in the first projection step and η_i in the step computing the separating hyperplane. For example, both of them can go to zero if needed. This is in contrast with the method of [9], where the stepsize in the first projection step can never go to zero, even when η_i does. We found this coordination mechanism important in our computational results, reported in section 3. Note also that both stepsizes can increase from one iteration to the next.

ALGORITHM 2.2. *Choose* $x^0 \in C$, $\eta_{-1} > 0$, *and three parameters* $\gamma \in (0,1)$, $\sigma \in (0,1)$, *and* $\theta > 1$.

Having x^i, *compute* $r(x^i, \mu_i)$, *where* $\mu_i := \min\{\theta\eta_{i-1}, 1\}$. *If* $r(x^i, \mu_i) = 0$, *stop. Otherwise, compute*

$$z^i = x^i - \eta_i r(x^i, \mu_i),$$

where $\eta_i = \gamma^{k_i}\mu_i$ *with* k_i *being the smallest nonnegative integer* k *satisfying*

$$\langle F\left(x^i - \gamma^k \mu_i r(x^i, \mu_i)\right), r(x^i, \mu_i)\rangle \geq \frac{\sigma}{\mu_i}\|r(x^i, \mu_i)\|^2.$$

Compute

$$x^{i+1} = P_{C\cap H_i}[x^i],$$

where

$$H_i = \{x \in \Re^n \mid \langle F(z^i), x - z^i\rangle \leq 0\}.$$

THEOREM 2.2. *Let* $F(\cdot)$ *be continuous. Suppose that the solution set* S *of VI*(F, C) *is nonempty and* (1.2) *is satisfied.*

Then any sequence $\{x^i\}$ *generated by Algorithm* 2.2 *converges to a solution of VI*(F, C).

Proof. The proof of convergence uses the same ideas as the proof for Algorithm 2.1, so we supply only a sketch.

As with (2.5), it can be established that

$$\langle F(x^i), r(x^i, \mu_i)\rangle \geq \frac{1}{\mu_i}\|r(x^i, \mu_i)\|^2.$$

In particular, it follows that the linesearch procedure is well defined.

The proof then follows the pattern of the proof of Theorem 2.1, with (2.7) replaced by

$$\|x^{i+1} - x^*\|^2 \leq \|x^i - x^*\|^2 - \|x^{i+1} - \bar{x}^i\|^2 - (\sigma\eta_i/M)^2\mu_i^{-2}\|r(x^i, \mu_i)\|^4.$$

We next use the fact (see [6, Lemma 1]) that

$$\|r(x^i, \mu_i)\| \geq \min\{1, \mu_i\}\|r(x^i)\|.$$

It follows that

$$\|x^{i+1} - x^*\|^2 \leq \|x^i - x^*\|^2 - \|x^{i+1} - \bar{x}^i\|^2 - (\sigma\eta_i/M)^2\mu_i^2\|r(x^i)\|^4.$$

Taking into account that $\eta_i = \gamma^{k_i}\mu_i \leq \mu_i$, we further obtain

$$\|x^{i+1} - x^*\|^2 \leq \|x^i - x^*\|^2 - \|x^{i+1} - \bar{x}^i\|^2 - (\sigma/M)^2\eta_i^4\|r(x^i)\|^4,$$

and the rest of the convergence proof is identical to that of Theorem 2.1. □

3. Computational experience. To give some insight into the behavior of the new projection algorithm (Algorithm 2.2), we implemented it 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 a benchmark, we compared the performance of this implementation with analogous implementations of two versions of the extragradient method (as described in [17] and [33]) and with the modified projection algorithm as given in [32, Algorithm 3.2]. We also implemented the algorithm of [9] and tested it on the same problems as the other four methods. We do not report here the full results for the method of [9], mainly because they were rather poor. In particular, like the extragradient method, the method of [9] failed on the first two problems, and was *by far* the worst among all the methods on the remaining test problems. Thus we found it to be not useful for a benchmark comparison (unfortunately, no computational experience was reported in [9]). By contrast, our algorithm seems to perform better than the alternatives in most cases.

The choice of linearly constrained variational inequalities for our experiments is not incidental. It is clear that the new method should be especially effective when feasible sets are "no simpler" than general polyhedra (so that an optimization problem has to be solved to find a projection). In that case, adding one more linear constraint to perform a projection onto $C \cap H_i$ does not increase the cost compared to projecting onto the feasible set C. Actually, if the constraints are nonlinear, projecting onto $C \cap H_i$ can sometimes turn easier than onto C. On the other hand, when C has some special structure (for example, C is a box), adding a linear constraint would require solving an optimization problem, while projecting onto C can be carried out explicitly. In that case, the extragradient methods may be more attractive than our new method. However, in the case when C is a box (or the nonnegative orthant), there are many other efficient methods available (as discussed in section 1), so we will not focus on this case.

Though our experience is limited in scope, it suggests that the new projection method is a valuable alternative to the extragradient [17, 33] and modified projection [32] methods. We describe the test details below.

All Matlab codes were run on the Sun UltraSPARCstation 1 under Matlab version 5.0.0.4064. Our test problems are the same as those used in [32] to test the modified projection method in the nonlinear case. The first test problem, used first by Mathiesen [21], and later in [24, 36], has

$$F(x_1, x_2, x_3) = \begin{bmatrix} .9(5x_2 + 3x_3)/x_1 \\ .1(5x_2 + 3x_3)/x_2 - 5 \\ -3 \end{bmatrix},$$

$$C = \left\{(x_1, x_2, x_3) \in \Re^3_+ \mid x_1 + x_2 + x_3 = 1, x_1 - x_2 - x_3 \leq 0\right\}.$$

For the other test problems, the feasible set is the simplex

$$C = \{x \in \Re^n_+ \mid x_1 + \cdots + x_n = n\},$$

and $F(\cdot)$ and n are specified as follows. For the third to fifth problems, $F(\cdot)$ is the function from, respectively, the Kojima–Shindo Nonlinear Complementarity Problem (NCP) (with $n = 4$) and the Nash–Cournot NCP (with $n = 5$ and $n = 10$) [24, pp. 321–322]. For the sixth problem, $n = 20$ and $F(\cdot)$ is affine, i.e.,

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

with the matrix M randomly generated as suggested in [8]:

$$M = AA^{\top} + B + D,$$

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), with every entry of q uniformly generated from $(-500, 0)$. For the last problem, we took the F from the sixth problem and added to its ith component the linear-quadratic term $\max\{0, x_i\}^2$ for $i = 1, \ldots, \lfloor n/2 \rfloor$.

In the implementation of our Algorithm 2.2, we choose $\sigma = .3$, $\eta_{-1} = 1$, $\gamma = .5$, and $\theta = 4$. Implementations of the modified projection method and the extragradient method of [17] are the same as those reported in [32]. In particular, the parameters for both algorithms were tuned to optimize the performance. In the implementation of the other version of the extragradient method [33, Algorithm C], we set parameters as follows: $\eta = .1$, $\alpha = .3$, and $\gamma = 1.7$. On the Mathiesen problem, we used the same x^0 as in [36]; on the other problems, we used $x^0 = (1, \ldots, 1)$. (The $F(\cdot)$ from the Mathiesen problem and from the Nash–Cournot NCP are defined on the positive orthant only.) The test results are summarized in Tables 3.1 and 3.2. In most cases, Algorithm 2.2 requires fewer iterations, function evaluations, and projections, and takes less CPU time than the other methods considered. Also note that our method solves problems, such as the Kojima–Shindo problem, even when $F(\cdot)$ is not monotone. We caution, however, that this study is very preliminary.

To investigate the effect of different starting points, we tested the methods on the problem Nash5, using as starting points the mesh points of the uniform triangulation of the simplex

$$\left\{ x \in \Re^5 \mid x \geq 0,\ \sum_{j=1}^{5} x_j = 5 \right\}.$$

The triangulation is obtained by cutting the simplex by equally spaced hyperplanes parallel to the faces of the simplex, with four cuts per face (the first cut is the face of the simplex, and the last is the opposing vertex). We have chosen the problem Nash5 because its function is defined on the positive orthant only, so that the *boundary effect* can also be studied. The modified projection method of [32] had trouble when starting at points with many zero components (such as, e.g., the point $(5, 0, 0, 0, 0)$). This is not very surprising, since the modified projection is actually an *infeasible* method; i.e., the iterates generated need not belong to the feasible set. Therefore, in principle, we may need to evaluate F at points outside of the nonnegative orthant, which gives trouble for problems like Nash5.However, the modified projection algorithm managed to solve the problem in most cases, with the average of 95 iterations (note that this is quite a bit more than the 74 iterations reported in Table 3.1 as needed for the starting point $x^0 = (1, 1, 1, 1, 1)$). In general, the more positive components x^0 has, the fewer iterations the modified projection method needs to solve the problem. The extragradient method of Marcotte [17] managed to solve the problem for all starting points, with the average of 62 iterations (which is again considerably higher than the 43 iterations needed when starting at the unit vector). Finally, Algorithm 2.2 managed to solve the problem from all the starting points, except from a few points for which the second and fourth components are *both* zero. On average, 32 iterations were required for convergence, which is fewer than for the modified projection or the

TABLE 3.1

Results for Algorithm 2.2 *and the modified projection method on linearly constrained variational inequality problems.*

		Algorithm 2.2[1]		**Modified Projection [32]** [2]	
Name	n	**iter.**(nf/np)[3]	CPU	**iter.**(nf/np)[3]	CPU
Mathiesen	3	14(53/28)	0.5	30(68/38)	0.9
		14(55/28)	0.5	25(56/31)	0.7
KojimaSh	4	7(16/14)	0.3	38(84/46)	0.7
Nash5	5	24(100/48)	1.5	74(155/81)	1.9
Nash10	10	34(140/68)	3	93(192/99)	2.7
HPHard	20	379(1520/758)	196	692(1391/699)	128
qHPHard	20	317(1272/634)	154	562(1131/569)	98

[1] Algorithm 2.2 with $\sigma = .3$, $\eta_{-1} = 1$, $\gamma = .5$, and $\theta = 4$.

[2] Modified projection method as described in [32, Algorithm 3.2] and parameters set as reported in that reference ($P = I$, $\alpha_{-1} = 1$, $\theta = 1.5$, $\rho = .1$, and $\beta = .3$).

[3] For all methods, the termination criterion is $\|r(x)\| \leq 10^{-4}$. (nf denotes the total number of times $F(\cdot)$ is evaluated and np denotes the total number of times a projection is performed.) CPU denotes time (in seconds) obtained using the intrinsic MATLAB function etime and with the codes run on a Sun UltraSPARCstation 1; does not include time to read problem data. 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 $x^0 = (1, \ldots, 1)$.

TABLE 3.2

Results for the two versions of extragradient method on linearly constrained variational inequality problems.

		Extragradient [17][4]		**Extragradient [33]**[5]	
Name	n	**iter.**(nf/np)[3]	CPU	**iter.**(nf/np)[3]	CPU
Mathiesen	3	—	—	—	—
		—	—	—	—
KojimaSh	4	16(36/36)	0.5	78(157/79)	2.5
Nash5	5	43(89/89)	1.8	92(184/93)	2.8
Nash10	10	84(172/172)	3.4	103(191/172)	5.5
HPHard	20	532(1067/1067)	163	1003(2607/1607)	562
qHPHard	20	461(926/925)	162	892(2536/1536)	503

[4] The extragradient method as described in [17], with $\beta = .7$ and initial $\alpha = 1$. Dash—indicates that the method did not converge.

[5] The extragradient method as described in [33, Algorithm C], with $\eta = .1$, $\alpha = .3$, $\gamma = 1.7$. Dash—indicates that the method did not converge.

extragradient methods. However, the extragradient method appeared to be somewhat more robust for this problem.

4. Concluding remarks.

A new projection algorithm for solving variational inequality problems was proposed. Under minimal assumptions of continuity of the underlying function and generalized monotonicity (for example, pseudomonotonicity), it was established that the iterates converge to a solution of the problem. The new method has some clear theoretical advantages over most of existing projection methods for general variational inequality problems with no special structure. Preliminary computational experience is also encouraging.

Some of the projection ideas presented here also proved to be useful in devising *truly* globally convergent (i.e., the *whole* sequence of iterates is globally convergent to a solution without any regularity assumptions) and locally superlinearly convergent inexact Newton methods for solving systems of monotone equations [30] and monotone NCPs [31].

Acknowledgments. We are grateful to the two anonymous referees and the editor for constructive suggestions that led to improvements in the paper. In particular, the observation that pseudomonotonicity is sufficient to establish convergence of the method is due to a referee.

REFERENCES

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

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

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

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

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

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

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

[8] P. T. Harker and J.-S. Pang, *A damped-Newton method for the linear complementarity problem*, in Computational Solution of Nonlinear Systems of Equations, Lectures in Appl. Math. 26, G. Allgower and K. Georg, eds., AMS, Providence, RI, 1990, pp. 265–284.

[9] 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.

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

[11] S. Karamardian, *Complementarity problems over cones with monotone and pseudomonotone maps*, J. Optim. Theory Appl., 18 (1976), pp. 445–455.

[12] E. N. Khobotov, *A modification of the extragradient method for the solution of variational inequalities and some optimization problems*, USSR Comput. Math. Math. Phys., 27 (1987), pp. 1462–1473.

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

[14] T. L. Magnanti and G. Perakis, *On the Convergence of Classical Variational Inequality Algorithms*, Working Paper, Operations Research Center, MIT, Cambridge, MA, May 1993.

[15] O. L Mangasarian and M. V. Solodov, *A linearly convergent derivative-free descent method for strongly monotone complementarity problems*, Comput. Optim. Appl., to appear.

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

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

[18] 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.

[19] P. Marcotte and J.-P. Dussault, *A sequential linear programming algorithm for solving monotone variational inequalities*, SIAM J. Control Optim., 27 (1989), pp. 1260–1278.

[20] 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.

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

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

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

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

[25] R. Pini and C. Singh, *A survey of recent* (1985–1995) *advances in generalized convexity with applications to duality theory and optimality conditions*, Optimization, 39 (1997), pp. 311–

360.

[26] S. Schaible, S. Karamardian, and J.-P. Crouzeix, *Characterizations of generalized monotone maps*, J. Optim. Theory Appl., 76 (1993), pp. 399–413.

[27] M. V. Solodov, *Implicit Lagrangian*, in Encyclopedia of Optimization, C. Floudas and P. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1999.

[28] M. V. Solodov, *Some optimization reformulations of the extended linear complementarity problem*, Comput. Optim. Appl., to appear.

[29] M. V. Solodov, *Stationary points of bound constrained reformulations of complementarity problems*, J. Optim. Theory Appl., 94 (1997), pp. 449–467.

[30] M. V. Solodov and B. F. Svaiter, *A globally convergent inexact Newton method for systems of monotone equations*, in Reformulation—Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, M. Fukushima and L. Qi, eds., Kluwer Academic Publishers, Norwell, MA, 1998, pp. 355–369.

[31] M. V. Solodov and B. F. Svaiter, *A truly globally convergent Newton-type method for the monotone nonlinear complementarity problem*, SIAM J. Optim., submitted.

[32] M. V. Solodov and P. Tseng, *Modified projection-type methods for monotone variational inequalities*, SIAM J. Control Optim., 34 (1996), pp. 1814–1830.

[33] D. Sun, *A new step-size skill for solving a class of nonlinear projection equations*, J. Comput. Math., 13 (1995), pp. 357–368.

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

[35] E. H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory*, in Contributions to Nonlinear Functional Analysis, E. Zarantonello, ed., Academic Press, New York, 1971, pp. 237–424.

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