Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 9, No. 3, pp. 493–509, July 2013

DERIVATIVE-FREE CONJUGATE GRADIENT TYPE METHODS FOR SYMMETRIC COMPLEMENTARITY PROBLEMS*

Qiong Li

Abstract: In this paper, we are concerned with the derivative-free method for solving symmetric nonlinear complementarity problems. We first transfer the problem into an equivalent nonsmooth equation. We then extend two recently developed modified PRP conjugate gradient methods to solve this nonsmooth equation. The methods are derivative-free and norm descent. Under mild conditions, we show that both methods are globally convergent. We also report preliminary numerical experiments to show the efficiency of the methods.

Key words: *symmetric nonlinear complementarity problems, derivative-free methods, conjugate gradient type methods*

Mathematics Subject Classification: *65K15, 90C33, 90C56*

1 Introduction

We consider the nonlinear complementarity problem, NCP(F) for short, which is to find a vector $x \in R^n$ satisfying

$$x \geq 0, \quad F(x) \geq 0, \quad \text{and} \quad x^T F(x) = 0, \tag{1.1}$$

where $F : R^n \to R^n$ is continuously differentiable. We focus on the iterative methods for solving problem (1.1) in which the derivative of F needs not to be computed. These methods are called derivative-free methods. Derivative-free methods form an important class of iterative methods for solving the NCP(F), and has received much attention, see, e.g., [2,3,6,7,12,13,17,18,23,26,27]. All these derivative-free methods are globally convergent under suitable conditions. Some methods, e.g., [18,27], are linearly convergent to the solution under the strong monotonicity assumption on F.

The existing derivative-free methods generate search directions by one of the following two ways. One is to simply let

$$d_k = -\frac{\partial \varphi}{\partial b}(x^k, F(x^k)), \tag{1.2}$$

*Supported by the NSF of China Grant 11071087 and science foundation of China Three Gorges University grant KJ2013B017.

ISSN 1348-9151

where $\varphi = \frac{1}{2}\phi_{\mathrm{FB}}^2$, ϕ_{FB} is the so-called Fischer-Burmeister function defined by

$$\phi_{\mathrm{FB}}(a,b) = a + b - \sqrt{a^2 + b^2} \tag{1.3}$$

(or its variants), see [2, 3, 6, 7, 12, 13, 23]. The other way is to let

$$d_k = -\partial\psi_\alpha(x^k, F(x^k))/\partial b - \rho[\partial\psi_\alpha(x^k, F(x^k))/\partial a], \tag{1.4}$$

where

$$\psi_\alpha(a,b) := ab + \frac{1}{2\alpha}((a-\alpha b)_+^2 - a^2 + (b-\alpha a)_+^2 - b^2),$$

see [13, 18, 26, 27]. Almost all of the existing derivative-free methods in the literature are limited to solve monotone or strongly monotone nonlinear complementarity problems. In other words, the descent property of d_k strongly depends on the monotone property of F. If F is not monotone, then it is not known whether d_k is still a descent direction of the merit function.

Another kind of derivative-free methods for solving NCP(F) is the quasi-Newton method [14]. The quasi-Newton methods are globally convergent and even superlinearly convergent if the strict complementarity holds at the solution [14]. Most existing quasi-Newton methods are also not descent in the sense that the direction generated by the method may not be descent for the merit function. The purpose of this paper is to develop descent derivative-free methods for solving NCP(F). We only consider the problem where $F'(x)$ is symmetric for any x. We call this problem symmetric complementarity problem. The study in the symmetric linear complementarity problems has received much attention in the last thirty years, see e.g. [5, 11, 19, 20, 22, 24]. Zhang and Li [28] proposed a descent BFGS type methods for symmetric nonlinear complementarity problems. It is an extension of the norm descent BFGS method proposed by Gu, Li, Qi and Zhou [8] for solving symmetric nonlinear equations. The method in [28] is monotone in the sense that the generated sequence of residual functions is decreasing. It possesses global and superlinear convergence properties under some reasonable conditions.

We develop another kind of derivative-free methods, which we call conjugate gradient type methods, for solving symmetric nonlinear complementarity problems. The basic idea is to extend some recently developed descent conjugate gradient methods in the solution of optimization problem to solve some equivalent nonsmooth equation reformulation to the NCP(F). Due to the low storage requirement, the methods can be applied for solving large-scale nonlinear complementarity problems. To design the method, we first reformulate the symmetric nonlinear complementarity problems into a nonsmooth system of equations. We then propose derivative-free conjugate gradient type methods (DFCGM) for solving the nonsmooth equations reformulation using a similar technique to the conjugate gradient type methods for symmetric smooth equations in [15, 16]. It should be pointed out that the directions generated by our methods are different from (1.2) and (1.4). Under appropriate conditions, we establish the global convergence of the proposed methods.

The remainder of the paper is organized as follows. In the next two sections, we propose two conjugate gradient type derivative-free methods that generate descent directions for some merit function and establish their global convergence respectively. In Section 4, we present some numerical results to show the efficiency of the proposed methods.

2 Two Derivative-Free Conjugate Gradient Type Algorithms

We begin this section with two descent conjugate gradient methods for solving the unconstrained optimization problem

$$\min_{x\in R^n} f(x),$$

where $f : R^n \to R$ is a continuously differentiable function.

Recently, Zhang, Zhou and Li [29] proposed a modified PRP conjugate gradient method for solving the above problem. We call it the MPRP method. At iteration k, the MPRP method generates a direction d_k by

$$d_k = \begin{cases} -\nabla f(x_k) & \text{if} \quad k = 0, \\ -\nabla f(x_k) + \beta_k^{PRP} d_{k-1} - \theta_k y_{k-1} & \text{if} \quad k \geq 1, \end{cases} \tag{2.1}$$

where

$$y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}), \quad \beta_k^{PRP} = \frac{\nabla f(x_k)^T y_{k-1}}{\|\nabla f(x_{k-1})\|^2}, \quad \theta_k = \frac{\nabla f(x_k)^T d_{k-1}}{\|\nabla f(x_{k-1})\|^2}.$$

Cheng [4] proposed another modified PRP method called two-term modified PRP (TMPRP) method in which the direction d_k takes the form

$$d_k = \begin{cases} -\nabla f(x_k) & \text{if} \quad k = 0, \\ -\nabla f(x_k) + \beta_k^{PRP}(I - \dfrac{\nabla f(x_k)\nabla f(x_k)^T}{\|\nabla f(x_k)\|^2})d_{k-1} & \text{if} \quad k \geq 1. \end{cases} \tag{2.2}$$

It is not difficult to see that the direction d_k determined by (2.1) or (2.2) provides a sufficient descent direction for f at x_k in the sense that it satisfies

$$\nabla f(x_k)^T d_k = -\|\nabla f(x_k)\|^2.$$

We are going to extend the above methods for solving a nonsmooth equation reformulation to the NCP(F). Let $\phi_{\text{FB}} : R^2 \to R$ be defined by (1.3) and $\Phi_{\text{FB}}(x) = (\phi_1(x), \ldots, \phi_n(x))^T$ with

$$\phi_i(x) = \phi_{\text{FB}}(x_i, F_i(x)), \quad i = 1, \ldots, n.$$

Then problem (1.1) is reformulated as the following system of semismooth equations

$$\Phi_{\text{FB}}(x) = 0. \tag{2.3}$$

The concept of semismoothness was introduced by Mifflin [21] and extended by Qi and Sun [25]. The following definition is due to Qi and Sun [25].

Definition 2.1. We say that function $F : R^n \to R^n$ is semismooth at a point $x \in R^n$ if it is locally Lipschitzian at x and

$$\lim_{V\in\partial F(x+th'),h'\to h,t\downarrow 0} Vh'$$

exists for any $h \in R^n$, where $\partial F(x)$ is the generalized Jacobian of F at x.

Denote

$$\Psi(x) = \frac{1}{2}\|\Phi_{\rm FB}(x)\|^2.$$

It is well known that function $\Phi_{\rm FB}(x)$ is semismooth and $\Psi(x)$ is smooth on R^n. Moreover, the generalized Jacobian of $\Phi_{\rm FB}(x)$ satisfies

$$\partial\Phi_{\rm FB}^T(x) \subseteq diag(a_i(x)) + \nabla F(x) diag(b_i(x)),$$

where

$$a_i(x) = \begin{cases} 1 - \dfrac{x_i}{\sqrt{x_i^2 + F_i^2(x)}} & \text{if} \quad (x_i, F_i(x)) \neq (0,0), \\ 1 - \xi_i & \text{otherwise}, \end{cases}$$

$$b_i(x) = \begin{cases} 1 - \dfrac{F_i(x)}{\sqrt{x_i^2 + F_i^2(x)}} & \text{if} \quad (x_i, F_i(x)) \neq (0,0), \\ 1 - \eta_i & \text{otherwise}, \end{cases}$$

where $(\xi_i, \eta_i) \in R^2, \|(\xi_i, \eta_i)\| \leq 1$. The gradient of the function $\Psi(x)$ takes the form

$$\nabla\Psi(x) = V^T \Phi_{\rm FB}(x), \quad \forall V \in \partial\Phi_{\rm FB}(x).$$

For the sake of convenience, we denote

$$\begin{cases} \tilde{\Phi}(x) = diag(b_i(x))\Phi_{\rm FB}(x), \\ p(x) = diag(a_i(x))\Phi_{\rm FB}(x), \\ q_\lambda(x) = \lambda^{-1}[F(x + \lambda\tilde{\Phi}(x)) - F(x)], \\ g_\lambda(x) = p(x) + q_\lambda(x). \end{cases} \tag{2.4}$$

It is easy to see that when λ is sufficiently small, we have $q_\lambda(x) \approx F'(x)\tilde{\Phi}(x)$. If $F'(x)$ is symmetric, we can use $q_\lambda(x)$ to approximate $\nabla F(x)\tilde{\Phi}(x)$. Therefore when λ is sufficiently small and $F'(x) = \nabla F(x)$, we can use $g_\lambda(x)$ to approximate $\nabla\Psi(x)$. Based on the above observation, we extend the MPRP method and the TMPRP method to solve the semismooth equation reformulation (2.3) in which the search directions d_k are determined by

$$d_k^{TT}(\lambda) = \begin{cases} -g_\lambda(x_0), & \text{if} \quad k = 0, \\ -g_\lambda(x_k) + \beta_k(\lambda) d_{k-1}^{TT} - \theta_k(\lambda) y_{k-1}(\lambda), & \text{if} \quad k \geq 1, \end{cases} \tag{2.5}$$

and

$$d_k^{TM}(\lambda) = \begin{cases} -g_\lambda(x_0), & \text{if} \quad k = 0, \\ -g_\lambda(x_k) + \beta_k(\lambda)(I - \dfrac{g_\lambda(x_k) g_\lambda(x_k)^T}{\|g_\lambda(x_k)\|^2}) d_{k-1}^{TM}, & \text{if} \quad k \geq 1, \end{cases} \tag{2.6}$$

respectively. The parameters in (2.5) and (2.6) are given by

$$y_{k-1}(\lambda) = g_\lambda(x_k) - g_{k-1}, \quad \beta_k(\lambda) = \frac{g_\lambda(x_k)^T (g_\lambda(x_k) - g_{k-1})}{\|g_{k-1}\|^2}$$

and

$$\theta_k(\lambda) = \frac{g_\lambda(x_k)^T d_{k-1}^{TT}}{\|g_{k-1}\|^2},$$

g_{k-1} will be determined by Procedure 1.

In the latter part of the paper, without confusion, we use d_k to denote either d_k^{TT} or d_k^{TM}.

The following Lemma shows that when λ is sufficiently small, the search direction $d_k(\lambda)$ is a descent direction of Ψ at x_k.

Lemma 2.2. *Let σ_1 and σ_2 be positive constants, if x_k is not a stationary point of the merit function $\Psi(x)$, then there exists a constant $\bar{\lambda} > 0$ depending on k such that when $\lambda \in (0, \bar{\lambda})$, $d_k(\lambda)$ satisfies*

$$\nabla\Psi(x_k)^T d_k(\lambda) < 0. \tag{2.7}$$

Moreover, the inequality

$$\Psi(x_k + \lambda d_k(\lambda)) - \Psi(x_k) \le -\sigma_1 \|\lambda d_k(\lambda)\|^2 - \sigma_2 \|\lambda \Phi_{\rm FB}(x_k)\|^2 \tag{2.8}$$

holds for all $\lambda > 0$ sufficiently small.

Proof. It is clear that

$$\lim_{\lambda \to 0} \nabla\Psi(x_k)^T d_k(\lambda) = -\|\nabla\Psi(x_k)\|^2 < 0.$$

This shows that (2.7) holds for all $\lambda > 0$ sufficiently small. We turn to the proof of (2.8). Notice that

$$\lim_{\lambda \to 0} \lambda^{-1}[\Psi(x_k + \lambda d_k(\lambda)) - \Psi(x_k)] = \lim_{\lambda \to 0} \nabla\Psi(x_k)^T d_k(\lambda) = -\|\nabla\Psi(x_k)\|^2 < 0.$$

Since

$$\lim_{\lambda \to 0} g_\lambda(x_k) = \nabla\Psi(x_k),$$

and $d_k(\lambda)$ is determined by (2.5) or (2.6), we have that there exists a constant $l > 0$ such that $\|d_k(\lambda)\| \le l$ for any $\lambda > 0$ sufficiently small. Hence the right-hand side of (2.8) is $o(\lambda)$. We claim that inequality (2.8) holds for all $\lambda > 0$ sufficiently small. □

The following Procedure 1 provides a way to determine a parameter $\lambda_k > 0$ such that the direction d_k satisfies (2.8). Procedure 2 gives a way to determine the steplength α_k.

Procedure 1. Let constant $\rho \in (0, 1)$ be given. Let i_k be the smallest nonnegative integer such that (2.8) holds with $\lambda = \rho^i, i = 0, 1, \ldots$. Let $\lambda_k = \rho^{i_k}$, $q_k = q_{\lambda_k}(x_k)$, $g_k = g_{\lambda_k}(x_k)$, $\beta_k = \beta_k(\lambda_k)$, $d_k = d_k(\lambda_k)$.

Procedure 2. Let i_k and d_k be determined by Procedure 1. If $i_k = 0$, let $\alpha_k = 1$. Otherwise, let j_k be the largest positive integer $j \in \{0, 1, 2, \ldots, i_k - 1\}$ satisfying

$$\Psi(x_k + \rho^{i_k - j} d_k) - \Psi(x_k) \le -\sigma_1 \|\rho^{i_k - j} d_k\|^2 - \sigma_2 \|\rho^{i_k - j} \Phi_{\rm FB}(x_k)\|^2.$$

Let $\alpha_k = \rho^{i_k - j_k}$.

It follows from Lemma 2.2 that Procedures 1 and 2 are well defined. It is easy to see from Procedure 2 that if $\alpha_k \neq 1$, then the following inequality holds

$$\Psi(x_k + \rho^{-1}\alpha_k d_k) - \Psi(x_k) > -\sigma_1 \|\rho^{-1}\alpha_k d_k\|^2 - \sigma_2 \|\rho^{-1}\alpha_k \Phi_{\rm FB}(x_k)\|^2.$$

Now, we propose a derivative-free method for solving (2.3). We call it the DFCG (Derivative-Free Conjugate Gradient type) algorithm. The steps of the algorithm are stated as follows.

DFCG Algorithm

Step 0 Given constants $\sigma_1 > 0, \sigma_2 > 0$, $\rho \in (0, 1)$ and an initial point $x_0 \in R^n$. Let $k := 0$.

Step 1 Determine d_k and α_k by Procedure 1 and Procedure 2, respectively. Let $x_{k+1} = x_k + \alpha_k d_k$.

Step 2 Let $k := k+1$. Go to **Step 1**.

When d_k is determined by d_k^{TT} or d_k^{TM}, we call the corresponding algorithms as Algorithm 1 or Algorithm 2 respectively.

3 Global Convergence

In this section, we prove the global convergence of the DFCG Algorithm under the following assumptions.

Assumption A

(1) The level set $\Omega = \{x \in R^n \mid \Psi(x) \le \Psi(x_0)\}$ is bounded.

(2) In some neighborhood N of Ω, $F'(x)$ is symmetric for any $x \in N$. Moreover, F' is Lipschitz continuous, namely, there exists a constant $L_2 > 0$ such that

$$\|F'(x) - F'(y)\| \le L_2\|x - y\|, \quad \forall x, y \in N.$$

Remark 3.1. (i) The assumption that the level set is bounded holds for uniform P-function F, see [12] for details.

(ii) The condition (2) of Assumption A obviously implies that there exists a constant $L_1 > 0$ such that

$$\|F(x) - F(y)\| \le L_1\|x - y\|, \quad \forall x, y \in N.$$

(iii) Since $\{\Psi(x_k)\}$ is decreasing, it is clear that the sequence $\{x_k\}$ is contained in Ω. In addition, we get from Assumption A that there are constants $\gamma_1 > 0$ and $\gamma_2 > 0$ such that

$$\|F(x)\| \le \gamma_1, \quad \|F'(x)\| \le \gamma_2, \quad \forall x \in N.$$

In the latter part of the paper, we always suppose that the conditions in Assumption A hold. Without specification, we let $\{x_k\}$ and $\{d_k\}$ be the iterative sequence and the direction sequence generated by DFCG Algorithm respectively.

The following Lemma comes from [7].

Lemma 3.2. *There exists a Lipschitz constant $L > 0$ such that*

$$\|\nabla\varphi(a^1, b^1) - \nabla\varphi(a^2, b^2)\| \le L\|(a^1, b^1) - (a^2, b^2)\|$$

holds for all $(a^1, b^1), (a^2, b^2) \in R^2$.

Lemma 3.3. *There exist positive constants L_3, L_4 and L_5 such that for any $x, y \in N$, the following inequalities hold:*

$$\|\tilde{\Phi}(x) - \tilde{\Phi}(y)\| \le L_3\|x - y\|, \tag{3.1}$$

$$\|p(x) - p(y)\| \le L_4\|x - y\|, \tag{3.2}$$

$$\|\nabla\Psi(x) - \nabla\Psi(y)\| \le L_5\|x - y\|. \tag{3.3}$$

Proof. Since

$$\frac{\partial\varphi}{\partial a}(x_i, F_i(x)) = (p(x))_i,$$
$$\frac{\partial\varphi}{\partial b}(x_i, F_i(x)) = (\tilde{\Phi}(x))_i,$$
$$\nabla\varphi(x_i, F_i(x)) = ((p(x))_i, (\tilde{\Phi}(x))_i),$$

by Lemma 3.2 we have

$$\begin{aligned} |(\tilde{\Phi}(x) - \tilde{\Phi}(y))_i| &\leq \|\nabla\varphi(x_i, F_i(x)) - \nabla\varphi(y_i, F_i(y))\| \\ &\leq L\|(x_i, F_i(x)) - (y_i, F_i(y))\| \\ &\leq L(|x_i - y_i| + |F_i(x) - F_i(y)|). \end{aligned}$$

Combining this with the Lipschitz continuity of F we deduce that (3.1) holds. Similarly,

$$\begin{aligned} |(p(x) - p(y))_i| &\leq \|\nabla\varphi(x_i, F_i(x)) - \nabla\varphi(y_i, F_i(y))\| \\ &\leq L(|x_i - y_i| + |F_i(x) - F_i(y)|). \end{aligned}$$

We easily deduce that (3.2) holds.

Due to

$$\begin{aligned} & \nabla F(x) diag(b_i(x))\Phi_{\mathrm{FB}}(x) - \nabla F(y) diag(b_i(y))\Phi_{\mathrm{FB}}(y) \\ = \; & \nabla F(x)(diag(b_i(x))\Phi_{\mathrm{FB}}(x) - diag(b_i(y))\Phi_{\mathrm{FB}}(y)) + (\nabla F(x) - \nabla F(y)) diag(b_i(y))\Phi_{\mathrm{FB}}(y) \\ = \; & \nabla F(x)(\tilde{\Phi}(x) - \tilde{\Phi}(y)) + (\nabla F(x) - \nabla F(y)) diag(b_i(y))\Phi_{\mathrm{FB}}(y). \end{aligned}$$

In view of the Lipschitz continuity of F', the boundness of F', (3.1) and (3.2), we deduce that (3.3) holds. □

The following Lemma is straightforward from the steps of the DFCG Algorithm.

Lemma 3.4. *The sequence $\{\Psi(x_k)\}$ is strictly decreasing. In addition, the following inequalities hold:*

$$\sum_{k=0}^{\infty} \|\alpha_k d_k\|^2 < \infty, \quad \sum_{k=0}^{\infty} \|\alpha_k \Phi_{\mathrm{FB}}(x_k)\|^2 < \infty.$$

Lemma 3.5. *If there exists a constant $\epsilon > 0$ such that*

$$\|\nabla\Psi(x_k)\| \geq \epsilon \quad \forall k, \tag{3.4}$$

then there exists a constant $M > 0$ such that

$$\|d_k^{TT}\| \leq M \quad \forall k. \tag{3.5}$$

Proof. From (2.4) we can deduce

$$\|g_k\| \leq 2\|\Phi_{\mathrm{FB}}(x_k)\| + L_1\|\tilde{\Phi}(x_k)\| \leq (2L_1 + 2)\|\Phi_{\mathrm{FB}}(x_k)\| \leq (2L_1 + 2)p_0, \tag{3.6}$$

where $p_0 = \|\Phi_{\mathrm{FB}}(x_0)\|$. By (2.5), the following inequality holds:

$$\|d_k^{TT}\| \leq \| - g_k\| + 2\frac{\|g_k\|\|g_k - g_{k-1}\|\|d_{k-1}^{TT}\|}{\|g_{k-1}\|^2}. \tag{3.7}$$

By the mean-value theorem, there are $h_k, h_{k-1} \in (0,1)$ such that

$$\begin{aligned}\|q_k - q_{k-1}\| &= \|F'(x_k + h_k\lambda_k\tilde{\Phi}(x_k))\tilde{\Phi}(x_k) - F'(x_{k-1} + h_{k-1}\lambda_{k-1}\tilde{\Phi}(x_{k-1}))\tilde{\Phi}(x_{k-1})\| \\ &\leq \|F'(x_k + h_k\lambda_k\tilde{\Phi}(x_k))\tilde{\Phi}(x_k) - F'(x_k + h_k\lambda_k\tilde{\Phi}(x_k))\tilde{\Phi}(x_{k-1})\| \\ &\quad + \|F'(x_k + h_k\lambda_k\tilde{\Phi}(x_k))\tilde{\Phi}(x_{k-1}) - F'(x_{k-1} + h_{k-1}\lambda_{k-1}\tilde{\Phi}(x_{k-1}))\tilde{\Phi}(x_{k-1})\| \\ &\leq \gamma_2 L_3\|x_k - x_{k-1}\| + 2p_0L_2(\|\lambda_k\tilde{\Phi}(x_k)\| + \|\lambda_{k-1}\tilde{\Phi}(x_{k-1})\| + \|x_k - x_{k-1}\|).\end{aligned}$$

This implies

$$\begin{aligned}\|g_k - g_{k-1}\| &\leq \|p(x_k) - p(x_{k-1})\| + \|q_k - q_{k-1}\| \\ &\leq (L_4 + \gamma_2L_3 + 2p_0L_2)\|x_k - x_{k-1}\| \\ &\quad + 4p_0L_2(\|\lambda_k\Phi_{\rm FB}(x_k)\| + \|\lambda_{k-1}\Phi_{\rm FB}(x_{k-1})\|). \qquad (3.8)\end{aligned}$$

It follows from Lemma 3.4 that

$$\lim_{k\to\infty}\|\alpha_k\Phi_{\rm FB}(x_k)\| = 0.$$

If $\limsup_{k\to\infty}\alpha_k > 0$, then $\liminf_{k\to\infty}\|\Phi_{\rm FB}(x_k)\| = 0$. Hence $\liminf_{k\to\infty}\|\nabla\Psi(x_k)\| = 0$, which contradicts (3.4). Therefore $\lim_{k\to\infty}\alpha_k = 0$, $\lim_{k\to\infty}\lambda_k = 0$, and $\lim_{k\to\infty}\|g_k - \nabla\Psi(x_k)\| = 0$. Since $\|\nabla\Psi(x_k)\| \geq \epsilon, \forall k$, there exists an integer k_0 such that the following inequality holds for all $k \geq k_0$:

$$\|g_{k-1}\|^2 \geq \frac{1}{2}\epsilon^2. \qquad (3.9)$$

It then follows from (3.6)-(3.9) that

$$\begin{aligned}\|d_k^{TT}\| &\leq c_0 + (c_1\|x_k - x_{k-1}\| + c_2\|\lambda_k\Phi_{\rm FB}(x_k)\| + c_3\|\lambda_{k-1}\Phi_{\rm FB}(x_{k-1})\|)\|d_{k-1}^{TT}\| \\ &\leq c_0 + (c_1\|\alpha_{k-1}d_{k-1}^{TT}\| + c_2\|\alpha_k\Phi_{\rm FB}(x_k)\| + c_3\|\alpha_{k-1}\Phi_{\rm FB}(x_{k-1})\|)\|d_{k-1}^{TT}\|\end{aligned}$$

where c_0, c_1, c_2, c_3 are positive constants. By Lemma 3.4, there exist a constant $r \in (0,1)$ and an integer k_1 with $k_1 > k_0$ such that for any $k > k_1$,

$$c_1\|\alpha_{k-1}d_{k-1}^{TT}\| + c_2\|\alpha_k\Phi_{\rm FB}(x_k)\| + c_3\|\alpha_{k-1}\Phi_{\rm FB}(x_{k-1})\| \leq r,$$

$$\|d_k^{TT}\| \leq c_0 + r\|d_{k-1}^{TT}\| \leq c_0(1 + r + \cdots + r^{k-k_1-1}) + r^{k-k_1}\|d_{k_1}^{TT}\| \leq \frac{c_0}{1-r} + \|d_{k_1}^{TT}\|.$$

Letting $M = \max\{\|d_1^{TT}\|, \|d_2^{TT}\|, \ldots, \|d_{k_1}^{TT}\|, \frac{c_0}{1-r} + \|d_{k_1}^{TT}\|\}$, we get (3.5). □

The following theorem establishes the global convergence of Algorithm 1.

Theorem 3.6. *Let $\{x_k\}$ be generated by Algorithm* 1. *We have*

$$\liminf_{k\to\infty}\|\nabla\Psi(x_k)\| = 0. \qquad (3.10)$$

Proof. By Lemma 3.4, we have

$$\lim_{k\to\infty}\|\alpha_k\Phi_{\rm FB}(x_k)\| = 0.$$

If $\limsup_{k\to\infty}\alpha_k > 0$, then $\liminf_{k\to\infty}\|\Phi_{\rm FB}(x_k)\| = 0$. Hence $\liminf_{k\to\infty}\|\nabla\Psi(x_k)\| = 0$. Hence we only need to show (3.10) for the case $\lim_{k\to\infty}\alpha_k = 0$. For the sake of contradiction, we suppose that the conclusion is not true. Then there exists a constant $\varepsilon > 0$ such that

$$\|\nabla\Psi(x_k)\| \geq \varepsilon \quad \forall k.$$

It is easy to see from Procedure 2 that if $\alpha_k \neq 1$, then the following inequality holds

$$\Psi(x_k + \rho^{-1}\alpha_k d_k^{TT}) - \Psi(x_k) > -\sigma_1\|\rho^{-1}\alpha_k d_k^{TT}\|^2 - \sigma_2\|\rho^{-1}\alpha_k \Phi_{\mathrm{FB}}(x_k)\|^2. \tag{3.11}$$

By the mean-value theorem, there exists a constant $h_k \in (0,1)$ such that

$$\begin{aligned}
&\Psi(x_k + \rho^{-1}\alpha_k d_k^{TT}) - \Psi(x_k) = \rho^{-1}\alpha_k \nabla\Psi(x_k + h_k\rho^{-1}\alpha_k d_k)^T d_k^{TT} \\
&= \quad \rho^{-1}\alpha_k\nabla\Psi(x_k)^T d_k^{TT} + \rho^{-1}\alpha_k(\nabla\Psi(x_k + h_k\rho^{-1}\alpha_k d_k^{TT}) - \nabla\Psi(x_k))^T d_k^{TT} \\
&\leq \quad \rho^{-1}\alpha_k\nabla\Psi(x_k)^T d_k^{TT} + L_5\|\rho^{-1}\alpha_k d_k^{TT}\|^2.
\end{aligned}$$

Substituting the last inequality into (3.11), we get

$$\rho^{-1}\alpha_k(L_5 + \sigma_1)\|d_k^{TT}\|^2 + \sigma_2\rho^{-1}\alpha_k\|\Phi_{\mathrm{FB}}(x_k)\|^2 > -\nabla\Psi(x_k)^T d_k^{TT}.$$

Since Lemma 3.5 implies that $\{d_k^{TT}\}$ is bounded, the last inequality yields

$$\lim_{k\to\infty} -\nabla\Psi(x_k)^T d_k^{TT} = \lim_{k\to\infty} \|\nabla\Psi(x_k)\|^2 = 0.$$

This leads a contradiction. The proof is complete. □

The global convergence of Algorithm 2 can be obtained in a similar way. For completeness, we give a proof.

Theorem 3.7. *Let $\{x_k\}$ be generated by Algorithm* 2. *We have*

$$\liminf_{k\to\infty} \|\nabla\Psi(x_k)\| = 0. \tag{3.12}$$

Proof. From Procedure 2, we have

$$\lim_{k\to\infty} \|\alpha_k\Phi_{\mathrm{FB}}(x_k)\| = 0. \tag{3.13}$$

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0. \tag{3.14}$$

If $\limsup_{k\to\infty} \alpha_k > 0$, then (3.13) implies $\liminf_{k\to\infty} \|\Phi_{\mathrm{FB}}(x_k)\| = 0$. Hence (3.12) hold. We only need to show (3.12) for the case $\lim_{k\to\infty} \alpha_k = 0$. For the sake of contradiction, we suppose that the conclusion is not true. Then there exists a constant $\varepsilon > 0$ such that

$$\|\nabla\Psi(x_k)\| \geq \varepsilon \quad \forall k.$$

Since $\lim_{k\to\infty} \alpha_k = 0$, we have

$$\lim_{k\to\infty} \lambda_k = 0 \quad \text{and} \quad \lim_{k\to\infty} \|g_k - \nabla\Psi(x_k)\| = 0.$$

Hence there exists an integer k_0 such that the following inequality holds for all $k \geq k_0$:

$$\|g_{k-1}\|^2 \geq \epsilon^2. \tag{3.15}$$

It follows from (3.8) and Lemma 3.4 that

$$\lim_{k\to\infty} \|g_k - g_{k-1}\| = 0.$$

Combining this with (3.6) and (3.15) gives

$$\lim_{k\to\infty} \beta_k = 0.$$

Without loss of generality, we suppose $|\beta_k| \le \frac{1}{2}$ for all k. Since

$$(I - \frac{g_k g_k^T}{\|g_k\|^2})g_k = 0,$$

we get from (2.6)

$$\|d_k^{TM} + g_k\|^2 = \beta_k^2 \|(I - \frac{g_k g_k^T}{\|g_k\|^2})(g_k + d_{k-1}^{TM})\|^2 \le \beta_k^2 \|(g_k + d_{k-1}^{TM})\|^2.$$

It implies

$$\|d_k^{TM} + g_k\| \le |\beta_k| \|g_k + d_{k-1}^{TM}\| \le \frac{1}{2}(\|g_k - g_{k-1}\| + \|g_{k-1} + d_{k-1}^{TM}\|).$$

We claim from the last inequality that the sequence $\{\|d_k^{TM} + g_k\|\}$ is bounded and hence the sequence $\{\|d_k^{TM}\|\}$ is bounded.

If $\alpha_k = 1$, k is sufficiently large,

$$\|\alpha_k d_k^{TM}\| = \|d_k^{TM}\| \ge \|\nabla\Psi(x_k)\| \ge \varepsilon,$$

which contracts with (3.14).

If $\alpha_k \ne 1$, the following inequality holds:

$$\sigma_1(\alpha_k/\rho)^2 \|d_k^{TM}\|^2 + \sigma_2(\alpha_k/\rho)^2 \Psi(x_k) \ge \Psi(x_k) - \Psi(x_{k+1}). \tag{3.16}$$

When k is sufficiently large,

$$\begin{aligned} \Psi(x_{k+1}) - \Psi(x_k) &= \alpha_k \nabla\Psi(x_k + t\alpha_k d_k^{TM})^T d_k^{TM} \\ &= \alpha_k \nabla\Psi(x_k)^T d_k^{TM} + \alpha_k(\nabla\Psi(x_k + t\alpha_k d_k^{TM}) - \nabla\Psi(x_k))^T d_k^{TM} \\ &\le -\alpha_k \|\nabla\Psi(x_k)\|^2 + L_5 \|\alpha_k d_k^{TM}\|^2. \end{aligned} \tag{3.17}$$

From (3.16) and (3.17) we obtain

$$\alpha_k \|d_k^{TM}\| \ge \frac{\|\nabla\Psi(x_k)\|^2 \|d_k^{TM}\|}{(L_5 + \sigma_1/\rho^2)\|d_k^{TM}\|^2 + (\sigma_2/\rho^2)\Psi(x_k)},$$

Since $\{\|d_k^{TM}\|\}$ is bounded and when k is sufficiently large $\|d_k^{TM}\| \ge \|\nabla\Psi(x_k)\| \ge \varepsilon$, there exists a constant $c > 0$ such that when k is sufficiently large $\alpha_k \|d_k^{TM}\| \ge c$, which contradicts with (3.14). Therefore, (3.12) holds. □

4 Numerical Experiments

In this section, we report some preliminary numerical experiments. We implement our algorithm in fortran 90 and run the codes on a PC with 1.60 GHz CPU and 1.87 GB memory. The test problems come from MCPLIB and reference [1, 9, 10]. We replace the linear constraints of some problems in [10] and the box constraints in [1] with nonnegativity constraints on all of the variables. Then we get the Karush-Kuhn-Tucker (KKT) conditions which are symmetric nonlinear complementarity problems and named by MHS4, MHS5, MHS38, MHS59, MHS62, MHS71, MHS93, MHS99, BGRS1-4, respectively. The parameters of our algorithms have the values $\rho = 0.1, \sigma_1 = \sigma_2 = 10^{-5}$.

Details about the problems and the initial points are given in the Appendix. We use the inequality $\|\Phi_{\rm FB}(x)\| < 10^{-6}$ as the termination criterion for both algorithms. Tables 1-4 report the numbers of the iterations and function evaluations for both algorithms, where "$-$" denotes the failure of the algorithm. The column of the table has the following meaning.

Problem: the name of the problem;
Dim: the dimension of the problem;
SP: the initial point;
It1: the number of iterations for algorithm 1;
F1: the number of function evaluations for algorithm 1;
Time1: the CPU time in seconds for algorithm 1;
It2: the number of iterations for algorithm 2.
F2: the number of function evaluations for algorithm 2;
Time2: the CPU time in seconds for algorithm 2.

The numerical results show that there is no much difference between the performance of Algorithms 1 and 2 for small problems. Algorithm 1 performs better than Algorithm 2 does for large-scale problems. As the dimension increases, Algorithm 2 requires more CPU time than Algorithm 1 does, that is partly due to the fact that Algorithm 2 uses much CPU time to compute a large dimension matrix in determining direction d_k at iteration k.

Table 1. Test results

Problem	Dim	SP	It1	F1	Time1	It2	F2	Time2
Cycle	1	a	4	9	0	4	9	0
		b	6	13	0	6	13	0
Billups	1	a	18	91	0	18	91	0
		b	18	91	0	18	91	0
FFK	2	a	8	17	0	8	17	0
		b	9	19	0	9	19	0
MHS4	2	a	4	9	0	5	11	0
		b	5	11	0	12	25	0
		c	8	17	0	12	25	0
		d	6	13	0	12	25	0
MHS5	2	a	39	196	0	38	191	0
		b	41	200	0	40	193	0
		c	41	206	0	40	201	0
		d	43	212	0	41	206	0
Watson4	2	a	3	11	0	4	14	0
		b	1	3	0	1	3	0
		c	1	3	0	1	3	0
		d	1	3	0	1	3	0
MHS59	2	a	658	1317	0	659	1319	0
BGRS1	2	a	115	1036	0	77	636	0
		b	136	1261	0	57	492	0
MHS62	3	a	0	1	0	0	1	0
		b	2	5	0	5	11	0
		c	4	9	0	11	23	0
		d	4	9	0	9	19	0
MHS71	4	a	52	319	0	44	255	0
		b	67	362	0	89	518	0
MHS38	4	a	3034	43076	0.015625	2311	32175	0.015625
MHS93	6	a	473	1610	0	104	481	0
MHS99	7	a	1	3	0	1	3	0
		b	1	3	0	1	3	0
		c	1	3	0	1	3	0
		d	1	3	0	1	3	0
		e	1	3	0	1	3	0
		f	1	3	0	1	3	0
		g	2	8	0	2	8	0
		h	3	11	0	11	51	0

Table 2. Test results for problem BGRS2

SP	Dim	It1	F1	Time1	It2	F2	Time2
0.1	50	3	7	0	3	7	0
0.2	50	3	7	0	3	7	0
0.1	100	2	5	0	2	5	0
0.2	100	3	7	0	3	7	0
0.1	200	1	3	0	1	3	0
0.2	200	2	5	0	2	5	0
0.1	300	1	3	0	1	3	0
0.2	300	2	5	0	2	5	0
0.1	400	1	3	0	1	3	0
0.2	400	4	16	0	3	11	0.015625

Table 3. Test results for problem BGRS3

SP	Dim	It1	F1	Time1	It2	F2	Time2
0.1	2000	3	7	0	3	7	1.1875
0.2	2000	4	9	0	4	9	1.765625
0.3	2000	4	9	0	4	9	1.765625
20	2000	5	18	0	5	18	2.328125
30	2000	5	18	0	5	18	2.3125
0.1	5000	3	7	0	3	7	22.21875
0.2	5000	4	9	0	4	9	33.296875
0.3	5000	4	9	0	4	9	33.09375
20	5000	5	18	0.015625	5	18	44.078125
30	5000	5	18	0.015625	5	18	43.984375
0.1	10000	3	7	0.015625	3	7	95.875
0.2	10000	4	9	0.015625	4	9	143.28125
0.3	10000	4	9	0.015625	4	9	143.71875
20	10000	5	18	0.03125	5	18	191.140625
30	10000	5	18	0.03125	5	18	190.265625
0.1	100000	3	7	0.125	-	-	-
0.2	100000	4	9	0.15625	-	-	-
0.3	100000	4	9	0.15625	-	-	-
20	100000	5	18	0.296875	-	-	-
30	100000	5	18	0.296875	-	-	-
0.1	500000	4	9	0.828125	-	-	-
0.2	500000	4	9	0.828125	-	-	-
0.3	500000	5	11	1.015625	-	-	-
20	500000	5	18	1.515625	-	-	-
30	500000	5	18	1.53125	-	-	-

Table 4. Test results for problem BGRS4

SP	Dim	It1	F1	Time1	It2	F2	Time2
1	300	8	17	0	8	17	0.015625
1	500	8	17	0	8	17	0.078125
1	800	8	17	0	8	17	0.359375
1	1000	8	17	0	8	17	0.62500
1	2000	8	17	0	8	17	4.078125
1	5000	8	17	0.015625	8	17	74.890625
1	6000	8	17	0.015625	8	17	100.359375
1	7000	8	17	0.015625	8	17	164.46875
1	8000	8	17	0.015625	8	17	195.015625
1	10000	8	17	0.03125	8	17	349.109375
1	50000	8	17	0.140625	-	-	-
1	500000	8	17	1.484375	-	-	-
1	1000000	8	17	2.890625	-	-	-
1	2000000	8	17	5.71875	-	-	-
1	5000000	8	17	14.28125	-	-	-

Acknowledgements The author would like to thank two anonymous referees and the editor for their useful suggestions and comments which improved the paper greatly.

References

[1] E.G. Birgin, E.M. Gozzi, M.G.C. Resende and R.M.A. Silva, Continuous GRASP with a local active-set method for bound-constrained global optimization, *J. Global Optim.* 48 (2010) 289–319.

[2] J.S. Chen, The semismooth-related properties of a merit function and a descent method for the nonlinear complementarity problem, *J. Global Optim.* 36 (2006) 565–580.

[3] J.S. Chen and S. Pan, A family of NCP functions and a descent method for the nonlinear complementarity problem, *Comput. Optim. Appl.* 40 (2008) 389–404.

[4] W.Y. Cheng, A two term PRP based descent method, *Numer. Funct. Anal. Optim.* 28 (2007) 1217–1230.

[5] J.L. Dong and M. Q. Jiang, A modified modulus method for symmetric positive-definite linear complementarity problems, *Numer. Linear Algebra Appl.* 16 (2009) 129–143.

[6] A. Fischer, Solution of monotone complementarity problems with locally Lipschitz functions, *Math. Program.* 76 (1997) 513–532.

[7] C. Geiger and C. Kanzow, On the resolution of monotone complementarity problems, *Comput. Optim. Appl.* 5 (1996) 155–173.

[8] G. Gu, D.H. Li, L. Qi and S. Zhou, Descent directions of quasi-Newton methods for symmetric nonlinear equations, *SIAM J. Numer. Anal.* 40 (2003) 1763–1774.

[9] F. Facchinei, A. Fischer and C. Kanzow, On the accurate identification of active constraints, *SIAM J. Optim.* 9 (1999) 14–32.

[10] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, 187, Berlin-Heidelberg-New York: Springer-Verlag,1981.

[11] J.M. Iopes, Accelerating iterative algorithms for symmetric linear complementarity problems, *Int. J. Comput. Math.* 50 (1994) 35–44.

[12] H. Jiang, Unconstrained minimization approaches to nonlinear complementarity problems, *J. Global Optim.* 9 (1996) 169–181.

[13] C. Kanzow, N. Yamashita and M. Fukushima, New NCP functions and their properties, *J. Optim. Theory Appl.* 94 (1997) 115–135.

[14] D.H. Li and M. Fukushima, Smoothing Newton and quasi-Newton methods for mixed complementarity problems, *Comput. Optim. Appl.* 17 (2000) 203–230.

[15] D.H. Li, J. Wu and Z. Sun, Descent derivative-free methods for symmetric nonlinear equations, manuscript.

[16] D.H. Li and X.L. Wang, A modified Fletcher-Reeves-type derivative-free method for symmetric nonlinear equations, *Numer. Algebra, Control Optim.* 1 (2011) 71–82.

[17] Z.Q. Luo and P. Tseng, A new class of merit functions for the nonlinear complementarity problem, in *Complementarity and Variational Problems: State of the Art*, M.C. Ferris and J.S. Pang (eds.), SIAM Publications, Philadelphia, 1997, pp. 204-225.

[18] O.L. Mangasarian and M.V. Solodov, A linearly convergent derivative-free descent method for strongly monotone complementarity problems, *Comput. Optim. Appl.* 5 (1999) 1–13.

[19] O.L. Mangasarian, Solution of symmetric linear complementarity problems by iterative methods, *J. Optim. Theory Appl.* 22 (1977) 465–485.

[20] N. Machida, M. Fukushima and T. Ibaraki, A multisplitting method for symmetric linear complementarity problems, *J. Comput. Appl. Math.* 62 (1995) 217–227.

[21] R. Mifflin, Semismooth and semiconvex functions in constrained optimization, *SIAM J. Control Optim.* 15 (1977) 957–972.

[22] J.L. Morales, J. Nocedal and M. Smelyanskiy, An algorithm for the fast solution of symmetric linear complementarity problems, August 23, 2008.

[23] J.M. Peng, Derivative-free methods for monotone variational inequality and complementarity problems, *J. Optim. Theory Appl.* 99 (1998) 235–252.

[24] A.R. Pierro, Nonlinear relaxation methods for solving symmetric linear complementarity problems, *J. Optim. Theory Appl.* 64 (1990) 87–99.

[25] L. Qi and J. Sun, A nonsmooth version of Newton's method, *Math. Program.* 58 (1993) 353–367.

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

[27] K. Yamada, N. Yamashita and M. Fukushima, A new derivative-free descent method for the nonlinear complementarity problems, in *Nonlinear Optimization and Related Topics*, G.D. Pillo and F. Giadnessi (eds.), Kluwer Academic Publishers, Dordrecht, 2000, pp. 463–489.

[28] J. Zhang and D.H. Li, A norm descent BFGS method for solving KKT systems of symmetric variational inequality problems, *Optim. Methods Softw.* 22 (2007) 237–252.

[29] L. Zhang, W.J. Zhou and D.H. Li, A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence, *IMA J. Numer. Anal.* 26 (2006) 629–640.

Appendix The Test Problems and Initial Points

1. Problem Cycle starting point: (a) 1, (b) 2.
2. Problem Billups starting point: (a)2, (b) 2.01.
3. Problem FFK

$$F_1(x) = 2x_1 + 4x_2,$$
$$F_2(x) = 2x_2 + 4x_1.$$

Starting points: (a) $(1,1)$, (b) $(2,2)$.

4. Problem MHS4

$$F_1(x) = (x_1+1)^2,$$
$$F_2(x) = 1.$$

Starting points: (a) $(0.125, 0.125)$, (b) $(1.125, 0.125)$, (c) $(1,1)$, (d) $(0.5, 0.5)$.

5. Problem MHS5

$$F_1(x) = \cos(x_1+x_2) + x_1 - x_2 - 1.5,$$
$$F_2(x) = \cos(x_1+x_2) - x_1 + x_2 + 2.5.$$

Starting point: (a) $(100,100)$, (b) $(3,3)$, (c) $(10,10)$, (d) $(50,50)$.

6. Watson4 Starting point: (a) $(2.4, 2.4)$, (b) $(2.5, 2.5)$, (c) $(3.0, 3.0)$, (d) $(3.5, 3.5)$.
7. Problem MHS59

$$\begin{aligned} F(1) = \; & 3.8112 + 0.0020567 \times 3x_1^2 - 4 \times 1.0345 \times 10^{-5} \times x_1^3 \\ & -0.030234 \times x_2 + 2 \times 1.28134 \times 10^{-3} \times x_1 x_2 \\ & +2.266 \times 10^{7} \times 4x_1^3 x_2 + 5.2375 \times 10^{-6} \times 2x_1 x_2^2 \\ & +6.3 \times 10^{-8} \times 3x_1^2 x_2^2 - 7.0 \times 10^{-10} \times 3x_1^2 x_2^3 \\ & -3.405 \times 10^{-4} \times x_2^2 + 1.6638 \times 10^{-6} \times x_2^3 \\ & +0.0005 \times 2.8673 \times x_2 \times e^{0.0005 x_1 x_2} - 3.5256 \times 10^{-5} \times 3x_1^2 x_2, \\ F(2) = \; & 6.8306 - 0.030234 \times x_1 + 1.28134 \times 10^{-3} x_1^2 + 2.266 \times 10^{-7} x_1^4 \\ & -0.25645 \times 2x_2 + 0.0034604 \times 3x_2^2 - 1.3514 \times 10^{-5} \times 4x_2^3 - \tfrac{28.106}{(x_2+1)^2} \\ & +5.2375 \times 10^{-6} \times 2x_1^2 x_2 + 2 \times 6.3 \times 10^{-8} \times x_1^3 x_2 - 7.0 \times 10^{-10} \times 3x_1^3 x_2^2 \\ & -2 \times 3.405 \times 10^{-4} \times x_1 x_2 + 3 \times 1.6638 \times 10^{-6} \times x_1 x_2^2 \\ & +0.0005 \times 2.8673 \times x_1 \times e^{0.0005 x_1 x_2} - 3.5256 \times 10^{-5} x_1^3. \end{aligned}$$

Starting point: (a) $(20,20)$.

8. Problem BGRS1

$$f(1) = 2x_1^3 + 2x_1x_2 + x_2^2 - 21x_1 - 7,$$
$$f(2) = 2x_2^3 + 2x_1x_2 + x_1^2 + x_2 - 25.$$

Starting point: (a) $(3,1)$, (b) $(5,6)$.

9. Problem MHS62

$$\begin{aligned} F(1) = &\ \tfrac{1}{x_1+x_2+x_3+0.03} - \tfrac{0.09}{0.09x_1+x_2+x_3+0.03}, \\ F(2) = &\ \tfrac{255}{x_1+x_2+x_3+0.03} - \tfrac{255}{0.09x_1+x_2+x_3+0.03} + \tfrac{280}{x_2+x_3+0.03} - 280 \times \tfrac{0.07}{0.07x_2+x_3+0.03}, \\ F(3) = &\ \tfrac{255}{x_1+x_2+x_3+0.03} - \tfrac{255}{0.09x_1+x_2+x_3+0.03} + \tfrac{280}{x_2+x_3+0.03} - \tfrac{280}{0.07x_2+x_3+0.03} \\ &+ \tfrac{290}{x_3+0.03} - 290 \times \tfrac{0.13}{0.13x_3+0.03}. \end{aligned}$$

Starting point: (a) (0,0,0), (b) (0.01,0.02,0.03), (c) (0.04,0.04,0.04), (d)(0.05,0.05,0.05).

10. Problem MHS71

$$\begin{aligned} F(1) &= (2x_1 + x_2 + x_3)x_4, \\ F(2) &= x_1x_4, \\ F(3) &= x_1x_4 + 1, \\ F(4) &= x_1(x_1 + x_2 + x_3). \end{aligned}$$

Starting point: (a) (3,3,2,1), (b) (3,1,4,2).

11. Problem MHS38

$$\begin{aligned} F(1) &= -400(x_2 - x_1^2)x_1 + 2(x_1 - 1), \\ F(2) &= 200(x_2 - x_1^2) + 20.2(x_2 - 1) + 19.8(x_4 - 1), \\ F(3) &= -360(x_4 - x_3^2)x_3 + 2(x_3 - 1), \\ F(4) &= 180(x_4 - x_3^2) + 20.2(x_4 - 1) + 19.8(x_2 - 1). \end{aligned}$$

Starting point: (a) (0.5,0.5,0.5,0.5).

12. Problem MHS93

$$\begin{aligned} F(1) &= 2x_1(0.0204 + 0.0607x_5^2)x_4 + (0.0187 + 0.0437x_6^2)x_3x_2, \\ F(2) &= x_1(0.0204 + 0.0607x_5^2)x_4 + (0.0187 + 0.0437x_6^2)x_3(x_1 + 2 \times 1.57x_2), \\ F(3) &= x_1(0.0204 + 0.0607x_5^2)x_4 + (0.0187 + 0.0437x_6^2)x_2(x_1 + 1.57x_2 + x_4), \\ F(4) &= x_1(0.0204 + 0.0607x_5^2)(x_1 + x_2 + x_3) + (0.0187 + 0.0437x_6^2)x_2x_3, \\ F(5) &= x_1x_4x_5(x_1 + x_2 + x_3), \\ F(6) &= x_2x_3x_6(x_1 + 1.57x_2 + x_4). \end{aligned}$$

Starting point: (a) $(4.2, \cdots, 4.2)$.

13. Problem MHS99

$$F(i) = -2\sum_{i=1}^{n}(t_{i+1} - t_i)\cos(x_i) \times a_i \times (t_{i+1} - t_i) \times (-\sin(x_i)), \quad i = 1, \cdots, 7,$$

where $a_1 = a_2 = 50$, $a_3 = a_4 = a_5 = 75$, $a_6 = a_7 = 100$, $t_1 = 0$, $t_2 = 25$, $t_3 = 50$, $t_4 = 100$, $t_5 = 150$, $t_6 = 200$, $t_7 = 290$, $t_8 = 380$.

Starting point: (a) $(0.1, \cdots, 0.1)$, (b) $(0.2, \cdots, 0.2)$, (c) $(0.3, \cdots, 0.3)$, (d) $(0.5, \cdots, 0.5)$, (e) $(1.0, \cdots, 1.0)$, (f) $(1.3, \cdots, 1.3)$, (g) $(1.4, \cdots, 1.4)$, (h) $(1.5, \cdots, 1.5)$.

14. Problem BGRS2

$$F(i) = 0.4x_i\frac{e^{-0.2\sqrt{0.1\sum_{i=1}^{n}x_i^2}}}{\sqrt{0.1\sum_{i=1}^{n}x_i^2}} + 0.2\pi\sin(2\pi x_i)e^{0.1\sum_{i=1}^{n}\cos(2\pi x_i)}, \quad i = 1, \cdots, n.$$

15. Problem BGRS3

$$
\begin{aligned}
&F(1) = 10\pi \sin(2\pi x_1) - 2(x_1 - 1)(1 + 10 \sin^2(\pi x_2)),\\
&F(i) = -10\pi(x_{i-1} - 1)^2 \sin(2\pi x_i) - 2(x_i - 1)(1 + 10 \sin^2(\pi x_{i+1})), \quad i = 2, \cdots, n-1,\\
&F(n) = -10\pi(x_{n-1} - 1)^2 \sin(2\pi x_n) - 2(x_n - 1).
\end{aligned}
$$

16. Problem BGRS4

$$
\begin{aligned}
&F(1) = 3\pi \sin(6\pi x_1) + 2(x_1 - 1)(1 + \sin^2(3\pi x_2)),\\
&F(i) = 3\pi(x_{i-1} - 1)^2 \sin(6\pi x_i) + 2(x_i - 1)(1 + \sin^2(3\pi x_{i+1})), \quad i = 2, \cdots, n-1,\\
&F(n) = 3\pi(x_{n-1} - 1)^2 \sin(6\pi x_n) + 1 + \sin^2(2\pi x_n) + 2\pi(x_n - 1)\sin(4\pi x_n).
\end{aligned}
$$

Manuscript received 10 February 2012
revised 1 August 2012
accepted for publication 25 October 2012

Qiong Li
College of Science, China Three Gorges University
Yichang, Hubei, 443002, P. R. China
E-mail address: liqiongmanj@yahoo.com