RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2017.43.195

A derivative-free projection method for solving convex constrained monotone equations

Na Yuan

College of General Education, Chongqing College of Electronic Engineering, Chongqing 401331 China

e-mail: yuanna_math@126.com

Received 9 Dec 2016
Accepted 2 Jun 2017

ABSTRACT: The CG_DESCENT (CGD) method is one of the most efficient conjugate gradient methods for solving unconstrained optimization problems. However, its applications in some other scenarios are relatively few. In this paper, inspired by one spectral PRP projection method, we extend the CGD method, and establish a derivative-free spectral CGD type projection method to solve large-scale nonlinear monotone equations with convex constraints. Due to it inheriting some nice properties of the conjugate gradient method such as the low memory requirement, the proposed method is very suitable to solve large-scale nonlinear monotone equations. Under appropriate conditions, we prove that the proposed method is globally convergent. Preliminary numerical results show that the proposed method works well.

KEYWORDS: conjugate gradient method, projection technique, global convergence

MSC2010: 90C30 65K05

INTRODUCTION

In this paper, we focus on solving the nonlinear monotone equations

$$F(x)=0,\ x\in\Omega, \tag{1}$$

where $F\colon\ \Omega\to R^n$ is continuous and monotone, and $\Omega\subseteq R^n$ is a nonempty closed convex set. The monotonicity of F means that

$$\langle F(x)-F(y), x-y\rangle \geqslant 0, \quad \forall x,y\in\Omega. \tag{2}$$

Nonlinear monotone equations have many practical applications such as some monotone variational inequality problems[1], economic equilibrium problems[2], and chemical equilibrium systems[3]. Even now, a lot of computational methods have been proposed to solve nonlinear monotone equations. Among these methods, Newton's method, the quasi-Newton method, and their variants are very popular because of their local superlinear convergence. However, these methods are not suitable for large-scale nonlinear monotone equations because they need to solve a linear system of equations using the Jacobian matrix of $F(x)$ or an approximation of the Jacobian matrix at each iteration.

It is well-known that the spectral gradient method is an efficient method to solve large-scale unconstrained optimization problems because of its simplicity and low storage requirements. Thus some people have extended some spectral gradient methods to solve large-scale nonlinear monotone equations. Recently, by combining the projection technique of Solodov and Svaiter[4] and the spectral gradient method of Barzilai and Borwein[5], Zhang and Zhou[6] proposed a derivative-free spectral gradient projection method to solve unconstrained monotone equations in which the search direction is defined as

$$d_0=-F_0,\ d_k=-\theta_k F_k,$$

where $\theta_k=s_{k-1}^{\mathrm{T}}s_{k-1}/s_{k-1}^{\mathrm{T}}y_{k-1},\ y_{k-1}=F_k-F_{k-1}+rs_{k-1}, s_{k-1}=x_k-x_{k-1}, r>0$. The proposed method is shown to be globally convergent to a solution of the equations if the nonlinear equations to be solved are monotone and Lipschitz continuous. Yu et al[7] used the above method to solve convex constrained nonlinear monotone equations. They proved that the proposed method is globally convergent under mild assumptions. Preliminary numerical results show that the proposed method works quite well for large-scale problems. Most recently, based on the projection technique[4], the spectral gradient method[5] and the classical PRP conjugate gradient method[8], Liu[9] proposed a derivative-free spectral PRP projection method (SPRP) to solve nonlinear monotone equations with convex constraints in which the search direction is computed from

$$d_0 = -F_0,\ d_k = -\theta_k F_k + \beta_k^{\mathrm{PRP}} d_{k-1}.$$

Here $\beta_k^{\mathrm{PRP}} = F_k^{\mathrm{T}} y_{k-1}/\|F_{k-1}\|^2$, and the definitions of θ_k and y_{k-1} are the same as those in Ref. 6. The global convergence of the SPRP method is established with some suitable conditions. Because of the derivative-free feature and lower storage requirement, the SPRP method is very suitable to solve large-scale non-smooth nonlinear monotone equations. Numerical results also show that the SPRP method is more efficient and robust than the method in Ref. 7.

In this paper, motivated by the idea of Liu[9], we combine the famous CG_DESCENT (CGD) method[10] and the spectral gradient method[5] with the projection technique[4], and construct a derivative-free spectral CGD projection method for solving large-scale convex constrained monotone equations in which the Jacobian matrix of F is not available or requires a large amount of storage. The proposed method is suitable for solving large-scale nonlinear monotone equations because it inherits some nice properties of the spectral conjugate gradient and CGD methods such as the low memory requirement and the high efficiency. We prove its global convergence under some appropriate conditions. By comparing with the SPRP method[9], numerical results show that the proposed method is very efficient.

The paper is organized as follows. In the next section, we introduce the CGD method and its variants, and establish the new algorithm with the backtracking line search. In the third section, some assumptions and lemmas are proposed, and the global convergence of the proposed method is proved. Preliminary numerical results are presented in the final section.

ALGORITHM

In this section, we first introduce the CGD method with its variants, some basic concepts, and properties of the projection operator. Then we give the specific steps of the proposed method.

The CGD method is one of the most efficient conjugate gradient methods for solving unconstrained optimization problems. Its search direction is defined as

$$d_0 = -g_0,\ d_k = -g_k + \beta_k d_{k-1},$$

where g_k is the gradient of the objective function f at x_k, $\beta_k = \max\{\beta_k^{\mathrm{HZ}}, \eta_k\}$. Here

$$\beta_k^{\mathrm{HZ}} = \frac{1}{d_{k-1}^{\mathrm{T}} y_{k-1}} \left(y_{k-1} - t d_{k-1} \frac{\|y_{k-1}\|^2}{d_{k-1}^{\mathrm{T}} y_{k-1}} \right)^{\mathrm{T}} g_k,$$

$$\eta_k = \frac{-\lambda}{\|d_k\| \min\{\eta, \|g_{k-1}\|\}},$$

and $y_{k-1} = g_k - g_{k-1}$, $\lambda > 0$, $\eta > 0$, $t = 2$. The global convergence of the CGD method is proved under the standard Wolfe line search. Numerical results also show that the CGD method, applying the appropriate strong Wolfe line search, is more efficient than the PRP method. Subsequently, based on the spectral scaling secant equation proposed by Cheng and Li[11], Liu et al[12] further studied some conjugate gradient methods in the Hager-Zhang family[10], and showed that the CGD method with $t = 1$ is more efficient than that with $t = 2$. Furthermore, Dai and Kou[13] used the spectral technique to analyse the CGD method, and suggested that the CGD method with $t = 1$ is better than that with $t = 2$.

Let Ω be a nonempty closed convex subset of R^n. For any $x \in \mathbb{R}^n$, its projection onto Ω is defined as

$$P_\Omega(x) = \arg\min\{\|x - y\| \mid y \in \Omega\}.$$

The mapping $P_\Omega : R^n \longrightarrow \Omega$ is a projection operator. For any $x, y \in \mathbb{R}^n$, it holds that

$$\|P_\Omega(x) - P_\Omega(y)\| \leqslant \|x - y\|. \tag{3}$$

We now describe the specific steps of the derivative-free spectral CGD projection method. In this paper, we select the parameter $t = 1$ in the algorithm. For convenience, we abbreviate $F(x_k)$ by F_k.

Algorithm 1

Step 1: Give an initial point $x_0 \in \mathbb{R}^n$, $\rho \in (0,1)$, $\sigma > 0$, $r \in (0,1)$.

Step 2: Set $d_0 = -F_0$. Set $k := 0$.

Step 3: Determine the step-size $\alpha_k = \max\{\rho^i \mid i = 0, 1, 2, \dots\}$ is by the following line search:

$$-\langle F(z_k), d_k \rangle \geqslant \sigma \alpha_k \|F(z_k)\| \cdot \|d_k\|^2, \tag{4}$$

where $z_k = x_k + \alpha_k d_k$.

Step 4: Obtain the next iterative point x_{k+1} by

$$x_{k+1} = P_\Omega[x_k - \lambda_k F(z_k)], \tag{5}$$

where $\lambda_k = F(z_k)^{\mathrm{T}}(x_k - z_k)/\|F(z_k)\|^2$.

Step 5: If $F_{k+1} = 0$, stop. Otherwise, generate the next search direction d_{k+1} by

$$d_{k+1} = -\theta_{k+1} F_{k+1} + \beta_{k+1} s_k. \tag{6}$$

Here

$$\beta_{k+1} = \frac{1}{s_k^{\mathrm{T}} w_k}\left(w_k - \frac{\|w_k\|^2}{s_k^{\mathrm{T}} w_k} s_k\right)^{\mathrm{T}} F_{k+1},$$
$$\theta_{k+1} = \frac{s_k^{\mathrm{T}} s_k}{s_k^{\mathrm{T}} w_k},$$

where

$$w_k = y_k + r s_k, \quad s_k = x_{k+1} - x_k,\ y_k = F_{k+1} - F_k.$$

Step 6: Set $k := k+1$ and go to step 3.

Remark 1 From the definitions of w_k and s_k, we have

$$s_k^{\mathrm{T}} w_k = s_k^{\mathrm{T}} y_k + r s_k^{\mathrm{T}} s_k \geqslant r\|s_k\|^2 > 0,$$

where the inequality follows from the monotonicity of F. This means that the denominators of β_{k+1} and θ_{k+1} are always greater than zero before the algorithm stops.

GLOBAL CONVERGENCE

In this section, we assume that mapping F satisfies the following assumption.
Assumption H The mapping $F : R^n \longrightarrow R^n$ is Lipschitz continuous, i.e., there exists a constant $L > 0$ such that

$$\|F(x) - F(y)\| \leqslant L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \qquad (7)$$

The remainder of this section is devoted to providing some interesting lemmas and one theorem. To prove the following lemma, we assume that $L < 4 - r$ holds, where r is defined in Step 1.

Lemma 1 *Let the sequences $\{d_k\}$ and $\{F_k\}$ be generated by Algorithm 1. Then we have*

$$F_k^{\mathrm{T}} d_k \leqslant -c\|F_k\|^2, \quad \forall k \geqslant 0, \qquad (8)$$

where $c = \min\{1, 1/(L+r) - \frac{1}{4}\}$.

Proof: From the definition of w_{k-1} in Step 5 and (7), we have

$$\begin{aligned} s_{k-1}^{\mathrm{T}} w_{k-1} &\leqslant \|s_{k-1}\| \cdot \|w_{k-1}\| \\ &\leqslant \|s_{k-1}\|(\|F_k - F_{k-1}\| + r\|s_{k-1}\|) \\ &\leqslant (L+r)\|s_{k-1}\|^2. \end{aligned}$$

This means that

$$\theta_k \geqslant \frac{1}{L+r}. \qquad (9)$$

For $k \geqslant 1$, by taking the inner product of (6) with the vector F_k, it follows from (9) that

$$\begin{aligned} F_k^{\mathrm{T}} d_k &= -\theta_k \|F_k\|^2 + \beta_k F_k^{\mathrm{T}} s_{k-1} \\ &\leqslant \frac{F_k^{\mathrm{T}} w_{k-1} s_{k-1}^{\mathrm{T}} w_{k-1} F_k^{\mathrm{T}} s_{k-1} - \|w_{k-1}\|^2 (F_k^{\mathrm{T}} s_{k-1})^2}{(s_{k-1}^{\mathrm{T}} w_{k-1})^2} \\ &\qquad - \frac{1}{L+r}\|F_k\|^2. \end{aligned}$$

Let $a = (s_{k-1}^{\mathrm{T}} w_{k-1})/\sqrt{2}F_k^{\mathrm{T}}, b = \sqrt{2}(F_k^{\mathrm{T}} s_{k-1}) w_{k-1}$. It follows from $a \cdot b \leqslant \frac{1}{2}(a^2 + b^2)$ that

$$F_k^{\mathrm{T}} d_k \leqslant -\frac{1}{L+r}\|F_k\|^2 + \tfrac{1}{4}\|F_k\|^2.$$

From Step 2, for $k = 0$, $F_0^{\mathrm{T}} d_0 = -\|F_0\|^2$. Thus (8) holds. □

Lemma 2 *Suppose that Assumption H holds. Then there exists a step-size α_k satisfying the line search (4) for any $k \geqslant 0$.*

Proof: Assume that there exists a constant $k_0 \geqslant 0$ such that (4) is not satisfied for any nonnegative integer i, i.e.,

$$-\langle F(x_{k_0} + \rho^i d_{k_0}), d_{k_0}\rangle < \sigma\rho^i \|F(x_k + \rho^i d_{k_0})\| \cdot \|d_{k_0}\|^2.$$

From the continuity of the mapping F, let $i \to \infty$. We have

$$-\langle F(x_{k_0}), d_{k_0}\rangle < 0.$$

It follows from (8) that

$$-F(x_{k_0})^{\mathrm{T}} d_{k_0} \geqslant 0.$$

The above two inequalities are contradictory. Thus the assumption does not hold, which means that the line search (4) is always well posed. □

Lemma 3 *Suppose that Assumption H holds. Let the sequences $\{x_k\}$ and $\{z_k\}$ be generated by Algorithm 1, and $\bar{x} \in \Omega$ is an arbitrary solution of (1). Then*

$$\|x_{k+1} - \bar{x}\|^2 \leqslant \|x_k - \bar{x}\|^2 - \sigma^2\|x_k - z_k\|^4. \qquad (10)$$

Proof: From the monotonicity of the mapping F, it holds that

$$\langle F(z_k) - F(\bar{x}), z_k - \bar{x}\rangle \geqslant 0.$$

Then

$$\langle F(z_k) - F(\bar{x}), x_k - \bar{x}\rangle \geqslant -\langle F(z_k) - F(\bar{x}), z_k - x_k\rangle.$$

From the definition of z_k and (4), we have

$$\begin{aligned}\langle F(z_k), x_k - \bar{x}\rangle &\geqslant -\langle F(z_k), z_k - x_k\rangle \\ &= \alpha_k^2 \sigma \|F(z_k)\| \cdot \|d_k\|^2 \geqslant 0. \quad (11)\end{aligned}$$

From (5) we have

$$\begin{aligned}\|x_{k+1} - \bar{x}\|^2 &= \|P_\Omega[x_k - \lambda_k F(z_k)] - P_\Omega[\bar{x}]\|^2 \\ &\leqslant \|x_k - \lambda_k F(z_k) - \bar{x}\|^2 \\ &= \|x_k - \bar{x}\|^2 - 2\lambda_k F(z_k)^{\mathrm{T}}(x_k - \bar{x}) + \lambda_k^2 \|F(z_k)\|^2 \\ &\leqslant \|x_k - \bar{x}\|^2 - \frac{F(z_k)^{\mathrm{T}}(x_k - z_k) \cdot F(z_k)^{\mathrm{T}}(x_k - z_k)}{\|F(z_k)\|^2} \\ &\leqslant \|x_k - \bar{x}\|^2 - \alpha_k^4 \sigma^2 \|d_k\|^4 \\ &= \|x_k - \bar{x}\|^2 - \sigma^2 \|x_k - z_k\|^4,\end{aligned}$$

where the first inequality follows from (3), the second inequality follows from (11) and the definition of λ_k, the last inequality follows from (4). □

Remark 2 From (10) we have

$$\sigma^2 \|x_k - z_k\|^4 \leqslant \|x_k - \bar{x}\|^2 - \|x_{k+1} - \bar{x}\|^2.$$

It is not difficult to show that

$$\sum_{k=0}^{+\infty} \sigma^2 \|x_k - z_k\|^4 \leqslant \|x_0 - \bar{x}\|^2 < +\infty,$$

which means that

$$\lim_{k\to\infty} \|x_k - z_k\| = 0. \quad (12)$$

Theorem 1 *Suppose that Assumption H holds. Let the sequence $\{F_k\}$ be generated by Algorithm 1. Then we have*

$$\liminf_{k\to\infty} \|F_k\| = 0. \quad (13)$$

Proof: We assume that (13) does not hold, i.e., there exists a constant $u > 0$ such that

$$\|F_k\| \geqslant u, \quad \forall k \geqslant 0. \quad (14)$$

It follows from the definition of w_k and (7) that

$$\|w_{k-1}\| \leqslant \|F_k - F_{k-1}\| + r\|s_{k-1}\| \leqslant (r + L)\|s_{k-1}\|.$$

From Remark 1 and the definition of β_k,

$$\begin{aligned}|\beta_k| &\leqslant \frac{\|F_k\| \cdot \|w_{k-1}\|}{s_{k-1}^{\mathrm{T}} w_{k-1}} + \frac{\|w_{k-1}\|^2 \cdot \|F_k\| \cdot \|s_{k-1}\|}{(s_{k-1}^{\mathrm{T}} w_{k-1})^2} \\ &\leqslant \left(1 + \frac{L}{r}\right)\frac{\|F_k\|}{\|s_{k-1}\|} + \frac{(L+r)^2 \cdot \|F_k\|}{r^2 \|s_{k-1}\|} \\ &= \left(2 + \frac{3L}{r} + \frac{L^2}{r^2}\right)\frac{\|F_k\|}{\|s_{k-1}\|}.\end{aligned}$$

Then from Remark 1 we have

$$\|d_k\| \leqslant \frac{1}{r}\|F_k\| + |\beta_k| \cdot \|s_{k-1}\| \leqslant \nu \|F_k\|, \quad (15)$$

where $\nu = 2 + (3L + 1)/r + L^2/r^2$. It follows from (4) that $\rho^{-1}\alpha_k$ satisfies

$$-\langle F(\widetilde{z_k}), d_k\rangle < \sigma \rho^{-1} \alpha_k \|F(\widetilde{z_k})\| \cdot \|d_k\|^2,$$

where $\widetilde{z_k} = x_k + \rho^{-1}\alpha_k d_k$. From (8) and (7) we have

$$\begin{aligned}c\|F_k\|^2 &\leqslant -F_k^{\mathrm{T}} d_k \\ &= (F(\widetilde{z_k}) - F_k)^{\mathrm{T}} d_k - F(\widetilde{z_k})^{\mathrm{T}} d_k \\ &\leqslant \rho^{-1} L \alpha_k \|d_k\|^2 + \rho^{-1} \sigma \alpha_k \|F(\widetilde{z_k})\| \cdot \|d_k\|^2,\end{aligned}$$

i.e., $\alpha_k \|d_k\| \geqslant c\rho \|F_k\|^2 / (L + \sigma\|F(\widetilde{z_k})\|)\|d_k\|$. It follows from (14) and (15) that

$$\alpha_k \|d_k\| \geqslant \frac{c\rho u}{\nu(L + \sigma\|F(\widetilde{z_k})\|)} > 0. \quad (16)$$

From (12) and the definition of z_k,

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0.$$

This result contradicts (16). Thus the assumption does not hold, which means that (13) holds. □

NUMERICAL RESULTS

In this section, we compare Algorithm 1 with the SPRP method, an efficient derivative-free projection method[9] based on the PRP conjugate gradient method and spectral gradient method for solving nonlinear monotone equations with convex constraints. The parameters in Algorithm 1 are $\rho = 0.5$, $\sigma = 0.01$, $r = 0.001$. The parameters in the SPRP method come from Ref. 9. All algorithms terminate whenever $\|F_k\| \leqslant 10^{-5}$ or the total number of iterates exceeds 100 000. All presented codes are written in MATLAB 7.0, and run on a PC with 3.30 GHz CPU, 8.00 GB memory, and Windows 7 operation system. The tested problems with different initial starting points and various dimensions are as follows.

Example 1 The function F is defined as

$$F(x) = x - \sin x,$$

where $x \in \Omega = \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i \leqslant n, x_i \geqslant -1, i = 1, 2, \dots, n\}$.

Table 1 The numerical results of Example 1 with the given initial points.

	dim	Algorithm 1		SPRP	
		niter	CPU	niter	CPU
x0	5000	337	0.17	616	0.30
	10 000	424	0.38	777	0.65
	20 000	534	0.87	980	1.58
x1	5000	347	0.17	625	0.30
	10 000	434	0.40	785	0.65
	20 000	544	0.87	989	1.61
x2	5000	347	0.15	625	0.30
	10 000	434	0.37	785	0.66
	20 000	544	0.91	989	1.64
x3	5000	337	0.17	616	0.28
	10 000	424	0.36	777	0.66
	20 000	534	0.86	980	1.59
x4	5000	66	0.05	111	0.09
	10 000	66	0.08	111	0.14
	20 000	66	0.14	111	0.22
x5	5000	342	0.15	618	0.30
	10 000	429	0.48	780	0.67
	20 000	538	0.89	982	1.62

Table 2 The numerical results of Example 2 with the given initial points.

	dim	Algorithm 1		SPRP	
		niter	CPU	niter	CPU
x0	5000	4	0.03	14	0.08
	10 000	4	0.03	14	0.16
	20 000	4	0.05	15	0.23
x1	5000	4	0.02	14	0.08
	10 000	4	0.03	15	0.16
	20 000	4	0.04	15	0.22
x2	5000	5	0.03	14	0.11
	10 000	5	0.05	14	0.13
	20 000	5	0.04	15	0.23
x3	5000	4	0.03	14	0.08
	10 000	4	0.04	14	0.11
	20 000	5	0.05	15	0.22
x4	5000	4	0.03	14	0.09
	10 000	4	0.04	14	0.13
	20 000	4	0.05	15	0.22
x5	5000	5	0.04	14	0.09
	10 000	5	0.04	14	0.13
	20 000	5	0.05	15	0.23

Example 2 The elements of function F are given as

$$F_1(x) = x_1 - \exp\left(\cos\frac{x_1 + x_2}{n+1}\right),$$

$$F_i(x) = x_i - \exp\left(\cos\frac{x_{i-1} + x_i + x_{i+1}}{n+1}\right),$$

$$F_n(x) = x_n - \exp\left(\cos\frac{x_{n-1} + x_n}{n+1}\right),$$

where $x \in \Omega = R^n_+$, $i = 2, 3, \dots, n-1$.

Example 3 The elements of function F are given as

$$F_i(x) = \sqrt{10^{-5}}(x_i - 1),$$

$$F_n(x) = \frac{1}{4n}\sum_{j=1}^{n} x_j^2 - \tfrac{1}{4},$$

where $x \in \Omega = R^n_+$, $i = 1, 2, \dots, n-1$.

Example 1 comes from Ref. 9. Example 2 is a modification of the tridiagonal exponential problem[14] by adding an additional convex constraint. Example 3 is a modification of the Penalty I function[15] by adding an additional convex constraint. It is not difficult to show that all the functions F in these problems are monotone.

Table 3 The numerical results of Example 3 with the given initial points.

	dim	Algorithm 1		SPRP	
		niter	CPU	niter	CPU
x0	5000	325	0.17	622	0.32
	10 000	507	0.48	984	0.92
	20 000	777	1.39	1516	2.84
x1	5000	325	0.17	622	0.32
	10 000	507	0.47	984	0.93
	20 000	777	1.37	1516	2.82
x2	5000	320	0.19	614	0.30
	10 000	502	0.44	974	0.85
	20 000	769	1.26	1502	2.53
x3	5000	324	0.17	622	0.33
	10 000	507	0.47	984	0.93
	20 000	777	1.38	1516	2.80
x4	5000	325	0.18	621	0.32
	10 000	507	0.47	984	0.93
	20 000	777	1.38	1516	2.81
x5	5000	321	0.19	618	0.33
	10 000	503	0.48	974	0.93
	20 000	770	1.37	1502	2.87

The numerical results are listed in Tables 1–4, where dim stands for the dimension of the test problems, and niter and CPU represent the total number of iterates and the CPU time in seconds, respectively. To show the practicability of Algo-

Table 4 The numerical results with random initial points.

	dim	Algorithm 1		SPRP	
		niter	CPU	niter	CPU
Example 1	5000	342	0.27	619	0.33
	10 000	429	0.42	780	0.73
	20 000	538	0.98	983	1.86
Example 2	5000	5	0.03	14	0.08
	10 000	5	0.05	14	0.12
	20 000	5	0.06	15	0.24
Example 3	5000	323	0.17	619	0.32
	10 000	506	0.47	980	0.94
	20 000	775	1.31	1512	2.80

rithm 1, we test the given problems with the number of variables $n = 5000$, $10\,000$, $20\,000$ and some given initial points: $x0 = (-0.1, -0.1, \dots, -0.1)^{\mathrm{T}}$, $x1 = (-1, -1, \dots, -1)^{\mathrm{T}}$, $x2 = (-1, 1, -1, \dots)^{\mathrm{T}}$, $x3 = (-0.1, 0.1, -0.1 \cdots)^{\mathrm{T}}$, $x4 = (1, \frac{1}{2}, \dots, 1/n)^{\mathrm{T}}$, $x5 = (1 - 1/n, 1 - 2/n, \dots, 1 - n/n)^{\mathrm{T}}$. In addition, for the given n, each problem is tested three times with random initial points generated by MATLAB's rand(n,1) in (−1,1), and the average of the three results is given in Table 4

We see from Tables 1–3 that, starting from the given initial points, the two methods terminate successfully at a solution of the problem. For Example 2, there seems to be not much difference among the performances of the two methods. However, Algorithm 1 performs better than the SPRP method for Example 1 and Example 3. Table 4 further indicates that Algorithm 1 is more stable and efficient than the SPRP method for the given problems.

REFERENCES

1. Zhao WY, Li D (2000) Monotonicity of fixed point and normal mapping associated with variational inequality and its application. *SIAM J Optim* **11**, 962–73.
2. Dirkse SP, Ferris MC (1995) MCPLIB: A collection of nonlinear mixed complementarity problems. *Optim Meth Software* **5**, 319–45.
3. Meintjes K, Morgan AP (1987) A methodology for solving chemical equilibrium systems. *Appl Math Comput* **22**, 333–61.
4. Solodov MV, Svaiter BF (1999) A globally convergent inexact Newton method for systems of monotone equations. In: *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, Springer, pp 355–69.
5. Barzilai J, Borwein JM (1988) Two point step size gradient methods. *IMA J Numer Anal* **8**, 141–8.
6. Zhang L, Zhou W (2006) Spectral gradient projection method for solving nonlinear monotone equations. *J Comput Appl Math* **196**, 478–84.
7. Yu ZS, Lin J, Sun J, Xiao YH, Liu LY, Li ZH (2009) Spectral gradient projection method for monotone nonlinear equations with convex constraints. *Appl Numer Math* **59**, 2416–23.
8. Polak BT (1969) The conjugate gradient method in extreme problems. *USSR Comput Math Math Phys* **4**, 94–112.
9. Liu JK (2016) Derivative-free spectral PRP projection method for solving nonlinear monotone equations with convex constraints. *Math Numer Sin* **38**, 113–24.[in Chinese].
10. Hager WW,Zhang H (2005) A new conjugate gradient method with guaranteed descent and an efficiet line search. *SIAM J Optim* **16**, 170–92.
11. Cheng WY, Li DH (2010) Spectral scaling BFGS method. *J Optim Theor Appl* **146**, 305–19.
12. Liu H, Wang HJ, Ni Q (2014) On Hager and Zhang's conjugate gradient method with guaranteed descent. *Appl Math Comput* **236**, 400–7.
13. Dai YH, Kou CX (2013) A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM J Optim* **23**, 296–320.
14. Bing Y, Lin G (1991) An efficient implementation of Merrill's method for sparse or partitially separable systems of nonlinear equations. *SIAM J Optim* **1**, 206–21.
15. La Cruz W, Martínez JM, Raydan M (2006) Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Math Comput* **75**, 1429–48.