MULTIVARIATE SPECTRAL DY-TYPE PROJECTION METHOD
FOR CONVEX CONSTRAINED NONLINEAR MONOTONE
EQUATIONS

JINKUI LIU*
College of Mathematics and Statistics
Chongqing University
Chongqing, 401331, China
School of Mathematics and Statistics
Chongqing Three Gorges University
Wanzhou, 404100, China

SHENGJIE LI
College of Mathematics and Statistics
Chongqing University
Chongqing, 401331, China

(Communicated by Masao Fukushima)

Abstract. In this paper, we consider a multivariate spectral DY-type projection method for solving nonlinear monotone equations with convex constraints. The search direction of the proposed method combines those of the multivariate spectral gradient method and DY conjugate gradient method. With no need for the derivative information, the proposed method is very suitable to solve large-scale nonsmooth monotone equations. Under appropriate conditions, we prove the global convergence and R-linear convergence rate of the proposed method. The preliminary numerical results also indicate that the proposed method is robust and effective.

1. Introduction. In this paper, we consider the solutions of the following nonlinear monotone equations

\[ F(x) = 0, \ x \in \Omega, \]

where \( \Omega \subseteq \mathbb{R}^n \) is a non-empty closed convex set, and \( F : \Omega \rightarrow \mathbb{R}^n \) is continuous and monotone, i.e. \( \langle F(x) - F(y), x - y \rangle \geq 0, \ \forall x, y \in \Omega. \) Due to the monotonicity of \( F(x) \), the solution set of problem (1) denoted by \( \Omega^* \) is convex. Throughout this paper, we assume that \( \Omega^* \) is nonempty.

The focus of this paper is on solving the nonlinear monotone equations whose Jacobian matrix is not available or requires a prohibitive amount of storage. This situation is very common when problem (1) comes from the practical applications.
such as the economic equilibrium problems [7], the power flow equations [18] and
the chemical equilibrium systems [13, 14], and so on.

There are numerous methods which can be used to solve problem (1), see Refs.[5],
[6], [9], [15], [16] and [19]. Among these methods, Newton method, quasi-Newton
method, Gauss-Newton methods, Levenberg-Marquardt method and their variants
are very attractive, because their superlinear convergence rate can be achieved under
some suitable assumptions. It is a pity that these methods need to solve linear
equations using the Jacobian matrix or an approximation of the Jacobian matrix
at each iteration, which leads to that these methods are not suitable for large-scale
nonsmooth monotone equations.

The spectral gradient method, originally proposed by Barzilai and Borwein [1]
for unconstrained optimization problems, has been successfully extended to solve
nonlinear monotone equations by Cruz and Raydan [10] [11]. Recently, Zhang and
Zhou [22] combined the spectral gradient method [1] with the projection technique
[16] to construct a spectral gradient projection method for solving nonlinear mono-
tone equations. When the nonlinear monotone equations is Lipschitz continuous,
the global convergence of this method was established. An attractive feature of
this method is that it can be used to solve nonsmooth equations. Han et al. [8]
popularized the spectral gradient method [1], and proposed a multivariate spectral
gradient method for unconstrained optimization problems, which is finitely conver-
gent for positive definite quadratics. The global convergence is established for the
multivariate spectral gradient method with a nonmonotone line search. Yu et al.
[21] further studied a multivariate spectral projected gradient method for bounded
constrained optimization. Most recently, Yu et al. [22] established a multivariate
spectral projection gradient method for nonlinear monotone equations with convex
constraints which can be viewed as an extension of multivariate spectral projected
gradient method [21].

The main purpose of this paper is to study a multivariate spectral DY-type
projection method for solving large-scale nonlinear monotone equations
with convex constraints. The search direction of the proposed method can be viewed
as a correction of the search direction generated in [22] using the search direction
of DY conjugate gradient method [3]. Under appropriate conditions, the global
convergence and R-linear convergence rate of the proposed method are proved.
Since there is no any derivative information, the proposed method is very suitable
to solve large-scale nonlinear monotone equations.

The rest of this paper is organized as follows. In Section 2, we give the mul-
tivariate spectral DY-type projection method. In Section 3, we prove the global
convergence of the proposed method under some suitable conditions. In Section
4, we establish the R-linear convergence rate of the proposed method with an as-
sumption. Preliminary numerical results of the given test problems are presented
in Section 5.

2. Multivariate spectral DY-type projection method. Spectral conjugate
gradient method was firstly proposed by Barzilai and Borwein [1] for solving the
unconstrained optimization problem

$$
\min_{x \in R^n} f(x),
$$
where \( f : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable and its gradient denoted by \( g \) is available. This method usually generates a sequence \( \{x_k\} \) by
\[
x_{k+1} = x_k - \frac{1}{\alpha_k} g_k,
\]
where \( \alpha_k > 0 \) is defined by
\[
\alpha_k = \frac{s_k^T y_{k-1}}{s_k^T s_{k-1}},
\]
with \( s_{k-1} = x_k - x_{k-1}, y_{k-1} = g_k - g_{k-1} \). The choice of \( \alpha_k \) imposes some quasi-Newton property, which can be obtained by minimizing \( \|\alpha Is_{k-1} - y_{k-1}\| \) with respect to \( \alpha \); \( \alpha I \) approximates the Hessian matrix of \( f \) at \( x_k \). Recently, Han et al. [8] replaced \( \alpha_k \) with a vector \( \text{diag}\{\lambda_1^0, \lambda_2^0, \ldots, \lambda_n^0\} \), and obtained a multivariate spectral gradient method for solving unconstrained optimization problems. The vector \( \text{diag}\{\lambda_1^0, \lambda_2^0, \ldots, \lambda_n^0\} \) is generated by minimizing
\[
\|\text{diag}\{\lambda_1^0, \lambda_2^0, \ldots, \lambda_n^0\} s_{k-1} - y_{k-1}\|
\]
with respect to \( \{\lambda^i\}^n_{i=1} \). Then the multivariate spectral gradient iterative scheme is
\[
x_{k+1} = x_k - \text{diag}\{1/\lambda_1^k, 1/\lambda_2^k, \ldots, 1/\lambda_n^k\} g_k.
\]

In what follows, we popularize the multivariate spectral gradient method, and propose a multivariate spectral DY-type projection method with the help of the search direction of DY method and the projection technique. Denote the \( i \)th component of \( y_k \) and \( s_k \) as \( y^i_k \) and \( s^i_k \), respectively. For the sake of notational convenience, we denote \( F(x_k) \) as \( F_k \). We are now ready to formally present the algorithmic framework as follows.

**Algorithm 2.1**

**Step 0:** Give \( x_0 \in \mathbb{R}^n \), and \( \beta, \sigma, \epsilon, \xi \in (0, 1], \rho \in (0, 1), \delta > 0 \). Set \( k = 0 \).

**Step 1:** Let \( d_0 = -F_0 \).

**Step 2:** Set \( z_k = x_k + \alpha_k d_k \). The step-size \( \alpha_k = \max\{\beta \rho^i | i = 0, 1, 2, \ldots \} \) is determined by the following line search:
\[
-F(z_k)^T d_k \geq \sigma \alpha_k ||d_k||^2.
\]

**Step 3:** If \( z_k \in \Omega \) and \( F(z_k) = 0 \), then \( x_{k+1} = z_k \) stop. Otherwise, we obtain the next iterative point \( x_{k+1} \) by
\[
x_{k+1} = P_{\Omega}[x_k - \eta_k F(z_k)],
\]
where
\[
\eta_k = \frac{(x_k - z_k)^T F(z_k)}{||F(z_k)||^2}.
\]

**Step 4:** If \( F(x_{k+1}) = 0 \), stop. Otherwise, generate the next search direction \( d_{k+1} \) as follows:

If \( y^i_k/s^i_k > 0 \), then we set \( \lambda^i_{k+1} = y^i_k/s^i_k \); otherwise we set \( \lambda^i_{k+1} = s^i_k/y^i_k \), where \( y_k = F(x_{k+1}) - F(x_k), s_k = z_k - x_k \).

If \( \lambda^i_{k+1} \leq \epsilon \) or \( \lambda^i_{k+1} \geq 1/\epsilon \), then we set \( \lambda^i_{k+1} = \delta \), for \( i = 1, 2, \ldots, n \). Set
\[
d_{k+1} = -A_{k+1} F_{k+1} + \beta_{k+1} d_{k+1}.
\]

Here \( A_{k+1} = \text{diag}\{1/\lambda_1^{k+1}, 1/\lambda_2^{k+1}, \ldots, 1/\lambda_n^{k+1}\} \), \( \beta_{k+1} = ||F_{k+1}||^2/d^T w_k \), \( w_k = y_k + t d_k \), \( t = 1 + \max\{0, -\frac{d^T y_k}{d^T d_k}\} \).
If $F_{k+1}^T d_{k+1} \geq -\xi ||F_{k+1}||^2$, then $d_{k+1} = -F_{k+1}$.

**Step 5:** Set $k := k + 1$, go to step 2.

**Remark 1.** The scalar $\frac{s_{k-1}^\star y_{k-1}}{s_{k-1}^\star s_{k-1}^\star}$ is the spectral coefficient which is an appropriate Rayleigh quotient with respect to a secant approximation of Jacobian. It was chosen as the step-size of the steepest descent method in [1], which can be obtained by minimizing $||\omega I s_{k-1} - y_{k-1}||$ with respect to $\omega$.

**Remark 2.** From the definitions of $t$ and $w_k$, we have that
\[
d_k^T w_k = d_k^T y_k + (1 + \max\{0, -\frac{d_k^T y_k}{d_k^T d_k}\}) \cdot d_k^T d_k
\]
\[
\geq d_k^T y_k + (1 - \frac{d_k^T y_k}{d_k^T d_k}) ||d_k||^2 = ||d_k||^2,
\]
which implies that the parameter $\beta_k^\star$ is interesting with any line search.

3. **Convergence property.** In this section, we show that Algorithm 2.1 is a contraction algorithm, and then we prove its convergence under the analytic framework of contraction type methods.

**Assumption 1.** The function $F : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous, i.e., there exists a constant $L > 0$ such that
\[
||F(x) - F(y)|| \leq L||x - y||, \ \forall \ x, y \in \Omega.
\] (5)

The following lemma shows that the sequence $\{x_k\}$ generated by Algorithm 2.1 is Fejér monotone with respect to $\Omega^\star$, and also indicates that Algorithm 2.1 is also a contraction method for problem [1], as stated in the following lemma.

**Lemma 3.1.** Let sequences $\{x_k\}$ and $\{z_k\}$ be generated by Algorithm 2.1. For any $x^\star \in \Omega^\star$, we have
\[
||x_{k+1} - x^\star||^2 \leq ||x_k - x^\star||^2 - c||x_k - z_k||^4, \ c \in (0, 1).
\] (6)

**Proof.** From the definition of $z_k$, it follows from (2) that
\[
(x_k - z_k)^T F(z_k) = -\alpha_k F^T(z_k) d_k \geq \sigma \alpha_k^2 ||d_k||^2 \geq 0,
\]
then we have
\[
(x_k - x^\star)^T F(z_k) \geq (x_k - z_k)^T F(z_k) + (z_k - x^\star)^T F(x^\star)
\] (7)
where the first inequality follows from the monotonicity of $F$.

From (3) it holds that
\[
||x_{k+1} - x^\star||^2 = ||P_\Omega[x_k - \eta_k F(z_k)] - x^\star||^2
\]
\[
\leq ||x_k - \eta_k F(z_k) - x^\star||^2
\]
\[
= ||x_k - x^\star||^2 - 2\eta_k(x_k - x^\star)^T F(z_k) + \eta_k^2 ||F(z_k)||^2
\]
\[
\leq ||x_k - x^\star||^2 - \frac{(x_k - z_k)^T F(z_k) \cdot (x_k - z_k)^T F(z_k)}{||F(z_k)||^2},
\] (8)
where the first inequality uses the nonexpansive property of the projection operator, the second inequality applies inequality (7). From (8), it is easy to obtain that
\[
||x_{k+1} - x^\star|| \leq ||x_k - x^\star||,
\]
which means that the sequence \( \{||x_k - x^*||\} \) is monotonically decreasing, and has the lower boundary. Thus, the sequence \( \{||x_k - x^*||\} \) is convergent which implies that the sequence \( \{x_k\} \) is bounded.

For any \( x^* \in \Omega^* \), it follows from (5) that
\[
||F(x_k)|| = ||F(x_k) - F(x^*)|| \leq L||x_k - x^*|| \leq L||x_0 - x^*||,
\]
which implies that the sequence \( \{||F(x_k)||\} \) is bounded, i.e., there exists a constant \( \vartheta = \max\{\sigma, L||x_0 - x^*||\} \) such that \( ||F(x_k)|| \leq \vartheta \). Then we have
\[
\sigma||x_k - z_k|| = \sigma \frac{||x_k - z_k||^2}{||x_k - z_k||} = \sigma \frac{\alpha_k^2 ||d_k||^2}{||x_k - z_k||} \leq -\frac{\alpha_k F(z_k)^T d_k}{||x_k - z_k||} = \frac{F(z_k)^T (x_k - z_k)}{||x_k - z_k||} \leq \frac{||F(x_k)||}{||x_k - z_k||} \leq \vartheta \sigma \tag{9}
\]
where the first inequality follows from (2), the second inequality follows from the monotonicity of \( F \), and the third inequality applies Cauchy-Schwartz inequality.

From Triangle inequality, (5) and (9), we have
\[
||F(z_k)|| - ||F(x_k)|| \leq ||F(z_k) - F(x_k)|| \leq L||z_k - x_k|| \leq L\vartheta \sigma ,
\]
then it holds that
\[
||F(z_k)|| \leq \vartheta + \frac{L\vartheta}{\sigma} . \tag{10}
\]

In addition, it follows from (9) that
\[
(x_k - z_k)^T F(z_k) \geq \sigma ||x_k - z_k||^2 ,
\]
which together with (8) and (10) gives
\[
||x_{k+1} - x^*||^2 \leq ||x_k - x^*||^2 - c||x_k - z_k||^4 ,
\]
where \( c = \frac{\sigma^2}{(\sigma + L^2)^2} \). From \( \vartheta = \max\{\sigma, L||x_0 - x^*||\} \), it is not difficult to obtain \( c \in (0, 1) \).

**Theorem 3.2.** The sequence \( \{x_k\} \) generated by Algorithm 2.1 converges to a solution point of problem (1).

**Proof.** From (6), the sequence \( \{x_k\} \) generated by Algorithm 2.1 is bounded. In fact, the sequence \( \{x_k\} \) is contained in the following compact set
\[
S := \{x \in \Omega | ||x - x^*|| \leq ||x_0 - x^*||\} .
\]
So, there exists a cluster point of \( \{x_k\} \), denoted by \( \tilde{x} \). We assume that the subsequence \( \{x_{k_j}\} \) converges to \( \tilde{x} \). From the standard techniques of the Fejér monotonicity [2], it is not difficult to find that \( \tilde{x} \) is a solution of problem (1). Moreover, \( \tilde{x} \) is the unique cluster of the sequence \( \{x_k\} \).
4. R-linear convergence rate. In this section, the R-linear convergence rate of Algorithm 2.1 is proved with the following assumption.

Assumption 2. For any \( x^* \in \Omega^* \), there exist constants \( \mu \in (0, 1) \) and \( \eta > 0 \) such that
\[
\mu \text{dist}(x, \Omega^*) \leq ||F(x)||^2, \forall x \in N(x^*, \eta),
\]
(11)
where \( \text{dist}(x, \Omega^*) \) denotes the distance from \( x \) to the solution set \( \Omega^* \), and \( N(x^*, \eta) := \{ x \in \mathbb{R}^n | ||x - x^*|| \leq \eta \} \).

From Theorem 3.1, we always assume that \( x_k \rightarrow x^* \) as \( k \rightarrow \infty \), where \( x^* \) belongs to the solution set \( \Omega^* \) of problem (1).

Theorem 4.1. Suppose Assumptions 1 and 2 hold. For the sequence \( \{x_k\} \) generated by Algorithm 2.1, the sequence \( \{\text{dist}(x_k, \Omega^*)\} \) \( Q \)-linearly converges to 0, thus the whole sequence \( \{x_k\} \) converges to \( x^* \) \( R \)-linearly.

Proof. Let \( \nu_k := \arg\min \{||x_k - \nu|| \mid \nu \in \Omega^*\} \), this means that \( \nu_k \) is the closest solution to \( x_k \), i.e., \( ||x_k - \nu_k|| = \text{dist}(x_k, \Omega^*) \).

From Step 4, it is not difficult to obtain that
\[
F_k^T d_k \leq -\max\{1, \xi\}||F_k||^2 = -\theta||F_k||^2, \forall k \geq 0,
\]
where \( \theta = \max\{1, \xi\} \). Then using Cauchy-Schwartz inequality, we have
\[
||d_k|| \geq \theta||F_k||, \forall k \geq 0.
\]
(12)
Due to \( \nu_k \in \Omega^* \), from (10) it holds that
\[
\text{dist}(x_{k+1}, \Omega^*)^2 \leq \text{dist}(x_k, \Omega^*)^2 - \mu_k^2||d_k||^4 \leq \text{dist}(x_k, \Omega^*)^2 - \theta^2\mu^2\alpha_k^4||F_k||^4
\]
\[
\leq \text{dist}(x_k, \Omega^*)^2 - \theta^4\mu^2\alpha_k^4\text{dist}(x_k, \Omega^*)^2
\]
\[= (1 - \theta^4\mu^2\alpha_k^4)\text{dist}(x_k, \Omega^*)^2,
\]
where the second inequality follows from (12), and the third inequality applies (11).

The last inequality shows that the sequence \( \{\text{dist}(x_k, \Omega^*)\} \) \( Q \)-linearly converges to 0. Thus, the sequence \( \{x_k\} \) \( R \)-linearly converges to \( x^* \).

5. Numerical results. In this section, we apply Algorithm 2.1 to test some commonly nonlinear monotone equations with convex constraints, and compare it with MSGP method, a multivariate spectral gradient projection method proposed by Yu et al. [22] which performed better than a spectral gradient method [12] and a projection method [17]. Our code is written in MATLAB 7.0, and run on a HP personal computer with Intel Core (TM) CPU 2.60GHZ and 2G memory. The test problems are listed as follows:

Problem 1. The problem can be viewed as a modification of Logarithmic function in [11]. \( F : \Omega \rightarrow \mathbb{R}^n \) with
\[
F(x) = \ln(|x| + 1) - x/n, \quad \Omega = \mathbb{R}^n_+.
\]

Problem 2. The problem can be viewed as a modification of Exponential function 2 in [11]. \( F : \Omega \rightarrow \mathbb{R}^n \) with
\[
F_i(x) = e^{x_i} - 1,
\]
\[
F_i(x) = e^{x_i} + x_{i-1} - 1, \quad i = 2, 3, \cdots, n,
\]
and \( \Omega = \mathbb{R}^n_+ \).
Problem 3. The problem can be viewed as a modification of one problem in [3]. Let \( F : \Omega \rightarrow \mathbb{R}^n \) with
\[
F_1(x) = 2x_1 - x_2 + e^{x_1} - 1, \\
F_i(x) = -x_{i-1} + 2x_i - x_{i+1} + e^{x_i} - 1, i = 2, 3, \ldots, n-1, \\
F_n(x) = -x_{n-1} + 2x_n + e^{x_n} - 1,
\]
and \( \Omega = \mathbb{R}^n_+ \).

Problem 4. The problem can be viewed as a modification of one problem in [20]. Let \( F : \Omega \rightarrow \mathbb{R}^n \) with
\[
F_1(x) = 2.5x_1 + x_2 - 1, \\
F_i(x) = x_{i-1} + 2.5x_i + x_{i+1} - 1, i = 2, 3, \ldots, n-1, \\
F_n(x) = x_{n-1} + 2.5x_n - 1,
\]
and \( \Omega = \mathbb{R}^n_+ \).

Throughout the numerical experiments, unless otherwise stated, the parameters in Algorithm 2.1 are set as: \( \rho = 0.65, \sigma = 0.001, \xi = 0.0001, \delta \) is chosen in the same way used in [10], i.e.
\[
\delta = \begin{cases} 
1, & \text{if } ||F(x_k)|| = 1, \\
||F(x_k)||^{-1}, & \text{if } 10^{-5} \leq ||F(x_k)|| \leq 1, \\
10^5, & \text{if } ||F(x_k)|| < 10^{-5}.
\end{cases}
\]
And the initial trial stepsize \( \beta \) is obtained by
\[
\beta = \frac{F(x_k)^T d_k}{d_k^T (F(x_k + \tau d_k) - F(x_k))/\tau} \approx \frac{F(x_k)^T d_k}{d_k^T \nabla F(x_k) d_k},
\]
where \( \tau = 10^{-8} \). If the parameter \( \beta \leq 10^{-6} \) holds, we set \( \beta = 1 \). In MSGP method, all parameters come from the reference [22]. We stop the iteration if \( ||F(x_k)|| \leq 10^{-5} \) is satisfied. We also stop the iteration when 10000 iterations are completed without achieving convergence.

In order to show the performance of Algorithm 2.1, we do a lot of numerical experiments. Firstly, we test the given problems with the number of variables \( n = 1000, 3000, 5000, 10000, 12000 \) and some given initial points: \( x_1 = (1, 1, \ldots, 1)^T, \)
\( x_2 = (1, \frac{1}{2}, \ldots, \frac{1}{n})^T, \)
\( x_3 = (0, \frac{1}{2}, \ldots, \frac{n-1}{n})^T, \)
\( x_4 = (\frac{n-1}{n}, \frac{n-2}{n}, \ldots, 0)^T, \)
\( x_5 = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})^T, \)
respectively. Secondly, we test each given problem with three different initial points randomly generated from \((0, 1)\). The results are reported in Tables 1-6, where “Dim” represents the dimension of problem (the dimension of the variable \( x \)). The detailed results are presented in the form “NI/NF/CPU/NFV”, where “NI” stands for the number of iterations, “NF” denotes the number of function value calculations, CPU stands for CPU time in seconds, “NFV” denotes the final value of \( ||F_k|| \) when the algorithm terminates.

For the \( i \)th group of experiment results, let \( ||F_{\text{Algorithm 2.1},i}|| \) and \( ||F_{\text{MSGP},i}|| \) be respectively the final values of \( ||F|| \) when Algorithm 2.1 and MSGP method terminate. We say that, the performance of Algorithm 2.1 method is better than the performance of MSGP method if
\[
| ||F_{\text{Algorithm 2.1},i}|| - ||F_{\text{MSGP},i}|| | \leq 10^{-5},
\]
and the number of iterations, or the number of function value calculations, or CPU time of Algorithm 2.1 is less than the number of iterations, or the number of function value calculations, or CPU time of MSGP method, respectively. By this rule, for the number of iterations, Algorithm 2.1 is better in 141 experiments (i.e., it achieves the...
minimum number of iterations in 141 experiments), and MSGP method is better in 18 experiments. For the number of function value calculations, Algorithm 2.1 is better in 136 experiments, and MSGP method is better in 24 experiments. For CPU time, Algorithm 2.1 is better in 150 experiments, and MSGP method is better in 9 experiments. Thus, the proposed method is more effective than MSGP method for the given problems. This should be attributed to the adjustment of search direction.

### Table 1. The results of Problem 1 with given initial points

| Dim | Initial points | MSGP method | Algorithm 2.1 |
|-----|----------------|--------------|---------------|
| 1000 | X1             | 7/22/0.06/1.31135e-007 | 13/40/0.05/3.38669e-006 |
|      | X2             | 2/7/0.05/0.00000e+000 | 4/13/0.03/0.00000e+000 |
|      | X3             | 11/34/0.08/2.19964e-007 | 3/40/0.05/3.36081e-006 |
|      | X4             | 3/10/0.05/0.00000e+000 | 6/19/0.05/0.00000e+000 |
|      | X5             | 11/34/0.09/8.93938e-008 | 8/25/0.06/6.13420e-006 |
| 3000 | X1             | 2/7/0.05/0.00000e+000 | 13/40/0.06/5.81725e-006 |
|      | X2             | 11/34/0.25/4.90071e-006 | 4/13/0.03/0.00000e+000 |
|      | X3             | 11/34/0.30/2.69613e-006 | 6/19/0.05/0.00000e+000 |
|      | X4             | 7/22/0.31/1.31003e-007 | 10/31/0.42/2.65195e-006 |
| 5000 | X1             | 2/7/0.05/0.00000e+000 | 13/40/0.08/7.49753e-006 |
|      | X2             | 11/34/0.53/7.57273e-006 | 4/13/0.05/0.00000e+000 |
|      | X3             | 3/10/0.13/0.00000e+000 | 13/40/0.09/7.48340e-006 |
|      | X4             | 12/37/0.78/8.40744e-008 | 6/19/0.05/0.00000e+000 |
|      | X5             | 7/22/0.64/1.39968e-007 | 10/31/1.05/3.78182e-006 |
| 10000| X1             | 2/7/0.06/0.00000e+000 | 14/43/0.14/2.89200e-006 |
|      | X2             | 11/34/1.77/6.56131e-006 | 4/13/0.06/0.00000e+000 |
|      | X3             | 3/10/0.36/0.00000e+000 | 14/43/0.16/2.88949e-006 |
|      | X4             | 13/40/3.00/6.90437e-008 | 6/19/0.08/0.00000e+000 |
|      | X5             | 7/22/2.23/1.30940e-007 | 10/31/3.20/5.11269e-006 |
| 12000| X1             | 2/7/0.06/0.00000e+000 | 14/43/0.17/3.16733e-006 |
|      | X2             | 11/34/2.48/6.34881e-006 | 4/13/0.06/0.00000e+000 |
|      | X3             | 3/10/0.47/0.00000e+000 | 14/43/0.16/3.16500e-006 |
|      | X4             | 13/40/4.53/6.04693e-008 | 6/19/0.09/0.00000e+000 |
|      | X5             | 7/22/3.15/1.30935e-007 | 10/31/4.67/5.34936e-006 |

Acknowledgments. We would like to thank the anonymous referees and editors whose comments helped a lot to improve our paper.

REFERENCES

[1] J. M. Barzilai and M. Borwein, Two point step size gradient methods IMA Journal on Numerical Analysis, 8 (1988), 141–148.
[2] H. H. Bauschke and P. L. Combettes, A weak-to-strong convergence principle for Fejé- 

...
Table 2. The results of Problem 2 with given initial points

| Dim | Initial points | MSGP method | Algorithm 2.1 |
|-----|----------------|-------------|---------------|
| 1000| X1 329/1164/1.28/9.8195e-006 | 33/161/0.05/9.43667e-006 | 33/161/0.05/9.43667e-006 |
|     | X2 258/1153/1.19/8.96144e-006 | 56/294/0.06/9.59066e-006 | 56/294/0.06/9.59066e-006 |
|     | X3 86/381/0.17/9.29618e-006 | 37/184/0.05/8.83590e-006 | 37/184/0.05/8.83590e-006 |
|     | X4 61/275/0.30/9.56339e-006 | 62/330/0.06/9.97633e-006 | 62/330/0.06/9.97633e-006 |
|     | X5 361/1457/1.55/8.71560e-006 | 47/246/0.23/9.34449e-006 | 47/246/0.23/9.34449e-006 |
| 3000| X1 61/281/0.98/9.44368e-006 | 44/219/0.09/8.60313e-006 | 44/219/0.09/8.60313e-006 |
|     | X2 347/1390/9.69/9.33905e-006 | 51/289/0.11/9.89992e-006 | 51/289/0.11/9.89992e-006 |
|     | X3 110/467/2.80/8.61958e-006 | 38/189/0.08/7.07861e-006 | 38/189/0.08/7.07861e-006 |
|     | X4 65/295/0.89/9.73034e-006 | 47/275/0.11/6.69019e-006 | 47/275/0.11/6.69019e-006 |
|     | X5 361/1457/10.36/8.71560e-006 | 47/246/1.31/8.93449e-006 | 47/246/1.31/8.93449e-006 |
| 5000| X1 73/324/3.75/9.96592e-006 | 57/291/0.19/9.69435e-006 | 57/291/0.19/9.69435e-006 |
|     | X2 305/1224/22.45/9.99741e-006 | 61/326/0.19/9.23331e-006 | 61/326/0.19/9.23331e-006 |
|     | X3 99/420/6.53/9.97198e-006 | 44/220/0.14/8.90390e-006 | 44/220/0.14/8.90390e-006 |
|     | X4 64/285/4.08/8.54432e-006 | 54/292/0.19/5.23666e-006 | 54/292/0.19/5.23666e-006 |
|     | X5 361/1457/26.92/8.71560e-006 | 47/246/3.28/8.93449e-006 | 47/246/3.28/8.93449e-006 |
| 10000| X1 67/301/15.72/9.73591e-006 | 43/212/0.25/6.39422e-006 | 43/212/0.25/6.39422e-006 |
|      | X2 367/1471/101.08/9.69111e-006 | 66/467/0.52/8.76219e-006 | 66/467/0.52/8.76219e-006 |
|      | X3 79/353/18.42/9.73575e-006 | 65/331/0.63/8.97609e-006 | 65/331/0.63/8.97609e-006 |
|      | X4 120/498/49.52/9.58265e-006 | 57/311/0.41/9.82673e-006 | 57/311/0.41/9.82673e-006 |
|      | X5 361/1457/144.38/8.71560e-006 | 47/246/12.08/8.93449e-006 | 47/246/12.08/8.93449e-006 |
| 12000| X1 67/301/15.72/9.73591e-006 | 43/212/0.25/6.39422e-006 | 43/212/0.25/6.39422e-006 |
|      | X2 390/1562/154.20/8.01930e-006 | 66/467/0.52/8.76219e-006 | 66/467/0.52/8.76219e-006 |
|      | X3 82/362/25.55/9.73575e-006 | 65/331/0.63/8.97609e-006 | 65/331/0.63/8.97609e-006 |
|      | X4 134/548/49.52/9.58265e-006 | 57/311/0.41/9.82673e-006 | 57/311/0.41/9.82673e-006 |
|      | X5 361/1457/144.38/8.71560e-006 | 47/246/17.19/8.93449e-006 | 47/246/17.19/8.93449e-006 |

[8] L. Han, G. H. Yu and L. T. Guan, Multivariate spectral gradient method for unconstrained optimization, *Applied Mathematics and Computation*, 201 (2008), 621–630.

[9] A. N. Iusem and M. V. Solodov, Newton-type methods with generalized distances for constrained optimization, *Optimization*, 41 (1997), 257–278.

[10] W. La Cruz and M. Raydan, Nonmonotone spectral methods for large-scale nonlinear systems, *Optimization Methods and Software*, 18 (2003), 583–599.

[11] W. La Cruz, J. M. Martínez and M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Mathematics of Computation*, 75 (2006), 1429–1448.

[12] Q. N. Li and D. H. Li, A class of derivative-free methods for large-scale nonlinear monotone equations, *IMA Journal on Numerical Analysis*, 31 (2011), 1625–1635.

[13] K. Meintjes and A. P. Morgan, A methodology for solving chemical equilibrium systems, *Applied Mathematics and Computation*, 22 (1987), 333–361.

[14] K. Meintjes and A. P. Morgan, Chemical equilibrium systems as numerical test problems, *ACM Transactions on Mathematical Software*, 16 (1990), 143–151.

[15] L. Qi and J. Sun, A nonsmooth version of Newton's method, *Mathematical Programming*, 58 (1999), 353–367.

[16] 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, Applied Optimization, 22, Kluwer Acad. Publ., Dordrecht, 1999, 355–369.

[17] C. W. Wang, Y. J. Wang and C. L. Xu, A projection method for a system of nonlinear monotone equations with convex constraints, *Mathematical Methods and Operations Research*, 66 (2007), 33–46.

[18] A. J. Wood and B. F. Wollenberg, *Power Generations Operations and Control*, Wiley, New York, 1996.

[19] N. Yamashita and M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, *Computing*, 15 (2001), 239–249.
Table 3. The results of Problem 3 with given initial points

| Dim | Initial points | MSGP method | Algorithm 2.1 |
|-----|----------------|--------------|---------------|
| 1000| X1 54/226/0.09/9.68368e-006 | 43/264/0.06/9.77466e-006 |
|     | X2 55/4532/1.11/9.59335e-006  | 36/230/0.06/8.53004e-006 |
|     | X3 1004/9105/1.70/9.63297e-006 | 48/311/0.06/1.82714e-006 |
|     | X4 871/7511/1.70/9.3555e-006 | 51/318/0.06/8.53404e-006 |
|     | X5 58/264/0.30/9.42953e-006 | 39/257/0.16/8.20033e-006 |
| 3000| X1 53/213/0.16/7.47095e-006  | 53/335/0.14/7.17598e-006 |
|     | X2 628/5226/4.92/9.53246e-006 | 36/226/0.09/6.14965e-006 |
|     | X3 1237/12041/7.30/9.87999e-006 | 47/303/0.13/7.53500e-006 |
|     | X4 1059/9647/13.73/9.81351e-006 | 54/386/0.16/9.6175e-006 |
|     | X5 58/264/1.73/9.42953e-006 | 39/257/0.11/8.20033e-006 |
| 5000| X1 51/212/0.20/8.70652e-006  | 49/311/0.19/7.89292e-006 |
|     | X2 635/5353/10.84/9.92649e-006 | 39/247/0.16/6.13645e-006 |
|     | X3 1397/13868/16.84/9.8769e-006 | 77/531/0.30/1.7119e-006 |
|     | X4 1093/10060/16.78/9.7031e-006 | 84/687/0.36/5.17010e-006 |
|     | X5 58/264/4.27/9.42953e-006 | 39/257/2.75/8.20033e-006 |
| 10000| X1 61/243/0.45/8.97371e-006 | 56/365/0.41/8.54468e-006 |
|     | X2 206/1273/16.94/9.54721e-006 | 42/275/0.30/5.75051e-006 |
|     | X3 1739/18415/57.48/9.96405e-006 | 61/418/0.45/6.13088e-006 |
|     | X4 1102/10030/135.05/9.7795e-006 | 4/125/0.69/0.00000e+00 |
|     | X5 58/264/15.72/9.42953e-006 | 39/257/10.11/8.20033e-006 |
| 12000| X1 65/267/0.58/9.69254e-006 | 52/337/0.45/8.60176e-006 |
|     | X2 655/5624/48.14/9.97926e-006 | 42/276/0.36/8.48727e-006 |
|     | X3 1439/14609/47.84/9.95483e-006 | 58/396/0.52/9.34846e-006 |
|     | X4 1037/9446/181.69/9.43490e-006 | 4/125/0.69/0.00000e+00 |
|     | X5 58/264/22.38/9.42953e-006 | 39/257/14.28/8.20033e-006 |

[20] N. Yamashita and M. Fukushima, Modified Newton methods for solving a semismooth reformulation of monotone complementary problems, *Mathematical Programming*, 76 (1997), 469–491.

[21] Z. S. Yu, J. Sun and Y. Qin, A multivariate spectral projected gradient method for bound constrained optimization, *Journal of Computational and Applied Mathematics*, 235 (2011), 2263–2269.

[22] G. H. Yu, S. Z. Niu and J. H. Ma, Multivariate spectral gradient projection method for nonlinear monotone equations with convex constraints, *Journal of Industrial and Management Optimization*, 9 (2013), 117–129.

[23] L. Zhang and W. J. Zhou, Spectral gradient projection method for solving nonlinear monotone equations, *Journal of Computational and Applied Mathematics*, 196 (2006), 478–484.

Received April 2015; 1st revision September 2015; final revision December 2015.

E-mail address: liujinkui2006@126.com
E-mail address: lisj@cqu.edu.cn
Table 4. The results of Problem 4 with given initial points

| Dim  | Initial points | MSGP method | Algorithm 2.1 |
|------|----------------|-------------|---------------|
| 1000 | X1             | 194/1015 /0.59/9.66433e-006 | 30/184/0.05/5.70062e-006 |
|      | X2             | 117/619 /0.20/9.09560e-006 | 58/375/0.08/8.92285e-006 |
|      | X3             | 157/833 /0.24/9.74765e-006 | 75/484/0.08/6.43351e-006 |
|      | X4             | 159/838 /0.25/9.55038e-006 | 70/453/0.06/6.92855e-006 |
|      | X5             | 156/800 /0.52/9.96648e-006 | 55/354/0.06/8.13127e-006 |
| 3000 | X1             | 174/908 /3.66/9.87344e-006 | 94/610/0.19/9.67519e-006 |
|      | X2             | 164/839 /0.89/9.10283e-006 | 43/272/0.09/4.67179e-006 |
|      | X3             | 168/886 /0.98/9.48568e-006 | 75/486/0.16/8.24368e-006 |
|      | X4             | 213/1147/1.39/9.27139e-006 | 61/394/0.13/5.88525e-006 |
|      | X5             | 183/990 /3.84/9.99259e-006 | 62/396/0.13/8.46852e-006 |
| 5000 | X1             | 174/915 /6.94/9.56987e-006 | 42/267/0.13/9.61170e-006 |
|      | X2             | 163/840 /1.94/9.18817e-006 | 63/404/0.19/9.53629e-006 |
|      | X3             | 186/1014/2.59/9.83818e-006 | 67/431/0.20/9.62859e-006 |
|      | X4             | 211/1127/2.70/9.61085e-006 | 62/402/0.19/6.30592e-006 |
|      | X5             | 170/881 /9.19/9.70906e-006 | 41/261/0.14/7.03502e-006 |
| 10000| X1             | 181/969 /30.69/9.49929e-006 | 29/178/0.17/9.26476e-006 |
|      | X2             | 161/808 /7.27/8.99994e-006 | 31/193/0.19/8.57493e-006 |
|      | X3             | 182/983 /7.89/9.62444e-006 | 78/504/0.94/8.99403e-006 |
|      | X4             | 206/1069/8.33/9.78939e-006 | 40/250/0.23/9.71324e-006 |
|      | X5             | 182/972 /30.75/9.75983e-006 | 68/439/0.38/8.26113e-006 |
| 12000| X1             | 190/1007/59.33/9.43919e-006 | 52/331/0.34/8.48201e-006 |
|      | X2             | 177/920 /0.55/9.60849e-006 | 68/467/0.45/6.51248e-006 |
|      | X3             | 180/960 /11.08/9.56932e-006 | 85/549/0.55/9.43656e-006 |
|      | X4             | 200/1059/11.67/9.87597e-006 | 28/171/0.19/8.94950e-006 |
|      | X5             | 174/936 /34.61/9.79661e-006 | 70/453/0.47/7.91811e-006 |
Table 5. The results with initial points randomly generated from (0,1)

| Dim | MSGP method | Algorithm 2.1 |
|-----|-------------|---------------|
| **Problem 1** | | |
| 1000 | 10/31/0.08/1.29174e-007 | 6/19/0.03/0.00000e+000 |
| | 10/31/0.05/3.60715e-006 | 6/19/0.00/0.00000e+000 |
| | 10/31/0.05/1.31904e-007 | 6/19/0.02/0.00000e+000 |
| 3000 | 11/34/0.30/4.53840e-006 | 6/19/0.03/0.00000e+000 |
| | 11/34/0.27/6.94297e-006 | 6/19/0.02/0.00000e+000 |
| | 11/34/0.25/9.30472e-006 | 6/19/0.02/0.00000e+000 |
| 5000 | 11/34/0.72/9.94443e-006 | 6/19/0.05/0.00000e+000 |
| | 12/37/0.75/1.10770e-006 | 6/19/0.03/0.00000e+000 |
| | 12/37/0.72/9.50443e-008 | 6/19/0.03/0.00000e+000 |
| 10000 | 13/40/3.03/8.74543e-008 | 6/19/0.08/0.00000e+000 |
| | 12/37/2.67/6.56124e-006 | 6/19/0.06/0.00000e+000 |
| | 12/37/2.64/4.66099e-006 | 6/19/0.05/0.00000e+000 |
| 12000 | 13/40/4.13/1.19951e-007 | 6/19/0.09/0.00000e+000 |
| | 14/43/4.67/1.13842e-007 | 6/19/0.06/0.00000e+000 |
| | 13/40/4.13/1.98039e-007 | 6/19/0.06/0.00000e+000 |
| **Problem 2** | | |
| 1000 | 105/506/0.33/9.97726e-006 | 84/487/0.09/7.86183e-006 |
| | 104/482/0.28/9.49717e-006 | 81/465/0.06/8.31377e-006 |
| | 113/554/0.31/9.22890e-006 | 83/449/0.06/9.84927e-006 |
| 3000 | 126/650/1.88/9.29974e-006 | 91/515/0.22/9.63881e-006 |
| | 139/677/2.05/8.98283e-006 | 94/521/0.19/8.70734e-006 |
| | 133/639/1.94/9.51118e-006 | 98/532/0.22/7.45756e-006 |
| 5000 | 143/727/5.03/8.99648e-006 | 97/565/0.39/7.92677e-006 |
| | 134/692/5.02/7.94479e-006 | 100/567/0.38/8.99827e-006 |
| | 141/695/5.00/8.93255e-006 | 102/650/0.41/9.83925e-006 |
| 10000 | 154/834/20.17/9.55153e-006 | 93/545/0.70/7.86166e-006 |
| | 152/796/20.00/8.59073e-006 | 122/745/0.92/8.76303e-006 |
| | 154/794/20.20/9.43874e-006 | 127/863/0.98/9.20702e-006 |
| 12000 | 162/855/30.14/9.10526e-006 | 122/748/1.17/9.67756e-006 |
| | 163/854/30.45/9.97329e-006 | 114/805/1.16/8.33603e-006 |
| | 158/828/29.94/9.08543e-006 | 112/751/1.08/9.98871e-006 |
Table 6. The results with initial points randomly generated from (0,1)

| Problem | Dim | MSGP method | Algorithm 2.1 |
|---------|-----|--------------|---------------|
| 1000    | 248/1765/0.52/8.14403e-006 | 66/443/0.09/5.73224e-006 |
|         | 233/1659/0.44/8.60251e-006 | 73/505/0.08/8.37567e-006 |
|         | 265/1839/0.53/8.07376e-006 | 61/410/0.06/6.30228e-006 |
| 3000    | 291/2170/2.63/6.57364e-006 | 77/535/0.23/9.31672e-006 |
|         | 284/2163/2.41/9.6384e-006  | 88/612/0.23/9.82561e-006  |
|         | 287/2196/2.55/9.7633e-006   | 85/583/0.23/7.58095e-006   |
| 5000    | 293/2317/6.02/9.41072e-006  | 106/758/0.58/8.60162e-006  |
|         | 300/2320/6.19/9.59322e-006  | 89/635/0.44/8.10417e-006   |
|         | 286/2259/5.86/7.50859e-006  | 108/784/0.56/6.29653e-006  |
| 10000   | 262/2147/17.13/9.37057e-006 | 181/1544/2.89/7.61750e-006 |
|         | 296/2454/18.55/8.82342e-006 | 94/645/1.03/9.49197e-006   |
|         | 277/2187/18.78/8.03849e-006 | 65/435/0.72/5.90105e-006   |
| 12000   | 378/3238/32.92/6.12712e-006 | 82/601/1.02/9.99908e-006   |
|         | 301/2446/28.58/9.40122e-006 | 139/1096/2.63/6.13477e-006 |
|         | 300/2497/26.70/9.76692e-006 | 148/1137/2.44/8.01365e-006 |
| Problem 4| 1000| 255/1646/0.55/9.88522e-006 | 148/969/0.14/8.52252e-006 |
|         | 315/2190/0.64/8.58459e-006  | 167/1132/0.14/7.14294e-006  |
|         | 263/1779/0.51/9.78341e-006  | 152/998/0.11/6.83836e-006   |
| 3000    | 206/1864/2.59/9.68351e-006  | 204/1466/0.52/8.15543e-006  |
|         | 269/1792/2.56/9.50426e-006  | 176/1151/0.33/7.26404e-006  |
|         | 303/2094/2.95/7.70559e-006  | 180/1257/0.42/7.24683e-006  |
| 5000    | 322/2299/7.41/6.83957e-006  | 208/1469/0.91/5.80054e-006  |
|         | 265/1790/5.94/9.84065e-006  | 181/1202/0.64/9.43111e-006  |
|         | 260/2023/5.14/9.80395e-006  | 195/1315/0.77/6.31066e-006  |
| 10000   | 271/2024/20.59/8.91775e-006 | 217/1519/2.16/6.92039e-006 |
|         | 283/1997/21.91/9.98757e-006 | 216/1535/2.09/9.27616e-006 |
|         | 272/2002/19.44/8.12148e-006 | 198/1299/1.64/7.45635e-006 |
| 12000   | 218/1786/13.67/5.52117e-006 | 235/1728/3.33/7.46955e-006 |
|         | 309/2230/32.14/9.85930e-006 | 231/1779/3.78/8.69710e-006 |
|         | 248/1926/24.22/8.80377e-006 | 209/1463/2.95/8.14360e-006 |