Int. J. Contemp. Math. Sciences, Vol. 5, 2010, no. 52, 2595 - 2604

An Interior Point Algorithm for Variational Inequality Problems*

Xiaona Fan and Qinglun Yan

College of Science
Nanjing University of Posts and Telecommunications
Nanjing, Jiangsu 210046, P. R. China
xiaonafan@126.com

Abstract

In this paper, a new interior algorithm for tracing the combined homotopy path of the variational inequality problems is proposed, and its global convergence is established under some conditions. The residual control criteria, which ensures that the obtained iterative points are interior points, is given by the condition that ensures the β-cone neighborhood to be included in the interior part of the feasible region. Hence, the algorithm avoids judging whether the iterative points are the interior points or not in every predictor step and corrector step of the Euler-Newton method so that the computation is reduced greatly. The preliminary numerical experiments demonstrate that the algorithm is efficient and promising.

Mathematics Subject Classifications: 90C33, 90C30, 65C20, 65L05

Keywords: variational inequality problems, combined interior homotopy, path following algorithm, global convergence

1 Introduction

The variational inequality problem (VIP for abbreviation) is to find a vector $x^* \in X \subset R^n$ such that

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

where X is a nonempty closed convex subset of R^n and $F : D \to R^n$ is continuously differentiable on some open set D, which contains X, denoted by VI(X, F).

*This work is supported by Foundation of NJUPT under Grant No. NY208069, the Natural Science Foundation of China under Grant No. 70972083 and 61070234.

In this paper, we restrict the feasible set X to be $X = \{x \in R^n : g(x) \leq 0\}$, where $g(x) = (g_1(x), \cdots, g_m(x))^T$ and $g_i's$ are assumed to be convex.

Let $X^0 = \{x \in R^n : g_i(x) < 0, i = 1, 2, \cdots, m\}$ be the strictly feasible set of (??), $\partial X = X - X^0$ be the boundary set of X and $I(x) = \{i \in \{1, \cdots, m\} : g_i(x) = 0\}$ be the binding set at x. Let R^m_+ and R^m_{++} denote the nonnegative and positive orthant of R^m, respectively.

In 1999, Lin and Li [?] proposed a combined homotopy interior point method for solving the variational inequality problem. They based on the KKT system of VI(X, F) and constructed the homotopy equation as follows:

$$H(w, w^{(0)}, \mu) = \begin{pmatrix} (1-\mu)(F(x) + \nabla g(x)y) + \mu(x - x^{(0)}) \\ Yg(x) - \mu Y^{(0)} g(x^{(0)}) \end{pmatrix} = 0, \tag{2}$$

where $w^{(0)} = (x^{(0)}, y^{(0)})^T \in X^0 \times R^m_{++}, w = (x, y)^T,\ g(x) = (g_1(x), \cdots, g_m(x))^T, \nabla g(x) = (\nabla g_1(x), \cdots, \nabla g_m(x)),\ y = (y_1, \cdots, y_m),\ Y = \mathrm{diag}(y)$.

In 2005, Xu, Yu and Feng [?] utilized the combined homotopy (??) to solve the VIP on the unbounded set. Under some well-known existence conditions given in the literature, it was proved that the combined homotopy method converges globally to a solution for VIPs for almost all $x^{(0)} \in X^0$. Later, based on a concept of solution at infinity, they [?] proposed a new condition for global convergence of a homotopy method for VIPs on unbounded sets.

For the Euler-Newton method(e.g. Allgower and Georg [?]), the applied procedure needs to check in every iterative step whether the obtained iterative vector is in X and scale it if it is not in X, which leads to a very large calculating quantity when the number of the constraints is large. In this paper, we utilize the technique of the β-cone neighborhood and base on the smoothing Newton method [?] to trace the combined interior homotopy pathway for VIP (??). The residual control criteria, which ensures that the obtained iterative points are interior points, is given by the condition that ensures the β-cone neighborhood to be included in the interior part of the feasible region. Hence, the algorithm avoids judging whether the iterative points are the interior points or not in every predictor step and corrector step so that the computation is reduced greatly. The preliminary numerical experiments demonstrate that the algorithm is efficient and promising.

The plan of this paper is organized as follows. In section 2, the definition of the β-cone neighborhood and the basic framework of the algorithm are presented. Then we devote to proving the global convergence of the algorithm in section 3. Finally, section 4 contains the numerical experiments.

2 Algorithm

First, we give the definition of the β-cone neighborhood. Letting

$$\mathcal{C} = \{(w,\mu) : H(w, w^{(0)}, \mu) = 0, \mu \in (0,1]\} \subset X^0 \times R^m_{++} \times (0,1] \triangleq D,$$

we call $\mathcal{C}$ the smooth homotopy path which is to be followed. We next define a β-cone neighborhood around the homotopy path

$$\mathcal{N}(\beta) = \{(w,\mu) : \|H(w, w^{(0)}, \mu)\| \leq \beta\mu, \mu \in (0,1]\},$$

where $\beta > 0$ is called the width of the neighborhood.

The following lemma plays an important role in guaranteeing the β-cone neighborhood to belong to the interior of D.

Lemma 1 *If $\beta_0 = \min_{i\in 1:m} |y_i^{(0)} g_i(x^{(0)})|$ and $\beta \in (0, \beta_0)$, we have $\mathcal{N}(\beta) \subset X^0 \times R^m_{++} \times (0,1]$.*

Proof. From the definition of the β-cone neighborhood, we know when $\|H(w, w^{(0)}, \mu)\| \leq \beta\mu$, by the second part of the homotopy equation (**??**), we have that the inequality $\|Yg(x) - \mu Y^{(0)} g(x^{(0)})\| \leq \beta\mu$ holds, and $|(Yg(x) - \mu Y^{(0)} g(x^{(0)}))_i| \leq \beta\mu$, $i = 1, 2, \cdots, m$, that is to say, $\mu y_i^{(0)} g_i(x^{(0)}) - \beta\mu \leq y_i g_i(x) \leq \mu y_i^{(0)} g_i(x^{(0)}) + \beta\mu$. For $\beta \in (0, \beta_0)$, in terms of $w^{(0)} \in X^0 \times R^m_{++}$ and $\beta_0 \leq -y_i^{(0)} g_i(x^{(0)}), i = 1, 2, \cdots, m$, we have $y_i g_i(x) < (y_i^{(0)} g_i(x^{(0)}) + \beta_0)\mu \leq 0, \quad i = 1, 2, \cdots, m$.

Hence, by $(x^{(0)}, y^{(0)}, 1) \in D$, we have $\mathcal{N}(\beta) \subset X^0 \times R^m_{++} \times (0,1]$.

Let

$$G(w, w^{(0)}, \mu) = \begin{pmatrix} H(w, w^{(0)}, \mu) \\ \mu \end{pmatrix}. \tag{3}$$

For convenience, in the rest paper, we write $G(w, w^{(0)}, \mu)$ and $H(w, w^{(0)}, \mu)$ as $G(w,\mu)$ and $H(w,\mu)$, respectively.

Algorithm 2.1 (the path following algorithm)

Step 0 (Initialization)

Set $k = 0$, $\mu_0 = 1$. Take the width of β-cone neighborhood $\beta \in (0, \beta_0)$, $(w^{(0)}, 1) \in \mathcal{N}(\beta)$, $\alpha \in (0,1)$, $\delta \in (0,1)$, $\epsilon > 0$.

Step 1 (Termination Criterion)

If $\mu_k < \epsilon$, stop, and $w^{(k)} := (x^{(k)}, y^{(k)})$ solves approximately the homotopy equation (**??**).

Step 2 (Computation of the Newton Direction)

Let $(\triangle w^{(k)}, \triangle\mu_k)$ solve the equation

$$G(w^{(k)}, \mu_k) + \nabla G(w^{(k)}, \mu_k)^T \begin{pmatrix} \triangle w^{(k)} \\ \triangle\mu_k \end{pmatrix} = \begin{pmatrix} 0 \\ (1-\alpha)\mu_k \end{pmatrix}. \tag{4}$$

Step 3 (Backtracking Line Search)
Let λ_k be the maximum of the values $1, \delta, \delta^2, \cdots$ such that

$$\|H(w^{(k)} + \lambda_k \triangle w^{(k)}, (1-\alpha\lambda_k)\mu_k)\| \leq (1-\alpha\lambda_k)\beta\mu_k. \tag{5}$$

Set $w^{(k+1)} := w^{(k)} + \lambda_k \triangle w^{(k)},\ \mu_{k+1} := (1-\alpha\lambda_k)\mu_k,\ k := k+1$, and go to Step 1.

Assumption 1 *For any $\mu \in (0,1]$ and $(w,\mu) \in \mathcal{N}(\beta)$, $H'_w(w,\mu)$ is nonsingular.*

Remark 1 *The assumption* **??** *holds if the mapping F is P_0 and $\nabla F(x)$ is a symmetry matrix.*

Proof. Following from (**??**), we have

$$H'_w(w,\mu) = \begin{pmatrix} (1-\mu)(\nabla F(x) + \nabla^2 g(x)) + \mu I & (1-\mu)\nabla g(x) \\ Y\nabla g(x)^T & diag(g(x)) \end{pmatrix} \triangleq \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

By elementary transformation of the matrix, we derive

$$\begin{pmatrix} I & -BD^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} A - BD^{-1}C & 0 \\ C & D \end{pmatrix}.$$

Since the mapping F is P_0, we get $\nabla F(x)$ is a P_0 matrix. Furthermore, $\nabla F(x)$ is a symmetry matrix, thus $\nabla F(x)$ is a positive semidefinite matrix [**?**]. For the convex function $g(x)$, we have the hessian matrix $\nabla^2 g(x)$ and $diag(g(x))$ are positive definite matrices, thus the matrix $A - BD^{-1}C$ is positive definite. Hence, the matrix $H'_w(w,\mu)$ is nonsingular.

Remark 2

(1). From Assumption **??**, by some simple computation, we know the Jacobian $\nabla G(w^{(k)}, \mu_k)$ is nonsingular if $(w^{(k)}, \mu_k) \in \mathcal{N}(\beta)$, and so the Newton equation (**??**) yields a unique solution.

(2). Due to Eq. (**??**) of Step 2, we derive $\triangle\mu_k = -\alpha\mu_k$.

3 Global linear convergence

Before the global linear convergence analysis, we discuss the property of the mapping $G(w,\mu)$.

Lemma 2 *Assume that $G(w,\mu)$ is defined by Eq.* (**??**). *Given the bounded convex set $\mathcal{M} \subset R^{n+m}$, for $\forall\ w \in \mathcal{M}$ and $\mu \in (0,1]$, there exists a positive constant $C > 0$ such that*

$$\left\|\frac{\partial^2 G_i(w,\mu)}{\partial (w,\mu)^2}\right\| \leq C, \quad i = 1, 2, \cdots, n+m+1.$$

Assumption 2 *Given $\mu \in (0,1]$ and the point $(w,\mu) \in \mathcal{N}(\beta)$, there exists $M > 0$ such that $\|\nabla G(w,\mu)^{-1}\| \le M$.*

It is a common assumption (e.g.[**?**, **?**]) which plays an important role in guaranteeing the algorithm has the convergence.

Theorem 1 *If $\mu_k > \epsilon$, then for all $k \ge 0$, $\lambda_k \ge \hat{\lambda} = \delta\bar{\lambda}$, where*

$$\bar{\lambda} = \min\left\{1, \frac{2(1-\alpha)\beta}{C\sqrt{n+m}M^2(\beta+\alpha)^2}\right\}.$$

Hence, the backtracking procedure for evaluating λ_k in Step 3 is finitely terminating.

Proof. Let $(w^{(k)},\mu_k) \in \mathcal{N}(\beta)$ be chosen to satisfy the Newton equation (**??**). It follows from Assumption **??** and Remark 2 and after some simple computation, we have

$$\begin{pmatrix} \triangle w^{(k)} \\ \triangle \mu_k \end{pmatrix} = \nabla G(w^{(k)},\mu_k)^{-1} \begin{pmatrix} -H(w^{(k)},\mu_k) \\ -\alpha\mu_k \end{pmatrix},$$

then

$$\left\| \begin{pmatrix} \triangle w^{(k)} \\ \triangle \mu_k \end{pmatrix} \right\| \le M(\|H(w^{(k)},\mu_k)\| + \alpha\mu_k) \le M(\beta+\alpha)\mu_k.$$

On the other hand, recalling the relation between the mappings G and H, we have

$$\begin{aligned}
&\begin{pmatrix} H(w^{(k)}+\lambda\triangle w^{(k)},(1-\alpha\lambda)\mu_k) \\ 0 \end{pmatrix}_i = \left(G(w^{(k)}+\lambda\triangle w^{(k)},(1-\alpha\lambda)\mu_k) - \begin{pmatrix} 0 \\ (1-\alpha\lambda)\mu_k \end{pmatrix}\right)_i \\
= \;& G_i(w^{(k)},\mu_k) + \lambda\frac{\partial G_i(w^{(k)},\mu_k)}{\partial(w^{(k)},\mu_k)} \begin{pmatrix} \triangle w^{(k)} \\ \triangle \mu_k \end{pmatrix} - \begin{pmatrix} 0 \\ (1-\alpha\lambda)\mu_k \end{pmatrix}_i \\
+ \;& \frac{1}{2}\lambda^2 \begin{pmatrix} \triangle w^{(k)} \\ \triangle \mu_k \end{pmatrix}^T \frac{\partial^2 G_i(\tilde{w}^{(k)},\tilde{\mu}_k)}{\partial(w^{(k)},\mu_k)^2} \begin{pmatrix} \triangle w^{(k)} \\ \triangle \mu_k \end{pmatrix} \\
= \;& (1-\lambda)\begin{pmatrix} H(w^{(k)},\mu_k) \\ 0 \end{pmatrix}_i + \frac{1}{2}\lambda^2 \begin{pmatrix} \triangle w^{(k)} \\ \triangle \mu_k \end{pmatrix}^T \frac{\partial^2 G_i(\tilde{w}^{(k)},\tilde{\mu}_k)}{\partial(w^{(k)},\mu_k)^2} \begin{pmatrix} \triangle w^{(k)} \\ \triangle \mu_k \end{pmatrix},
\end{aligned}$$

where $(\tilde{w}^{(k)},\tilde{\mu}_k) = (w^{(k)}+\lambda\theta_i\triangle w^{(k)}, \mu_k+\lambda\theta_i\triangle\mu_k), \theta_i \in (0,1)$, and the second equality follows from (**??**). Let $A = \left(\partial^2 G_1/\partial(w^{(k)},\mu_k)^2, \cdots, \partial^2 G_{n+m}/\partial(w^{(k)},\mu_k)^2\right)$, then $\|A(\tilde{w}^{(k)},\tilde{\mu}_k)\| \le \sqrt{n+m}C$. From the deduction given above, we have

$$\begin{aligned}
&\|H(w^{(k)}+\lambda\triangle w^{(k)},(1-\alpha\lambda)\mu_k))\| \\
\le \;& (1-\lambda)\|H(w^{(k)},\mu_k)\| + \frac{1}{2}\lambda^2\|A(\tilde{w}^{(k)},\tilde{\mu}_k))\|\|(\triangle w^{(k)},\triangle\mu_k)\|^2 \\
\le \;& (1-\lambda)\beta\mu_k + \frac{\sqrt{n+m}}{2}\lambda^2 CM^2(\beta+\alpha)^2\mu_k.
\end{aligned}$$

It is easy to verify that

$$(1-\lambda)\beta\mu_k + \frac{\sqrt{n+m}}{2}\lambda^2 CM^2(\beta+\alpha)^2\mu_k \le (1-\alpha\lambda)\beta\mu_k$$

whenever

$$\lambda \le \frac{2(1-\alpha)\beta}{C\sqrt{n+m}M^2(\beta+\alpha)^2}.$$

Therefore, taking

$$\bar{\lambda} = \min\left\{1, \frac{2(1-\alpha)\beta}{C\sqrt{n+m}M^2(\beta+\alpha)^2}\right\},$$

we have $\lambda_k \ge \hat{\lambda}$ with $\hat{\lambda} = \delta\bar{\lambda}$.

We are now in the position to show that the algorithm is well defined.

Theorem 2 *The above algorithm is well defined, that is to say, if* $(w^{(k)}, \mu_k) \in \mathcal{N}(\beta)$ *with* $\mu_k > 0$*, we have that* $(w^{(k+1)}, \mu_{k+1})$ *is well defined with the backtracking routine in Step 3 finitely terminating. And we have* $(w^{(k+1)}, \mu_{k+1}) \in \mathcal{N}(\beta)$*, where* $0 < \mu_{k+1} \le \mu_k$*.*

Proof. Let $(w^{(k)}, \mu_k) \in \mathcal{N}(\beta)$, where $\mu_k \in (0, 1]$. $\mu_k < \epsilon$ if and only if $w^{(k)}$ solves approximately Eq.(**??**). If $w^{(k)}$ does not solve Eq.(**??**), by Remark (1), $(\triangle w^{(k)}, \triangle\mu_k)$ is unique. Let $(w^{(k+1)}, \mu_{k+1}) = (w^{(k)} + \lambda_k\triangle w^{(k)}, (1-\alpha\lambda_k)\mu_k)$. By Theorem **??**, we have the backtracking routine in Step 3 is finitely terminating. Hence, (**??**) can also be regarded as an instance of a standard backtracking line search routine and is finitely terminating with $0 < \mu_{k+1} \le \mu_k$. (**??**) implies $(w^{(k+1)}, \mu_{k+1}) \in \mathcal{N}(\beta)$. This completes the proof.

We are now in the position to state and prove the global linear convergence result for the algorithm described in the preceding section. Assume that the algorithm does not terminate finitely, then we have the following conclusion.

Theorem 3 *Suppose that Assumption 1 holds for the infinite sequence* $\{(w^{(k)}, \mu_k)\}$ *generated by the algorithm. Then*

(i) *For* $k = 0, 1, 2, \cdots$,

$$(w^{(k)}, \mu_k) \in \mathcal{N}(\beta), \tag{6}$$

$$(1-\alpha\lambda_{k-1})\cdots(1-\alpha\lambda_0) = \mu_k. \tag{7}$$

(ii) *For all* $k \ge 0$, $\lambda_k \ge \hat{\lambda} = \delta\bar{\lambda}$*, where*

$$\bar{\lambda} = \min\left\{1, \frac{2(1-\alpha)\beta}{C\sqrt{n+m}M^2(\beta+\alpha)^2}\right\}.$$

Therefore, μ_k *converges to* 0 *at a global linear rate.*

(iii) *The sequence* $\{(x^{(k)}, y^{(k)})\}$ *converges to a solution of* (**??**)*, i.e.,* $\{x^{(k)}\}$ *converges to a solution of the VIP* (**??**)*.*

Proof.

(i) We establish (**??**) and (**??**) by induction on k. Clearly these relations hold for $k = 0$. Now assume that they hold for some $k > 0$. By Theorem **??**, the algorithm is well defined and so (**??**) and (**??**) hold with k replaced by $k + 1$. Hence, by induction, (**??**) and (**??**) hold for all k.

(ii) By Theorem **??**, we have $\lambda_k \geq \hat{\lambda} = \delta\bar{\lambda}$, which combines with (**??**) implying $\mu_k \leq (1 - \alpha\hat{\lambda})^k \mu_0 = (1 - \alpha\hat{\lambda})^k$, for all k sufficiently large. Thus $\{\mu_k\}$ converges globally linearly to zero.

(iii) Let $(\triangle w^{(k)}, \triangle\mu_k)$ be chosen to satisfy the Newton equation (**??**). Then,

$$\begin{aligned} \left\| \begin{pmatrix} w^{(k+1)} \\ \mu_{k+1} \end{pmatrix} - \begin{pmatrix} w^{(k)} \\ \mu_k \end{pmatrix} \right\| &= \lambda_k \left\| \begin{pmatrix} \triangle w^{(k)} \\ \triangle\mu_k \end{pmatrix} \right\| \leq \left\| \begin{pmatrix} \triangle w^{(k)} \\ \triangle\mu_k \end{pmatrix} \right\| \\ &\leq M(\beta + \alpha)\mu_k \leq M(\beta + \alpha)(1 - \alpha\hat{\lambda})^k. \end{aligned}$$

Therefore, $\{w^{(k)}\}$ is a Cauchy sequence and converges to a point w^*. It follows from $(w^{(k)}, \mu_k) \in \mathcal{N}(\beta)$ that $w^* \in X \times R^m_+$. Hence, $(w^*, 0)$ is a solution of the homotopy equation (**??**) and correspondingly, the point x^* solves the VIP (**??**).

4 Numerical experiments

We have compared Algorithm 2.1 with the Euler-Newton algorithm applied in [**?**] for several examples. Both of them were implemented in MATLAB. For all test problems, we choose the accuracy parameter $\epsilon = 10^{-6}$ and the constants $\mu_0 = 1.0, \alpha = 0.7, \delta = 0.5, \beta = \beta_0$ in Algorithm 2.1 and in Euler-Newton algorithm, we take the accuracy parameters $\epsilon_1 = 10^{-4}$, $\epsilon_2 = 10^{-3}$, $\epsilon_3 = 10^{-6}$ and $h_0 = 0.3$. **IT** denotes the number of iterations, **ACPU** denotes the average total cost time for solving the problem among the ten runs and **CPU** which we use in the last table denotes the total cost time for solving the problem, x^* denotes the approximate solution of the considered problem, and μ^* denotes the value when the algorithm terminates.

Example 5.1

$$\begin{aligned} F(x_1, x_2) &= \begin{pmatrix} -2(x_1 - 2) \\ 2(x_2 - 4) \end{pmatrix}; \\ g_1(x) &= -x_1; g_2(x) = -1 - x_2; \\ g_3(x) &= x_1 + x_2 - 3; g_4(x) = (x_1 - 1)^2 + x_2^2 - 4. \end{aligned}$$

In this example, we take the initial point $x_0 = (1, 1)^T$. The numerical results are summarized in Table 1.

Table 1: The numerical results of Example 5.1

method	ACPU	IT	x^*	μ^*
Algorithm 2.1	0.0139	15	$(0.0000, 1.7321)^T$	7.1513×10^{-7}
Euler-Newton	0.0641	79	$(0.0000, 1.7321)^T$	2.9948×10^{-7}

Example 5.2([?], Problem 100)

$$\begin{aligned}
F(x) &= (2(x_1-10), 10(x_2-12), 4x_3^3, 6(x_4-11), 60x_5^5, 14x_6-4x_7-10, -4x_6+4x_7^3-8)^T;\\
g_1(x) &= 2x_1^2+3x_2^4+x_3+4x_4^2+5x_5-127;\\
g_2(x) &= 7x_1+3x_2+10x_3^2+x_4-x_5-282;\\
g_3(x) &= 23x_1+x_2^2+6x_6^2-8x_7-196;\\
g_4(x) &= 4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-11x_7.
\end{aligned}$$

In this example, we take the initial point $x_0 = (1, 2, -0.4, 4, -0.6, 1, 1.6)^T$. The numerical results are summarized in Table 2.

Table 2: The numerical results of Example 5.2

method	ACPU	IT	x^*	μ^*
Algorithm 2.1	0.0359	22	$(2.3305, 1.9514, -0.4775, 4.3657, -0.6245, 1.0381, 1.5942)^T$	9.6714×10^{-7}
Euler-Newton	0.3625	344	$(2.3305, 1.9514, -0.4775, 4.3657, -0.6245, 1.0381, 1.5942)^T$	1.9096×10^{-7}

Example 5.3([?], Problem 34)

$$\begin{aligned}
F(x_1,x_2,x_3) &= (-1,0,0)^T;\\
g_1(x) &= e^{x_1}-x_2; g_2(x) = e^{x_2}-x_3;\\
g_3(x) &= x_1-100; g_4(x) = -x_1;\\
g_5(x) &= x_2-100; g_6(x) = -x_2;\\
g_7(x) &= x_3-10; g_8(x) = -x_3.
\end{aligned}$$

In this example, we take the initial point $x_0 = (0.8, 2.3, 9.99)^T$. The numerical results are summarized in Table 2.

Table 3: The numerical results of Example 5.3

method	ACPU	IT	x^*	μ^*
Algorithm 2.1	0.2013	95	$(0.8340, 2.3026, 10.0000)^T$	3.0572×10^{-7}
Euler-Newton	0.3578	333	$(0.8340, 2.3026, 10.0000)^T$	1.2760×10^{-7}

Example 5.4

$$\begin{aligned} F(x) &= Ax + r; \\ g_j(x) &= x_j - 3; \\ g_{j+n}(x) &= 1 - x_j; \\ j &= 1, \cdots, n, \end{aligned}$$

where $A = UDU^T$ is a $n \times n$ matrix, $U = I_n - \frac{2}{\|z\|^2} zz^T$, D is a stochastic diagonal matrix whose diagonal elements belong to $(0, 1)$; $z \in R^n$ is stochastic vector whose components belong to $(0, 1)$. In this example, we take the initial point $x_0 = 2e$, where e is a n-dimensional column vector whose components are all 1. The numerical results are summarized in Table 4.

Table 4: The numerical results of Example 5.4

$n \times 2n$	method	CPU	IT	μ^*
5×10	Algorithm 2.1	0.0310	15	4.0135×10^{-7}
	Euler-Newton	0.1250	129	8.9900×10^{-7}
10×20	Algorithm 2.1	0.0310	16	3.3112×10^{-7}
	Euler-Newton	0.2350	170	8.7670×10^{-7}
50×100	Algorithm 2.1	0.5160	26	5.5573×10^{-7}
	Euler-Newton	23.6720	943	8.0000×10^{-7}
100×200	Algorithm 2.1	2.7350	27	5.3251×10^{-7}
	Euler-Newton	485.3280	1473	8.8982×10^{-7}
200×400	Algorithm 2.1	26.3280	37	4.1287×10^{-7}
	Euler-Newton	–	–	–

References

[1] E. L. Allgower and K. Georg, Numerical Continuation Methods: An Introduction, Springer-Vergal, Berlin, New York, 1990.

[2] J. Burke and S. Xu, A non-interior predictor-corrector path following algorithm for the monotone linear complementarity problem, Technical report, Department of Mathematics, University of Washington, Seattle, WA 98195, 1997.

[3] J. Han, N. Xiu and H. Qi, The Theory and Algorithm for Nonlinear Complementarity Problems(Chinese), Science Publishers, Beijing, 2004.

[4] W. Hock and K. Schittkowski, Test examples for nonlinear programming codes, Springer, Berlin, 1981.

[5] Z.H. Huang, J. Han and Z. Chen, Predictor-Corrector Smoothing Newton Method, Based on a New Smoothing Function, for Solving the Nonlinear Complementarity Problem with a P_0 Function, Journal of Optimization Theory and Applications, 117 (2003), 39-68.

[6] Z. Lin and Y. Li, Homotopy method for solving variational inequalities, Journal of Optimization Theory and Applications, 100 (1999), 207-218.

[7] L. Qi, D. Sun and G. Zhou, A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities, Mathematical Programming-Series A 87 (2000), 1-35.

[8] Q. Xu, B. Yu and G. Feng, Homotopy method for solving variational inequalities in unbounded sets, Journal of Global Optimization, 31 (2005), 121-131.

[9] Q. Xu, B. Yu, G. Feng and C. Dang, A condition for global convergence of a homotopy method for variational inequality problems on unbounded sets, Optimization Methods and Software, 22 (2007), 587-599.

Received: June, 2010