J. Appl. Math. Comput. (2018) 56:195–216
https://doi.org/10.1007/s12190-016-1068-x

ORIGINAL RESEARCH

A new derivative-free SCG-type projection method for nonlinear monotone equations with convex constraints

Yigui Ou[1] · **Jingya Li**[1]

Received: 26 August 2016 / Published online: 4 November 2016

Abstract Based on a modified line search scheme, this paper presents a new derivative-free projection method for solving nonlinear monotone equations with convex constraints, which can be regarded as an extension of the scaled conjugate gradient method and the projection method. Under appropriate conditions, the global convergence and linear convergence rate of the proposed method is proven. Preliminary numerical results are also reported to show that this method is promising.

Keywords Nonlinear monotone equations · Scaled conjugate gradient method · Derivative-free method · Projection method · Convergence analysis

Mathematics Subject Classification 90C30 · 65K05 · 65H10

1 Introduction

In this paper, we consider the following nonlinear monotone equations:

$$F(x) = 0, \; x \in X, \tag{1.1}$$

where $X \subseteq R^n$ is a nonempty closed convex set, and $F\colon X \to R^n$ is continuous and monotone, i.e.,

$$(F(x) - F(y))^T (x - y) \geq 0, \quad \forall x, y \in X. \tag{1.2}$$

✉ Yigui Ou
ouyigui@126.com

[1] Department of Applied Mathematics, Hainan University, Haikou 570228, China

Throughout the paper, R^n denotes the n-dimensional Euclidean space of column vector, $\| \cdot \|$ denotes the Euclidean norm on R^n, and F_k denotes $F(x_k)$.

Nonlinear systems of monotone equations commonly arise in many applications, for instance, they are used as subproblems of the generalized proximal algorithms with Bregman distance [1]; Some monotone variational inequality problems can be converted into systems of nonlinear monotone equations [2]. Moreover, the equations with convex constraints come from these problems such as the economic equilibrium problems, the power flow equations and the l_1-norm regularized optimization problems in compressive sensing, see [3–5] for instance. Accordingly, numerical methods for solving problem (1.1) have been proposed by many researchers in recent years. Among these methods, Newton method, quasi-Newton method, Levenberg-Marquardt method, projection method, and their variants are very attractive because of their local superlinear convergence property from any sufficiently good initial guess, see [6–10] for instance. However, these methods are not suitable for solving large scale nonlinear monotone system of equations because they need to solve a linear system of equations at each iteration using the Jacobian matrix or an approximation of it. Therefore, the derivative-free methods for solving system of nonlinear equations have been attracting more and more attention from researchers, see [11] for instance.

The spectral gradient method, originally proposed by Barzilai and Borwein [12] for unconstrained optimization problems, has been successfully extended to solve nonlinear monotone equations by Cruz and Raydan [13,14]. An attractive feature of these methods is that they do not need the computation of the first order derivative as well as the solution of some linear equations, and thus they are suitable for solving large scale nonlinear equations. Recently, Zhang and Zhou [15] combined the spectral gradient method with the projection technique [6] to solve nonlinear monotone equations with $X = R^n$. Subsequently, Yu et al. [16] extended the method [15] to nonlinear monotone equations with convex constraints. Most recently, Yu et al. [17] established a multivariate spectral projection gradient method for nonlinear monotone equations with convex constraints.

The conjugate gradient (CG) methods (see [18] for instance) are particularly effective for solving large scale unconstrained optimization problems, due to their simplicity and low storage requirement. Thus, some researchers have tried to extend CG method to solve large-scale nonlinear monotone system of equations. For example, Cheng [19] introduced a derivative-free PRP-type method for nonlinear monotone equations with $X = R^n$, which can be viewed as an extension of the well-known PRP conjugate gradient method and the hyperplane projection method [6]. Subsequently, Li and Li [20] proposed a three-term PRP based derivative-free iterative method for solving the nonlinear monotone equations with $X = R^n$, which can be regarded as an extension of the spectral gradient method and some developed modified CG methods. Recently, Xiao and Zhu [21] developed a projected CG to the monotone nonlinear equations with convex constraints, which based on the projection technique [6] and the well-known CG-DESCENT method of Hager and Zhang [22]; Liu and Li [23] also established a multivariate spectral DY-type projection method for solving nonlinear monotone equations with convex constraints, which can be viewed as a combination of the famous DY–CG method [18] and the multivariate spectral projection gradient method [17]. It should be pointed out that these conjugate gradient-based and spectral gradient-based derivative-free methods for nonlinear monotone equations are function value-based methods and thus can be applied to solve nonsmooth equations.

In [24], Birgin and Martínez proposed a spectral conjugate gradient (SPCG) method by combining conjugate gradient method and spectral gradient method. The reported numerical results show that the SPCG method is more efficient than the CG method. Unfortunately, the SPCG method [24] cannot guarantee to generate descent directions. So in [25], Andrei proposed a new scaled conjugate gradient (SCG) algorithm for solving large scale unconstrained optimization problems using a hybridization of the memoryless BFGS preconditioned CG method suggested by Shanno [26] and the spectral CG method suggested by Birgin and Martínez [24], based on the standard secant equation (see [27] for details). Numerical comparisons show that the SCG algorithm [25] outperforms several well known CG algorithms, such as the SPCG algorithm [24], the PRP-CG algorithm (see [27] for instance) and the CG-DESCENT algorithm [22]. To the best of our knowledge, however, there is not any scaled conjugate gradient type derivative-free method for solving large-scale nonlinear monotone equations with convex constraints, based on the idea of SCG in [25].

As we know, the most computational cost of projection-based methods depends on determining the search direction and finding the stepsize. Thus, given a search direction, choosing an inexpensive line search can considerably improve the efficiency of this class of methods. For this purpose, most of practical approaches exploit an inexact line search strategy to obtain a stepsize guaranteeing the global convergence property in minimal cost. So far researchers have proposed two main line search schemes which are used in projection-based methods for solving problem (1.1), see [6, 15, 20] for instance. More precisely, given an approximation x_k and a search direction d_k, the stepsize α_k are determined respectively as follows:

LS1 Choose a stepsize $\alpha_k = \max\{\beta\rho^i : i = 0, 1, 2, \ldots\}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma\alpha_k \|d_k\|^2, \tag{1.3}$$

where $\beta > 0$ is an initial stepsize, $\rho \in (0, 1)$ and $\sigma > 0$ are two constants.

LS2 Choose a stepsize $\alpha_k = \max\{\beta\rho^i : i = 0, 1, 2, \ldots\}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma\alpha_k \|F(x_k + \alpha_k d_k)\| \|d_k\|^2. \tag{1.4}$$

Numerical experiments show that these line search schemes have different performance during solving a problem. In fact, it can be easily seen that the right hand side of LS2 will be too large when x_k is far from the solution of problem (1.1) and $\|F(x_k + \alpha_k d_k)\|$ is too large. As a result, the computing cost of line search increases. A similar case may occur for LS1 when x_k is near to the solution of problem (1.1) and $\|d_k\|$ is too large. Therefore, it is essential for us to introduce a new adaptive line search scheme which has suitable performance in both situations.

Motivated by the above observations as well as the projection technique [6], in this paper we propose a new derivative-free projection type method for solving nonlinear monotone equations with convex constraints, based on a modified line search scheme and the idea of SCG [25]. The proposed method possess some nice properties. Firstly, the proposed method is very suitable for solving large-scale problem (1.1) due to the simplicity and lower memory requirement as well as the excellent performance of computation. Secondly, it is function value-based method and thus can be applied to solve nonsmooth equations. Finally, the global convergence and linear convergence rate of the proposed method are established without the differentiability assumption on the equations

The rest of the paper is organized as follows. In Sect. 2, we propose a new derivative-free projection-based method for solving nonlinear equations with convex constraints, and then give its some properties. Section 3 is devoted to analyze the convergence properties of our algorithm under some appropriate conditions. In Sect. 4, numerical results are reported to show the efficiency of the proposed method for large scale problem (1.1). Some conclusions are summarized in the final section.

2 New method

This section is devoted to devising a new derivative-free SCG-type projection-based method for solving problem (1.1), based on a modified line search scheme and the idea of SCG [25]. Then some properties of the proposed method are given.

We first recall the iterative scheme of SCG method [25] for solving large-scale unconstrained optimization problem:

$$\min_{x\in R^n} f(x), \tag{2.1}$$

where $f\colon R^n \to R$ is a continuously differentiable function whose gradient at x_k is denoted by $g_k := \nabla f(x_k)$. Given any starting point $x_0 \in R^n$, the algorithm in [25] is to generate a sequence $\{x_k\}$ of approximations to the minimum x^* of f, in which

$$x_{k+1} = x_k + \alpha_k d_k,$$

where $\alpha_k > 0$ is a stepsize satisfying the standard Wolfe conditions (see [27] for details), and d_k is a search direction defined by

$$\begin{aligned} d_0 &= -g_0, \\ d_{k+1} &= -Q_{k+1} g_{k+1}, \quad k = 0, 1, 2, \ldots, \end{aligned} \tag{2.2}$$

with the matrix $Q_{k+1} \in R^{n\times n}$ defined by

$$Q_{k+1} = \theta_{k+1} I - \theta_{k+1} \frac{y_k s_k^T + s_k y_k^T}{y_k^T s_k} + \left[1 + \theta_{k+1} \frac{y_k^T y_k}{y_k^T s_k}\right] \frac{s_k s_k^T}{y_k^T s_k}, \tag{2.3}$$

where θ_{k+1} is a scaling parameter determined based on a two-point approximation of the standard secant equation [12], i.e.,

$$\theta_{k+1} = \frac{s_k^T s_k}{y_k^T s_k}, \tag{2.4}$$

with $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$.

Note that the BFGS update to the inverse Hessian (see [27] for details) is defined by

$$H_{k+1} = H_k - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{y_k^T s_k} + \left[1 + \frac{y_k^T H_k y_k}{y_k^T s_k}\right] \frac{s_k s_k^T}{y_k^T s_k}. \tag{2.5}$$

Therefore, we can immediately see that the matrix Q_{k+1} defined by (2.3) is precisely the self-scaling BFGS update in which the approximation of the inverse Hessian H_k is restarted as $\theta_{k+1} I$. Since the standard Wolfe conditions can ensure that $y_k^T s_k > 0$, Q_{k+1} is a symmetric positive definite matrix (see [27] for details).

Modified by the above discussions, in this paper we define the search direction d_k as follows:

$$\begin{aligned} d_0 &= -F_0, \\ d_{k+1} &= -\widetilde{Q}_{k+1} F_{k+1}, \quad k = 0, 1, 2, \ldots, \end{aligned} \tag{2.6}$$

where the matrix $\widetilde{Q}_{k+1} \in R^{n \times n}$ is defined by

$$\widetilde{Q}_{k+1} = \widetilde{\theta}_{k+1} I - \widetilde{\theta}_{k+1} \frac{w_k s_k^T + s_k w_k^T}{w_k^T s_k} + \left[1 + \widetilde{\theta}_{k+1} \frac{w_k^T w_k}{w_k^T s_k}\right] \frac{s_k s_k^T}{w_k^T s_k}, \tag{2.7}$$

with

$$\widetilde{\theta}_{k+1} = \frac{s_k^T s_k}{w_k^T s_k}, \tag{2.8}$$

and $w_k = F_{k+1} - F_k + t s_k$ $(t > 0)$.

Remark 2.1 From the definition of w_k and the monotonicity of F, it follows that

$$w_k^T s_k = (F_{k+1} - F_k)^T (x_{k+1} - x_k) + t\|s_k\|^2 \geq t\|s_k\|^2 > 0, \tag{2.9}$$

which is sufficient to ensure that $\widetilde{\theta}_{k+1}$ and d_{k+1} in (2.6) are well-defined.

It should be pointed out that in practical computation, the direction d_{k+1} in (2.6) does not actually require the matrix $\widetilde{Q}_{k+1}$ but four scalar products, i.e., it can be computed as

$$\begin{aligned} d_{k+1} = &-\widetilde{\theta}_{k+1} F_{k+1} + \widetilde{\theta}_{k+1} \left(\frac{F_{k+1}^T s_k}{w_k^T s_k} \right) w_k \\ &- \left[\left(1 + \widetilde{\theta}_{k+1} \frac{w_k^T w_k}{w_k^T s_k} \right) \frac{F_{k+1}^T s_k}{w_k^T s_k} - \widetilde{\theta}_{k+1} \frac{F_{k+1}^T w_k}{w_k^T s_k} \right] s_k. \end{aligned} \tag{2.10}$$

In what follows we review the projection approach, which was originally proposed by Solodov and Savaiter [6] for solving systems of monotone equations. Given a current iterate x_k, the projection procedure generates a trial point sequence $\{z_k\}$ such that $z_k = x_k + \alpha_k d_k$, where the stepsize $\alpha_k > 0$ is obtained using some line search along the search direction d_k such that

$$F(z_k)^T(z_k - x_k) < 0. \tag{2.11}$$

On the other hand, the monotonicity of F implies that for any $\bar{x}$ such that $F(\bar{x}) = 0$, we have

$$F(z_k)^T(z_k - \bar{x}) \geq 0. \tag{2.12}$$

This together with (2.11) implies that the hyperplane

$$H_k = \{x \in R^n | F(z_k)^T(x - z_k) = 0\} \tag{2.13}$$

strictly separates the current iterate x_k from the solution set of problem (1.1). Based on this fact, the next iterate x_{k+1} is determined by projecting x_k onto the intersection of the feasible set X with the halfspace $H_k^- = \{x \in R^n | F(z_k)^T(x - z_k) \leq 0\}$ in this way

$$x_{k+1} = P_X[x_k - \xi_k F(z_k)], \tag{2.14}$$

where $\xi_k = \frac{(x_k - z_k)^T F(z_k)}{\|F(z_k)\|^2}$.

Now, we introduce a modified line search scheme which is used to complete our projection method in this paper. Based on the above argument in Sect. 1, it seems that it is better for us to perform the scheme LS1 when the approximation x_k is far from the solution of problem (1.1) and the scheme LS2 when it is near to the solution of problem (1.1). So we introduce a new line search scheme whose main goal is trying to overcome the disadvantages of LS1 and LS2.

A modified line search scheme (MLS) Given the current approximation x_k and the direction d_k. Compute a stepsize $\alpha_k = \max\{\beta\rho^i : i = 0, 1, 2, \ldots\}$ such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \gamma_k \|d_k\|^2, \tag{2.15}$$

where $\beta > 0$ is an initial stepsize, $\rho \in (0, 1)$, $\sigma > 0$, and γ_k is defined as

$$\gamma_k = \lambda_k + (1 - \lambda_k)\|F(x_k + \alpha_k d_k)\|, \tag{2.16}$$

with $\lambda_k \in [\lambda_{\min}, \lambda_{\max}] \subseteq (0, 1]$.

Remark 2.2 Note that γ_k is a convex combination of 1 and $\|F(x_k + \alpha_k d_k)\|$. Obviously, the scheme MLS behaves like LS2 as $\lambda_k \to 0$ and LS1 as $\lambda_k \to 1$. As a result, the scheme MLS may be viewed as a modified version of LS1 and LS2. In numerical experiments, we can select λ_k adaptively in order to increase its value as x_k being far from the solution and decrease its value as x_k being near to the solution. For example, we may update λ_k by the following recursive formula

$$\lambda_k = \begin{cases} \frac{\lambda_0}{2}, & k = 1, \\ \frac{\lambda_{k-1}+\lambda_{k-2}}{2}, & k \geq 2, \end{cases} \tag{2.17}$$

with $\lambda_0 = 1$.

To describe our algorithm, we introduce the definition of projection operator $P_\Omega[\cdot]$ which is defined as a mapping from R^n to a nonempty closed convex subset Ω:

$$P_\Omega[x] = \arg\min_{y \in \Omega}\{\|y - x\|\}, \ \forall x \in R^n. \tag{2.18}$$

A well-known property of this operator is that it is nonexpansive (see [6]), namely,

$$\|P_\Omega[x] - P_\Omega[y]\| \leq \|x - y\|, \quad \forall x, y \in R^n. \tag{2.19}$$

Now we are ready to state the steps of our algorithm for solving problem (1.1) as follows.

Algorithm 2.1

Step 0 Given an initial point x_0, $\epsilon \geq 0$, $\sigma > 0$, $\rho \in (0, 1)$, $t \in (0, 1)$, and an initial trial stepsize $\beta > 0$. Set $k := 0$.
Step 1 Compute F_k. If $\|F_k\| \leq \epsilon$, stop.
Step 2 Construct a search direction d_k by using (2.6)–(2.8).
Step 3 Compute the trial point $z_k = x_k + \alpha_k d_k$, where α_k is determined by the scheme MLS (2.15).
Step 4 If $\|F(z_k)\| \leq \epsilon$, stop. Otherwise, compute the new iterate x_{k+1} by (2.14).
Step 5 Set $k := k + 1$, and go to Step 1.

Remark 2.3 From Lemma 3.2 below, it follows that the line search scheme MLS (2.15) is well defined, namely, it terminates in a finite number of steps.

Lemma 2.1 *The matrix $\widetilde{Q}_{k+1}$ defined by (2.7)–(2.8) is symmetric positive definite.*

Proof Obviously, $\widetilde{Q}_{k+1}$ is a symmetric matrix. Furthermore, it follows from (2.8) and (2.9) that $\widetilde{\theta}_{k+1} > 0$. Since $\widetilde{Q}_{k+1}$ is precisely the self-scaling BFGS update where H_k in (2.5) is restarted as $\widetilde{\theta}_{k+1} I$ at every step, it is positive definite matrix (see [27] for details). This completes the proof. □

Remark 2.4 Since $\widetilde{Q}_{k+1}$ is symmetric positive definite, and thus it is nonsingular. Using the Sherman-Morrison Theorem (see [25] for instance), it can be shown that the matrix $\widetilde{P}_{k+1}$ defined by

$$\widetilde{P}_{k+1} = \frac{1}{\widetilde{\theta}_{k+1}} I - \frac{1}{\widetilde{\theta}_{k+1}} \frac{s_k s_k^T}{s_k^T s_k} + \frac{w_k w_k^T}{w_k^T s_k} \tag{2.20}$$

is the inverse of $\widetilde{Q}_{k+1}$. Hence, $\widetilde{P}_{k+1}$ is also a symmetric positive definite matrix.

Lemma 2.2 *The direction d_k defined by (2.6)–(2.8) satisfies $F_k^T d_k < 0$ for any k. Furthermore, if the mapping F is Lipschitz continuous, then there exists a constant $c > 0$ such that*

$$F_k^T d_k \leq -c\|F_k\|^2. \tag{2.21}$$

Proof For $d_0 = -F_0$, we have $F_0^T d_0 = -\|F_0\|^2 < 0$. Furthermore, by multiplying (2.6) by F_{k+1}^T, we get

$$F_{k+1}^T d_{k+1} = \frac{1}{\left(w_k^T s_k\right)^2}\Big[-\widetilde{\theta}_{k+1}\|F_{k+1}\|^2\left(w_k^T s_k\right)^2 + 2\widetilde{\theta}_{k+1}\left(F_{k+1}^T w_k\right)\left(F_{k+1}^T s_k\right)\left(w_k^T s_k\right) \\ -\left(F_{k+1}^T s_k\right)^2\left(w_k^T s_k\right) - \widetilde{\theta}_{k+1}\left(w_k^T w_k\right)\left(F_{k+1}^T s_k\right)^2\Big].$$

Applying the inequality $2a^T b \leq \|a\|^2 + \|b\|^2$ to the second term in the right-hand side of the above equality with $a = (w_k^T s_k)F_{k+1}$ and $b = (F_{k+1}^T s_k)w_k$, we get

$$F_{k+1}^T d_{k+1} \leq -\frac{\left(F_{k+1}^T s_k\right)^2}{\left(w_k^T s_k\right)},$$

which, together with (2.9), implies that $F_k^T d_k < 0$ for $k \geq 0$. So, we prove the first part of this lemma.

For the second part of the proof, please see Lemma 3.1 next section. This completes the proof. □

Remark 2.5 It should be mentioned that if F is the gradient of some real-valued function $f\colon R^n \to R$, the condition (2.21) means that d_k is a sufficiently descent direction of f at x_k, which plays an important role in analyzing the global convergence of CG methods.

3 Convergence analysis

In this section, we analyze the global convergence and linear convergence rate of Algorithm 2.1 when it is applied to problem (1.1). For this purpose, we first make the following assumptions.

A1 The solution set X^* of problem (1.1) is nonempty.

A2 The mapping $F\colon X \to R^n$ is Lipschitz continuous, namely, there exists a positive constant $L > 0$ such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in X. \tag{3.1}$$

In what follows we assume that $F_k \neq 0$ and $F(z_k) \neq 0$ for all k, namely, Algorithm 2.1 generate two infinite sequences $\{x_k\}$ and $\{z_k\}$. Otherwise, we obtain a solution of problem (1.1).

Lemma 3.1 *Suppose that Assumption* A2 *holds true. Then there exist two positive constants* $c_1 > 0$ *and* $c_2 > 0$ *such that*

$$F_k^T d_k \le -c_1 \|F_k\|^2, \quad \forall k, \tag{3.2}$$

and

$$\|d_k\| \le c_2 \|F_k\|, \quad \forall k. \tag{3.3}$$

Proof By computing the trace of $\widetilde{Q}_{k+1}$ and $\widetilde{P}_{k+1}$ defined respectively by the update formulas (2.7)–(2.8) and (2.20), we can easily obtain that

$$\text{trace}(\widetilde{Q}_{k+1}) = (n-2)\widetilde{\theta}_{k+1} + \left(1 + \widetilde{\theta}_{k+1} \frac{\|w_k\|^2}{w_k^T s_k}\right)\widetilde{\theta}_{k+1}, \tag{3.4}$$

and

$$\text{trace}(\widetilde{P}_{k+1}) = \frac{n-1}{\widetilde{\theta}_{k+1}} + \frac{\|w_k\|^2}{w_k^T s_k}. \tag{3.5}$$

From Assumption A2 and the definition of w_k, it follows that

$$\|w_k\| \le (L+t)\|s_k\|. \tag{3.6}$$

This together with (2.9) implies that

$$\widetilde{\theta}_{k+1} = \frac{\|s_k\|^2}{w_k^T s_k} \le \frac{1}{t}, \tag{3.7}$$

$$\frac{\|w_k\|^2}{w_k^T s_k} \le \frac{(L+t)^2}{t}, \tag{3.8}$$

and

$$\frac{1}{\widetilde{\theta}_{k+1}} = \frac{w_k^T s_k}{\|s_k\|^2} \le \frac{\|w_k\|^2}{w_k^T s_k} \le \frac{(L+t)^2}{t}. \tag{3.9}$$

Combining the above three inequalities with (3.4) and (3.5) yields

$$\text{trace}(\widetilde{Q}_{k+1}) \le \frac{(n-1)t^2 + (L+t)^2}{t^3}, \tag{3.10}$$

and

$$\operatorname{trace}(\widetilde{P}_{k+1}) \leq \frac{n(L+t)^2}{t}. \tag{3.11}$$

Now, we can verify the validity of (3.2) and (3.3). For $k = 0$, $d_0 = -F_0$. Then

$$F_0^T d_0 = -\|F_0\|^2, \ \|d_0\| = \|F_0\|. \tag{3.12}$$

For $k \geq 0$, using (3.10) and (3.11), we have

$$\begin{aligned} -F_{k+1}^T d_{k+1} &= F_{k+1}^T \widetilde{Q}_{k+1} F_{k+1} \\ &\geq \frac{\|F_{k+1}\|^2}{\|\widetilde{P}_{k+1}\|} \\ &\geq \frac{\|F_{k+1}\|^2}{\operatorname{trace}(\widetilde{P}_{k+1})} \\ &\geq \frac{t}{n(L+t)^2}\|F_{k+1}\|^2, \end{aligned} \tag{3.13}$$

and

$$\begin{aligned} \|d_{k+1}\| &\leq \|\widetilde{Q}_{k+1}\|\|F_{k+1}\| \\ &\leq \operatorname{trace}(\widetilde{Q}_{k+1})\|F_{k+1}\| \\ &\leq \frac{(n-1)t^2+(L+t)^2}{t^3}\|F_{k+1}\|. \end{aligned} \tag{3.14}$$

By letting $c_1 = \min\{1, \frac{t}{n(L+t)^2}\}$ and $c_2 = \max\{1, \frac{(n-1)t^2+(L+t)^2}{t^3}\}$, then the desired inequalities (3.2) and (3.3) follow immediately from (3.12), (3.13) and (3.14). This completes the proof. □

Remark 3.1 From the Cauchy-Schwarz inequality and (3.2), it follows that

$$\|d_k\| \geq c_1\|F_k\|, \ \forall k. \tag{3.15}$$

So, $F_k \neq 0$ for all k implies that $d_k \neq 0$ for all $k > 0$. Otherwise, a solution of problem (1.1) is found.

The following lemma shows that Algorithm 2.1 is well defined.

Lemma 3.2 *Suppose that Assumption* A2 *holds true, then there exists a nonnegative number i_k satisfying the line search scheme MLS* (2.15) *for all k.*

Proof By contradiction, suppose that there exists $k_0 \geq 0$ such that the scheme MLS (2.15) fails to hold for any nonnegative number i, namely

$$-F(x_{k_0} + \beta\rho^i d_{k_0})^T d_{k_0} < \sigma\beta\rho^i\gamma_{k_0}\|d_{k_0}\|^2, \ \forall i \geq 0, \tag{3.16}$$

where γ_{k_0} is defined by (2.16). It is clear that

$$\lambda_{\min} \leq \gamma_{k_0} \leq \max\{1, \|F(x_{k_0} + \beta\rho^i d_{k_0})\|\}. \tag{3.17}$$

Taking $i \to +\infty$ on both sides of (3.16) and using the continuity of F, we have

$$-F(x_{k_0})^T d_{k_0} \leq 0,$$

which contracts this fact that $-F_k^T d_k \geq c_1\|F_k\|^2 > 0$ for all k. This completes the proof. □

Lemma 3.3 *Suppose that Assumptions* A1 *and* A2 *hold true. Let the sequences* $\{x_k\}$ *and* $\{z_k\}$ *be generated by Algorithm* 2.1*. Then, for any* $x^* \in X^*$*, there exists a constant* $c_3 > 0$ *such that*

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \frac{c_3\|x_k - z_k\|^4}{\|F(z_k)\|^2}, \tag{3.18}$$

and

$$\lim_{k\to+\infty} \|x_k - z_k\| = \lim_{k\to+\infty} \alpha_k\|d_k\| = 0, \tag{3.19}$$

Furthermore, both $\{x_k\}$ *and* $\{z_k\}$ *are bounded.*

Proof From the line search scheme MLS (2.15), it follows that

$$F(z_k)^T(x_k - z_k) = -\alpha_k F(z_k)^T d_k \geq \sigma\alpha_k^2\gamma_k\|d_k\|^2, \tag{3.20}$$

where γ_k is defined by (2.16). By (3.20) and $\gamma_k \geq \lambda_{\min} > 0$ for all k, we get

$$F(z_k)^T(x_k - z_k) \geq \sigma\lambda_{\min}\alpha_k^2\|d_k\|^2 = \sigma\lambda_{\min}\|x_k - z_k\|^2. \tag{3.21}$$

Using the monotonicity of mapping F and $F(x^*) = 0$, we have

$$F(z_k)^T(z_k - x^*) \geq F(x^*)^T(z_k - x^*) = 0. \tag{3.22}$$

This together with (3.20) implies that

$$F(z_k)^T(x_k - x^*) \geq F(z_k)^T(x_k - z_k) \geq 0. \tag{3.23}$$

Combining this inequality with (2.19) and (3.21) gives

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \left\| P_X\left[x_k - \frac{(x_k - z_k)^T F(z_k)}{\|F(z_k)\|^2}F(z_k)\right] - P_X[x^*]\right\|^2 \\
&\leq \left\| x_k - \frac{(x_k - z_k)^T F(z_k)}{\|F(z_k)\|^2}F(z_k) - x^*\right\|^2 \\
&\leq \|x_k - x^*\|^2 - \frac{[F(z_k)^T(x_k - z_k)]^2}{\|F(z_k)\|^2} \\
&\leq \|x_k - x^*\|^2 - \frac{\sigma^2\lambda_{\min}^2\|x_k - z_k\|^4}{\|F(z_k)\|^2}.
\end{aligned} \tag{3.24}$$

Hence, this inequality (3.18) holds true, where $c_3 = \sigma^2\lambda_{\min}^2$.

Similarly to the proof of Lemma 3.2 in Yu et al [16], we can prove the remaining conclusions of this lemma, and we therefore omit it. This completes the proof. □

Using these lemmas mentioned above, we can obtain the global convergence property of Algorithm 2.1 as follows.

Theorem 3.4 *Suppose that Assumptions* A1 *and* A2 *hold true. Let the sequence* $\{x_k\}$ *be generated by Algorithm* 2.1*. Then*

$$\lim_{k\to+\infty} \inf \|F_k\| = 0. \tag{3.25}$$

Furthermore, the sequence $\{x_k\}$ *converges to a solution of problem* (1.1).

Proof The proof is similar to that of Theorem 3.1 in Yu et al [16], and thus we omit it here. This completes the proof. □

Now, we will analyze the linear convergence rate of Algorithm 2.1. For this purpose, we also need the following assumption.

A3 For any $x^* \in X^*$, there exist two positive constants $c_4 > 0$ and $c_5 > 0$ such that

$$c_4 \mathrm{dist}(x, X^*) \le \|F(x)\|, \ \forall x \in N(x^*, c_5), \tag{3.26}$$

where $\mathrm{dist}(x, X^*)$ denotes the distance from x to the solution set X^*, and $N(x^*, c_5) := \{x \in R^n | \|x - x^*\| \le c_5\}$.

Remark 3.2 The condition (3.26) is called as the local error bound assumption which is weaker than the nonsingularity condition of Jacobi matrix $F'(x)$ (if it exists), and is usually used to analyze the convergence rate of some algorithms for solving nonlinear systems of equations, see [7,9] and references therein, for instance. Note that Assumption A3 here is different from Assumption A2 in [23].

To analyze the convergence rate of Algorithm 2.1, we need to investigate the lower boundary of the stepsize α_k in Step 3 of Algorithm 2.1.

Lemma 3.5 *Suppose that Assumptions* A1 *and* A2 *hold true. Then there exists a constant* $c_6 > 0$ *such that*

$$\alpha_k \ge \min\left\{\beta, \ c_6 \frac{\|F_k\|^2}{\|d_k\|^2}\right\}, \ \forall k. \tag{3.27}$$

Proof If $\alpha_k \ne \beta$, then it follows from the acceptance rule of stepsize α_k in Step 3 of Algorithm 2.1 that $\alpha_k' = \frac{\alpha_k}{\rho}$ does not satisfy (2.15), namely

$$-F(x_k + \alpha_k' d_k)^T d_k < \sigma \alpha_k' \gamma_k' \|d_k\|^2, \tag{3.28}$$

where

$$\gamma_k' = \lambda_k + (1 - \lambda_k)\|F(x_k + \alpha_k' d_k)\| \le \max\{1, \|F(x_k + \alpha_k' d_k)\|\}. \tag{3.29}$$

From (3.19), the boundedness of $\{x_k\}$ and the continuity of F, we conclude that there exists a constant $c_7 > 0$ such that

$$\|F(x_k + \alpha'_k d_k)\| \le c_7, \ \forall k. \tag{3.30}$$

Combining (3.28) with (3.29) and (3.30) yields

$$-F(x_k + \alpha'_k d_k)^T d_k < \sigma c_8 \alpha'_k \|d_k\|^2, \tag{3.31}$$

where $c_8 = \max\{1, c_7\}$. So, by A2, (3.2) and (3.31), we get

$$\begin{aligned} c_1\|F_k\|^2 &\le -F_k^T d_k \\ &= [F(x_k + \alpha'_k d_k) - F_k]^T d_k - F(x_k + \alpha'_k d_k)^T d_k \\ &\le L\alpha'_k\|d_k\|^2 + \sigma c_8 \alpha'_k \|d_k\|^2 \\ &= (L + \sigma c_8)\alpha_k \rho^{-1}\|d_k\|^2, \ \forall k. \end{aligned}$$

This inequality together with Remark 3.1 implies that

$$\alpha_k \ge \frac{\rho c_1}{L + \sigma c_8} \frac{\|F_k\|^2}{\|d_k\|^2}, \ \forall k, \tag{3.32}$$

which means that the inequality (3.27) holds true, where $c_6 = \frac{\rho c_1}{L+\sigma c_8}$. This completes the proof. □

In the remainder of this paper, we always assume that $x_k \to x^*$ as $k \to \infty$ due to the conclusion of Theorem 3.4, where $x^* \in X^*$.

Theorem 3.6 *Suppose that Assumptions* A1*,* A2 *and* A3 *hold true. Let the sequence* $\{x_k\}$ *be generated by Algorithm* 2.1*. Then the sequence* $\{dist(x_k, X^*)\}$ *Q-linearly converges to 0.*

Proof Let $\bar{x}_k := arg\min\{\|x_k - x\|: \ x \in X^*\}$, which implies that $\bar{x}_k$ is the closest solution to x_k, namely,

$$\|x_k - \bar{x}_k\| = \text{dist}(x_k, X^*). \tag{3.33}$$

Denote x^* by $\bar{x}_k$, then it follows from (3.18) that

$$\|x_{k+1} - \bar{x}_k\|^2 \le \|x_k - \bar{x}_k\|^2 - \frac{c_3\|x_k - z_k\|^4}{\|F(z_k)\|^2}, \tag{3.34}$$

By (3.3) and (3.27), we have

$$\alpha_k \ge \min\left\{\beta, \ \frac{c_6}{c_2^2}\right\} := c_9, \ \forall k. \tag{3.35}$$

This together with (3.15) and A3 implies that

$$\begin{aligned}\|x_k - z_k\| &= \alpha_k \|d_k\| \\ &\geq c_9 c_1 \|F_k\| \\ &\geq c_9 c_1 c_4 \text{dist}(x_k, X^*) \\ &= c_{10} \text{dist}(x_k, X^*),\end{aligned} \tag{3.36}$$

where $c_{10} = c_9 c_1 c_4$. Furthermore, by (3.33), (3.3), A2 and $\alpha_k \leq \beta$ for all k, we obtain

$$\begin{aligned}\|F(z_k)\| &= \|F(z_k) - F(\bar{x}_k)\| \\ &\leq L\|z_k - \bar{x}_k\| \\ &\leq L(\|z_k - x_k\| + \|x_k - \bar{x}_k\|) \\ &= L(\alpha_k \|d_k\| + \|x_k - \bar{x}_k\|) \\ &\leq L(\beta c_2 \|F_k\| + \|x_k - \bar{x}_k\|) \\ &\leq L(\beta c_2 L \|x_k - \bar{x}_k\| + \|x_k - \bar{x}_k\|) \\ &= L(\beta c_2 L + 1)\|x_k - \bar{x}_k\| \\ &\leq c_{11} \text{dist}(x_k, X^*),\end{aligned} \tag{3.37}$$

where $c_{11} = \max\{L(\beta c_2 L + 1), \sqrt{c_3} c_{10}^2\}$. Combining (3.37) with (3.36) and (3.34) yields

$$\begin{aligned}\text{dist}(x_{k+1}, X^*)^2 &\leq \|x_{k+1} - \bar{x}_k\|^2 \\ &\leq \text{dist}(x_k, X^*)^2 - \frac{c_3 c_{10}^4 \text{dist}(x_k, X^*)^4}{c_{11}^2 \text{dist}(x_k, X^*)^2} \\ &= \left(1 - \frac{c_3 c_{10}^4}{c_{11}^2}\right) \text{dist}(x_k, X^*)^2,\end{aligned}$$

which implies that the sequence $\{\text{dist}(x_k, X^*)\}$ Q-linearly converges to 0. This proof is completed. □

Remark 3.3 In view of Theorem 3.6, we know that the distance $\text{dist}(x_k, X^*)$ from the iterates x_k to the solution set X^* converges to zero with a Q-linear rate. However, this says little about the sequence $\{x_k\}$ itself. Whether is it also linearly convergent? We will answer this question as follows.

To investigate the local behaviour of the sequence $\{x_k\}$ generated by Algorithm 2.1, we need the following definition [27]:

Definition 3.7 The mapping $F\colon X \to R^n$ is called strongly monotone with modulus $\mu > 0$, if the inequality

$$(F(x) - F(y))^T (x - y) \geq \mu \|x - y\|^2$$

holds true for all $x, y \in X$.

Theorem 3.8 *Suppose that Assumptions* A1 *and* A2 *hold true. If the mapping F is strongly monotone with modulus $\mu > 0$, then the sequence $\{x_k\}$ generated by Algorithm* 2.1 *R-linearly converges to x^*.*

Proof Clearly, it follows from the Cauchy-Schwarz inequality and the strong monotonicity assumption of F that

$$\|F_k\| = \|F_k - F(x^*)\| \geq \mu\|x_k - x^*\|. \tag{3.38}$$

This together with (3.15) and (3.35) implies that

$$\begin{aligned}\|x_k - z_k\| &= \alpha_k\|d_k\| \\ &\geq c_9c_1\|F_k\| \\ &\geq \mu c_9c_1\|x_k - x^*\| \\ &= c_{12}\|x_k - x^*\|,\end{aligned} \tag{3.39}$$

where $c_{12} = \mu c_9c_1$. Moreover, similarly to the proof of (3.37), we can conclude that

$$\|F(z_k)\| \leq L(\beta c_2L + 1)\|x_k - x^*\| \leq c_{13}\|x_k - x^*\|, \tag{3.40}$$

where $c_{13} = \max\{L(\beta c_2L + 1), \sqrt{c_3}c_{12}^2\}$. Combining (3.40) with (3.39) and (3.18) gives

$$\begin{aligned}\|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - \frac{c_3c_{12}^4\|x_k - x^*\|^4}{c_{13}^2\|x_k - x^*\|^2} \\ &= \left(1 - \frac{c_3c_{12}^4}{c_{13}^2}\right)\|x_k - x^*\|^2,\end{aligned}$$

which implies that the whole sequence $\{x_k\}$ R-linearly converges to x^*. This proof is completed. □

Remark 3.4 It should be mentioned that the R-linear convergence of the sequence $\{x_k\}$ generated by Algorithm 2.1 is established under the strong monotonicity assumption of F. This assumption seems to be a little strong. However, it may be removed if the solution set X^* of problem (1.1) is unique.

4 Numerical tests

In this section, we present some numerical experiments to evaluate the performance of the proposed method on a set of test problems with different initial points. At the same time, we give some comparisons with the related algorithms, including the performance profiles of Dolan and More [28].

The test problems are listed as follows:

Problem 1 The elements of function F are given by $F_i(x) = exp(x_i) - 1,\ i = 1, 2, \dots, n$, and $X = \{x \in R^n | x_i \geq 0, \quad i = 1, 2, \dots, n\}$.

Problem 2 The elements of function F are given by $F_i(x) = x_i - sin(|x_i - 1|),\ i = 1, 2, \dots, n$, and $X = \{x \in R^n | \sum_{i=1}^n x_i \leq n,\ x_i \geq 0,\ i = 1, 2, \dots, n\}$.

Obviously, this problem is nonsmooth at point $(1, 1, \dots, 1)^T \in R^n$.

Problem 3 The elements of function F are given by

$$F_1(x) = x_1 - exp\left(cos\left(\frac{x_1 + x_2}{n+1}\right)\right),$$
$$F_i(x) = x_i - exp\left(cos\left(\frac{x_{i-1} + x_i + x_{i+1}}{n+1}\right)\right), \quad i = 2, 3, \dots, n-1,$$
$$F_n(x) = x_n - exp\left(cos\left(\frac{x_{n-1} + x_n}{n+1}\right)\right),$$

and $X = \{x \in R^n | x_i \geq 0,\ i = 1, 2, \dots, n\}$.

Problem 4 The elements of function F are given by $F_i(x) = ln(|x_i| + 1) - \frac{x_i}{n},\ i = 1, 2, \dots, n$, and $X = \{x \in R^n | x_i \geq 0,\ i = 1, 2, \dots, n\}$.
Obviously, this problem is nonsmooth at point $(0, 0, \dots, 0)^T \in R^n$.

Problem 5 The elements of function F are given by

$$F_1(x) = exp(x_1) - 1,$$
$$F_i(x) = exp(x_i) + x_{i-1} - 1, \quad i = 2, 3, \dots, n,$$

and $X = \{x \in R^n | x_i \geq 0,\ i = 1, 2, \dots, n\}$.

Problem 6 The elements of function F are given by

$$F_1(x) = 2x_1 - x_2 + exp(x_1) - 1,$$
$$F_i(x) = -x_{i-1} + 2x_i - x_{i+1} + exp(x_i) - 1, \quad i = 2, 3, \dots, n-1,$$
$$F_n(x) = -x_{n-1} + 2x_n + exp(x_n) - 1,$$

and $X = \{x \in R^n | x_i \geq 0,\ i = 1, 2, \dots, n\}$.

Problem 7 The elements of function F are given by

$$F_1(x) = 2.5x_1 + x_2 - 1,$$
$$F_i(x) = x_{i-1} + 2.5x_i + x_{i+1} - 1, \quad i = 2, 3, \dots, n-1,$$
$$F_n(x) = x_{n-1} + 2.5x_n - 1,$$

and $X = \{x \in R^n | x_i \geq 0,\ i = 1, 2, \dots, n\}$.

Problem 8 The function F is given by

$$F(x) = Ax + g(x),$$

where $g(x) = (exp(x_1) - 1, exp(x_2) - 1, \ldots, exp(x_n) - 1)^T$, and

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}$$

The different initial points are listed as follows:

$$x1 = (10, 10, \ldots, 10)^T, \quad x2 = -(10, 10, \ldots, 10)^T, \quad x3 = (1, \tfrac{1}{2}, \ldots, \tfrac{1}{n}),$$
$$x4 = (0.1, 0.1, \ldots, 0.1)^T, \quad x5 = (\tfrac{1}{n}, \tfrac{2}{n}, \ldots, 1)^T, \quad x6 = (1 - \tfrac{1}{n}, 1 - \tfrac{2}{n}, \ldots, 0)^T,$$

We first present some numerical experiments to compare **Algorithm 2.1** (i.e., α_k is determined by the new MLS (2.15)) with **Algorithm 1** (i.e., α_k is determined by LS1 (1.3)) and **Algorithm 3** (i.e., α_k is determined by LS2(1.4)). We implemented all the procedures with the number of variables $n = 10{,}000, 50{,}000$ and $100{,}000$. The codes were written in Matlab 7.0 and run on a PC computer (CPU 2.60 GHZ, 2.00 GB memory) with Windows operating system. Throughout the computational experiments, the parameters used in these algorithms are chosen as follows: $\rho = 0.5$, $\sigma = 0.001$, $t = 0.01$, and the initial trial stepsize β is computed by [20]

$$\beta = \frac{F_k^T d_k}{d_k^T (F(x_k + \delta d_k) - F_k)/\delta},$$

with $\delta = 10^{-8}$. If the above parameter β satisfies $\beta \le 10^{-6}$, then $\beta = 1$. We stop the iteration if the following condition

$$\|F_k\| \le 10^{-5} \text{ or } \|F(z_k)\| \le 10^{-5}$$

is satisfied. We also terminate the iteration when it exceeds the preset iteration limit 5000.

We use the performance profile proposed by Dolan and More [28] to display the performance of each implementation on the set of test problems. The performance profile has some advantages over other existing benchmarking tools, especially for large test sets where tables tend to be overwhelming. Let $l_{p,s}$ denote the number of iterations required to solve problem p by solver s. Define the performance ratio as

$$r_{p,s} = \frac{l_{p,s}}{l_p^*}$$

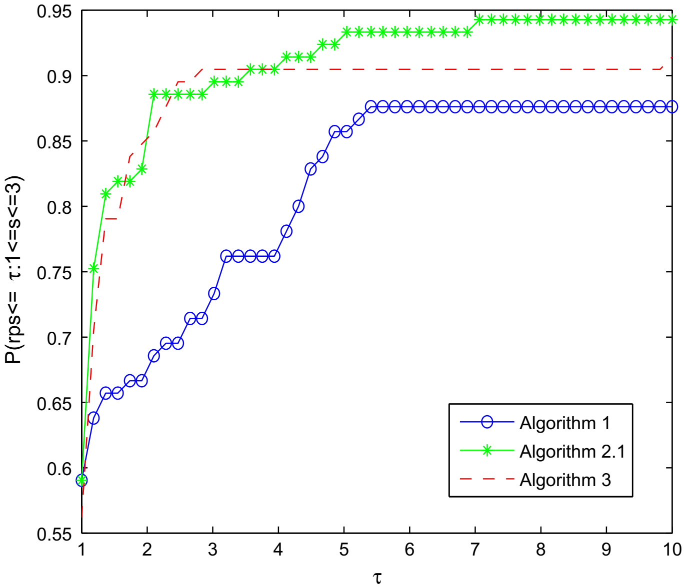

Fig. 1 Performance profile for the number of iterations

where l_p^* is the smallest number of iterations required by any solver to solve problem p. Therefore, $r_{p,s} \geq 1$ for all p and s. If a solver does not solve a problem, the ratio $r_{p,s}$ is assigned a large number M, which satisfies $r_{p,s} < M$ for all p and s, where solver s succeeds in solving problem p. Then the performance profile for each solver s is defined as the cumulative distribution function for the performance ratio $r_{p,s}$, which is

$$P_s(\tau) = \frac{\text{No. of problems s.t. } r_{p,s} \leq \tau}{\text{Total no. of problems}}$$

Obviously, $P_s(1)$ represents the probability that the solver will win over the rest of the solvers (has the higher probability of being the optimal solver), in terms of the number of iterations. For more details about the performance profile, please see [28]. The performance profile will also be used to analyze the CPU time or others required.

Figures 1 and 2 give the performance profiles of the three line search schemes for the number of iterations and the CPU time, respectively. From Figs. 1 and 2, we observe the facts: for these test problems mentioned above, Algorithm 2.1 performs better than Algorithms 1 and 3 in terms of the number of iterations and the CPU time. Therefore, we could say that the line search scheme MLS (2.15) is more effective than the line search schemes LS1(1.3) and LS2(1.4) in terms of the computational effort.

To validate Algorithm 2.1 from a computational point of view, in the second set of numerical experiments, we compare the performance of Algorithm 2.1 with the algorithm in [16] (abbreviated **Algorithm YZS**), the algorithm in [21] (abbreviated **Algorithm XYH**), and the algorithm in [23] (abbreviated **Algorithm LJK**) for the same problems mentioned above, where the latter three algorithms were devised espe-

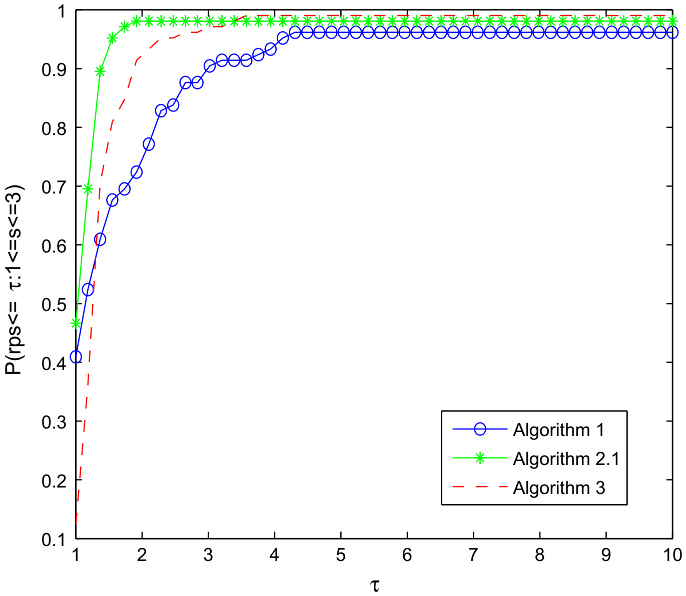

Fig. 2 Performance profile for the CPU time

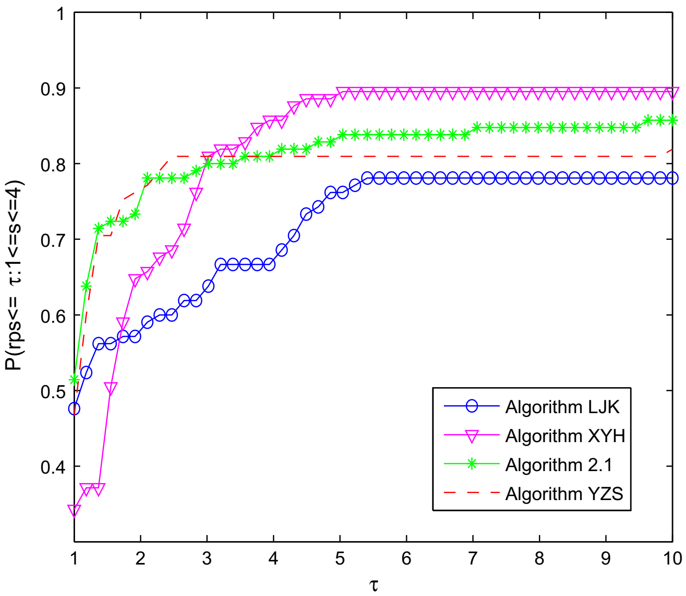

Fig. 3 Performance profile for the number of iterations

cially for problem (1.1). Throughout the computational experiments, the parameters used in Algorithm 2.1 are chosen to be the same as that mentioned above; the parameters used in Algorithm YZS are chosen as follows: $\beta = 0.5,\ \sigma = 0.01,\ r = 0.01$;

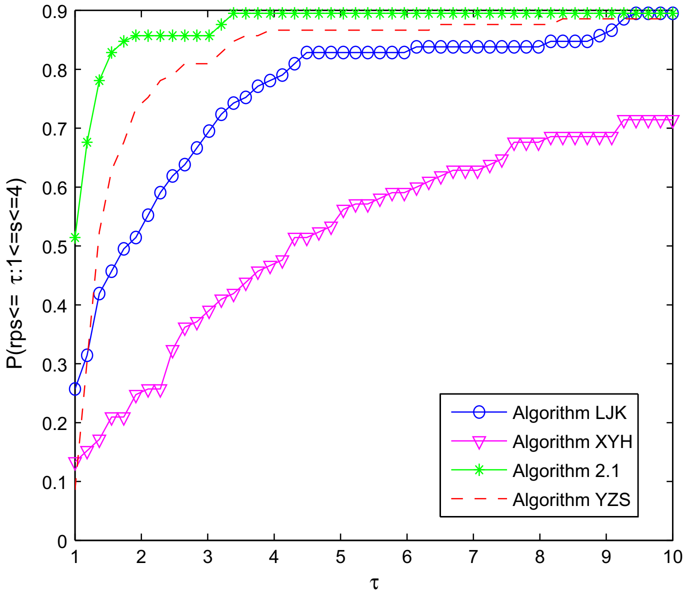

Fig. 4 Performance profile for the CPU time

the parameters used in Algorithm XYH are chosen as follows: $\xi = 1,\ \rho = 0.5,\ \sigma = 0.0001$, while the parameters used in Algorithm LJK are the same as that in [23] . Moreover, the stopping condition for each algorithm is the same as that mentioned above.

We also use the performance profile proposed by Dolan and More [28] to display the performance of each implementation on the set of test problems with the number of variables $n = 10{,}000,\ 50{,}000,\ 100{,}000$. Figures 3 and 4 give the performance profiles of the four algorithms for the number of iterations and the CPU time, respectively.

From Figs. 3 and 4, we observe the facts: for these test problems mentioned above, Algorithm 2.1 is the best one among the four solvers, in terms of the number of iterations and the CPU time.

While it would be unwise to draw any firm conclusions from the limited numerical results and comparisons, they indicate some promise for the new approaches proposed in this paper, compared with the related methods mentioned above. Further improvement is expected from more suitable implementation.

5 Conclusion

In this paper we propose a new derivative-free projection type method for solving nonlinear monotone equations with convex constraints, The main properties of the proposed methods are that we establish the global convergence and local linear convergence rate without using any merit function and making the differentiability assumption on the equations. Furthermore, these methods do not solve any subproblem and store any matrix, and thus can be applied to solve large-scale nonlinear equations. Preliminary numerical results are reported to show that the proposed method is promising, especially for the large-scale problems with convex constraints.

Since the most computational cost of each algorithm for problem (1.1) is to determine the search direction d_k and find the stepsize α_k in line search, we will study some more effective methods for defining an appropriate search direction d_k and an inexpensive line search in our future research. Furthermore, we will continue to investigate the local behaviour of the sequence $\{x_k\}$ generated by Algorithm 2.1 under some weaker conditions.

Acknowledgements The authors would like to thank the anonymous referees and the editor for their patience and valuable comments and suggestions that greatly improved this paper. This work is partially supported by NNSF of China (No. 11261015) and NSF of Hainan Province (No. 2016CXTD004, No. 111001).

References

1. Iusem, A.N., Solodov, M.V.: Newton-type methods with generalized distance for constrained optimization. Optimization **41**, 257–278 (1997)
2. Zhao, Y.B., Li, D.: Monotonicity of fixed point and normal mapping associated with variational inequality and is applications. SIAM J. Optim. **11**, 962–973 (2001)
3. Dirkse, S.P., Ferris, M.C.: MCPLIB: a collection of nonlinear mixed complementarity problems. Optim. Methods Softw. **5**, 319–345 (1995)
4. Wood, A.J., Wollenberg, B.F.: Power Generations, Operations, and Control. Wiley, New York (1996)
5. Figueiredo, M., Nowak, R., Wright, S.J.: Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Topics Signal Process. **1**, 586–597 (2007)
6. Solodov, M.V., Svaiter, B.F.: A globally convergent inexact Newton method for system of monotone equations. In: Fukushima, M., Qi, L. (eds.) Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, pp. 355–369. Kluwer Academic Publishers, Dordrecht (1999)
7. Fan, J.Y., Yuan, Y.X.: A regularized Newton method for monotone nonlinear equations and its application. Optim. Methods Softw. **29**, 102–119 (2014)
8. Zhou, W.J., Li, D.H.: A globally convergent BFGS method for solving nonlinear monotone equations without any merit functions. Math. Comput. **77**, 2231–2240 (2008)
9. He, Y.D., Ma, C.F., Fan, B.: A corrected Levenberg–Marquardt algorithm with a nonmonotone line search for the system of nonlinear equations. Appl. Math. Comput. **260**, 159–169 (2015)
10. Wang, C.W., Wang, Y.J.: A superlinearly convergent projection method for constrained systems of nonlinear equations. J. Global Optim. **44**, 283–296 (2009)
11. Huang, N., Ma, C.F., Xie, Y.J.: The derivative-free double Newton step methods for solving system of nonlinear equations. Mediterr. J. Math. **13**, 2253–2270 (2016)
12. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**, 141–148 (1988)
13. Cruz, W., Raydan, M.: Nonmonotone spectral methods for large-scale nonlinear systems. Optim. Methods Softw. **18**, 583–599 (2003)
14. Cruz, W., Martínez, J.M., Raydan, M.: Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. Math. Comput. **75**, 1429–1448 (2006)
15. Zhang, L., Zhou, W.J.: Spectral gradient projection method for solving nonlinear monotone equations. J. Comput. Appl. Math. **196**, 478–484 (2006)
16. Yu, Z.S., Lin, J., Sun, J., X, Y.H., Liu, L.Y., Li, Z.H.: Spectral gradient projection method for monotone nonlinear equations with convex constraints. Appl. Numer. Math. **59**, 2416–2423 (2009)
17. Yu, G.H., Niu, S.Z., Ma, J.H.: Multivariate spectral gradient projection method for nonlinear monotone equations with convex constraints. J. Ind. Manag. Optim. **9**, 117–129 (2013)
18. Dai, Y.H., Yuan, Y.X.: Nonlinear Conjugate Gradient Methods (in Chinese). Shanghai Scientific and Technical Publishers, Shanghai (2000)
19. Cheng, W.Y.: A PRP type method for systems of monotone equations. Math. Comput. Model. **50**, 15–20 (2009)

20. Li, Q., Li, D.H.: A class of derivative-free methods for large-scale nonlinear monotone equations. IMA J. Numer. Anal. **31**, 1625–1635 (2011)
21. Xiao, Y.H., Zhu, H.: A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing. J. Math. Anal. Appl. **405**, 310–319 (2013)
22. Hager, W.W., Zhang, H.C.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**, 170–192 (2005)
23. Liu, J.K., Li, S.G.: Multivariate spectral DY-type projection method for convex constrained nonlinear monotone equations. J. Ind. Manag. Optim. (2016). doi:10.3934/jimo.2016017
24. Birgin, E., Martínez, J.M.: A spectral conjugate gradient method for unconstrained optimization. Appl. Math. Optim. **43**, 117–128 (2001)
25. Andrei, N.: Scaled conjugate gradient algorithms for unconstrained optimization. Comput. Optim. Appl. **38**, 401–416 (2007)
26. Shanno, D.F.: On the convergence of a new conjugate gradient algorithm. SIAM J. Numer. Anal. **15**, 1247–1257 (1978)
27. Sun, W.Y., Yuan, Y.X.: Optimization Theory and Methods : Nonlinear Programming. Springer Optimization and Its Applications, vol. 1. Springer, New York (2006)
28. Dolan, E.D., More, J.J.: Benchmarking optimization software with performance profiles. Math. Program. Ser. A **91**, 201–213 (2002)