A UNIFIED DERIVATIVE-FREE PROJECTION METHOD MODEL FOR LARGE-SCALE NONLINEAR EQUATIONS WITH CONVEX CONSTRAINTS

YIGUI OU* AND WENJIE XU

Department of Mathematics
Hainan University, Haikou 570228, China

(Communicated by Jan Joachim Ruckmann)

Abstract. Motivated by recent derivative-free projection methods proposed in the literature for solving nonlinear constrained equations, in this paper we propose a unified derivative-free projection method model for large-scale nonlinear equations with convex constraints. Under mild conditions, the global convergence and convergence rate of the proposed method are established. In order to verify the feasibility and effectiveness of the model, a practical algorithm is devised and the corresponding numerical experiments are reported, which show that the proposed practical method is efficient and can be applied to solve large-scale nonsmooth equations. Moreover, the proposed practical algorithm is also extended to solve the obstacle problem.

1. Introduction. Let us consider the following problem, which is to find a point \( x \in \mathbb{R}^n \) such that

\[
F(x) = 0, \quad x \in X,
\]

where the mapping \( F : \mathbb{R}^n \to \mathbb{R}^n \) is continuous, and \( X \subset \mathbb{R}^n \) is a nonempty closed convex set. Throughout the paper, \( X^* \) denotes the solution set of (1), \( \| \cdot \| \) denotes the Euclidean norm on \( \mathbb{R}^n \), \( F_k \) denotes \( F(x_k) \), and \( \text{dist}(x_k, X) \) denotes the distance from the iterates \( x_k \) to \( X \).

Nonlinear equations originate in many applications, see [15, 14, 23, 26, 8, 29] for instance. These various applications have provoked many scholars to study numerical methods for the problem (1). Up to now, many iterative methods have been proposed and analyzed, which may be mainly divided into two categories. One is the derivative methods that need to calculate the Jacobian matrix of \( F \) or its approximation at each iteration, such as Newton-type methods, Levenberg-Marquardt method, trust region method and projection method, see [20, 10, 9, 24] for instance. However, this kind of method is not suitable for solving large-scale nonlinear equations, due to the fact that it involves matrix computation and storage. The other is derivative-free methods using the projection strategy (usually called as derivative-free projection methods). Since the derivative-free projection methods do...
not involve any matrix computation and storage, they are particularly effective for solving large-scale nonlinear system of equations. Therefore, this kind of method has attracted much more attention and many numerical methods have been proposed, see for example [13, 1, 12, 7, 17, 25, 27, 11, 3, 28] and references therein.

In general, the main difference in the existing derivative-free projection methods for solving problem (1) lies in the construction of direction \( d_k \) and the choice of stepsize \( \alpha_k \). Regarding the construction of the direction \( d_k \), it is always required to satisfy the following condition, i.e., there exists a positive constant \( c_1 \) such that

\[
F_k^T d_k \leq -c_1 \|F_k\|^2, \quad \forall k,
\]

see [13, 1, 12, 7, 17, 25, 27, 11, 3, 28] for instance. Obviously, If \( F \) is the gradient of a real-valued function \( f \) defined in \( \mathbb{R}^n \), the condition (2) implies that \( d_k \) is a sufficiently descent direction of \( f \) at \( x_k \), which plays an important role in analyzing the convergence property of gradient-related methods.

As for the choice of \( \alpha_k \), two well known line search procedures have been proposed in projection-based methods for solving problem (1) [2]. More precisely, given a current iterate \( x_k \) and a search direction \( d_k \) at the \( k \)-th iteration, \( \alpha_k \) is determined, respectively as follows:

I. Compute a stepsize \( \alpha_k = \max\{\tau \rho^i : i = 0, 1, 2, \ldots \} \) such that

\[
-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \|d_k\|^2,
\]

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

II. Compute a stepsize \( \alpha_k = \max\{\tau \rho^i : i = 0, 1, 2, \ldots \} \) such that

\[
-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \|F(x_k + \alpha_k d_k)\| \|d_k\|^2.
\]

Numerical experiments indicate that the above-mentioned line search schemes I and II have different performance during solving problem (1), see [2] for details. In order to maintain their advantages and overcome their disadvantages, a new adaptive line search scheme is introduced in [2], i.e., it chooses a stepsize \( \alpha_k = \max\{\tau \rho^i : i = 0, 1, 2, \ldots \} \) such that

\[
-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \gamma_k \|d_k\|^2,
\]

where \( \gamma_k = \frac{\|F(x_k + \alpha_k d_k)\|}{\|F(x_k)\|} \). Preliminary numerical results and related comparisons in [2] show that the new scheme is very effective and can improve the efficiency of projection-based methods. Subsequently, Ou and Li [16] also gave an adaptive scheme, i.e., it computes a stepsize \( \alpha_k = \max\{\tau \rho^i : i = 0, 1, 2, \ldots \} \) such that

\[
-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \zeta_k \|d_k\|^2,
\]

where

\[
\zeta_k = \lambda_k + (1 - \lambda_k) \|F(x_k + \alpha_k d_k)\|
\]

with the weight \( \lambda_k \in [\lambda_{\text{min}}, \lambda_{\text{max}}] \subseteq (0, 1] \).

It should be noted that almost all of the above-mentioned derivative-free projection methods for problem (1) require that the mapping \( F \) is monotone and Lipschitz continuous, see [13, 1, 12, 7, 17, 25, 27, 11, 3, 28, 2, 16] for instance. However, the requirement of monotonicity or Lipschitz continuity seems too stringent for the purpose of establishing global convergence property. Recently, Zheng [30] proposed a new projection algorithm for a system of nonlinear equations with convex constraints, which possesses a nice global convergence property without the Lipschitz continuity and monotonicity assumptions. Unfortunately, Zheng’s method needs to
solve linear equations exactly at each iteration, which may lead to expensive computation and thus is generally not suitable for solving large-scale nonlinear equations. Therefore, it is necessary for us to develop a different derivative-free projection method for solving problem (1), which not only possesses good global convergence property without the requirement of monotonicity and Lipschitz continuity, but also has excellent numerical performance in solving large-scale nonlinear system of equations.

As is well known, the analysis on the convergence rate is also important for numerical optimization algorithms. However, most of the existing derivative-free projection methods for problem (1) only discuss their global convergence, while the convergence rate is not analyzed, see [1, 17, 25, 27, 11, 3, 28, 2] for example. In addition, even for those algorithms in which the convergence rate has been analyzed, the obtained result is only that the distance sequence \( \{\text{dist}(x_k, X^*)\} \) converges to zero with a Q-linear rate, rather than the sequence \( \{x_k\} \) itself converges to a solution of problem (1) with a Q-linear rate, see [13, 12, 7] for example. So far, only a few literatures have discussed the convergence rate of iterative sequence \( \{x_k\} \) itself, see [16] for example. These facts motivate us to further explore derivative-free projection methods for problem (1), especially those with fast convergence and good numerical performance, it is another motivation behind the present study.

Based on the above discussions, in this paper we further investigate the numerical method for problem (1) and propose a unified derivative-free projection method model. Under much weaker assumptions than monotonicity and Lipschitz continuity, the proposed method model is proved to be global convergence. Moreover, we also discuss the convergence rate of the method model under some suitable conditions.

The rest of the paper is organized as follows. In Section 2, we outline a unified method model for solving problem (1). Section 3 is devoted to analyze the global convergence of the proposed model under some mild conditions. In Section 4, we analyze the convergence rate of the proposed model under some assumptions. In Section 5, a practical derivative-free projection algorithm for solving large-scale nonlinear equations (1) is devised, then numerical experiments and related comparisons are reported to show the efficiency of the practical algorithm. Application of the proposed practical method in obstacle and free boundary problems is introduced in Section 6. Some conclusions are summarized in the final section.

2. Description of method model. In this section, we propose a derivative-free projection method model for solving the problem (1), and then give some remarks and results on this model.

To describe the method model, we first introduce the definition of projection operator \( P_X[\cdot] \) (see [24] for example), which is defined as a mapping from \( \mathbb{R}^n \) to a nonempty closed convex subset \( X \):

\[
P_X[x] = \arg \min_{y \in X} \{\|y - x\|\}, \quad \forall x \in \mathbb{R}^n.
\]

A famous property of this operator is that it is nonexpansive, namely,

\[
\|P_X[x] - P_X[y]\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^n.
\]

Now, the method model is outlined as follows.

Algorithm Model (AM)

**Step 0.** Given an initial point \( x_0 \in X \), \( \sigma \in (0, 1) \), \( \rho \in (0, 1) \), and \( \tau > 0 \). Set \( k := 0 \).
Step 1. Stop if $\|F_k\| = 0$.

Step 2. Construct a direction $d_k$ satisfying the condition (2).

Step 3. Compute the trial point $z_k = x_k + \alpha_k d_k$, where $\alpha_k$ is chosen to satisfy the line search scheme (5).

Step 4. If $\|F(z_k)\| = 0$, stop. Otherwise, compute the next iterate $x_{k+1}$ using

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

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

Step 5. Set $k := k + 1$, and go to Step 1.

Remark 2.1. It is clear that this is a conceptual algorithm model, since we have omitted some details needed to specify a complete procedure, for example, an iterative format for determining $d_k$ in Step 2.

Remark 2.2. Note that in Step 3, the stepsize $\alpha_k$ can be chosen to satisfy one of the above-mentioned four line search schemes. Here, the reason why we choose the line search scheme (5) to obtain $\alpha_k$ is that it can be viewed as an improved version of the line schemes I and II, and that it can avoid calculating the weight $\lambda_k$ in the scheme (6).

The following lemma shows that Algorithm AM is well defined when the solution of (1) is not found.

Lemma 2.1. Algorithm AM is well defined, i.e., there exists a nonnegative integer $i_k$ satisfying the line search scheme (5) for any $k$.

Proof. Suppose on the contrary that there is an integer $k_0 \geq 0$ such that (5) is not satisfied for any nonnegative integer $i$, i.e.,

$$-F(x_{k_0} + \tau \rho^i d_{k_0})^T d_{k_0} < \sigma \tau \rho^i \gamma_{k_0} \|d_{k_0}\|^2,$$

where $\gamma_{k_0} = \frac{\|F(x_{k_0} + \tau \rho^i d_{k_0})\|}{1 + \|F(x_{k_0} + \rho^i d_{k_0})\|}$. Since $F$ is continuous and $\rho \in (0, 1)$, passing onto the limit as $k \to +\infty$ in (11), we obtain

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

On the other hand, it follows from (2) and $F_k \not= 0$ for any $k$ that

$$-F(x_k)^T d_k \geq c_1 \|F_k\|^2 > 0, \; \forall k.$$

This contradicts (12). This proof is then completed. \hfill \Box

3. Global convergence analysis. This section is devoted to studying the global convergence of Algorithm AM. For this purpose, we first make the following assumptions.

A1. There exists a point $x^* \in X^*$ such that

$$F(x)^T (x - x^*) \geq 0, \; \forall x \in \mathbb{R}^n.$$

A2. The sequence $\{d_k\}$ is bounded if whenever the sequence $\{F_k\}$ tends to a nonzero vector.

Remark 3.1. Assumption A1 was first introduced to established the global convergence of the projection algorithms for variational inequality problem by Solodov
and Svaiter [21]. Subsequently, it was used by Zheng [30] to prove the global convergence of the projection method for constrained equations. Note that Assumption A1 holds if the mapping $F$ is monotone, i.e.,

$$[F(x) - F(y)]^T(x - y) \geq 0, \ \forall x, y \in \mathbb{R}^n,$$

(14)

or pseudomonotone, i.e., for all $x, y \in \mathbb{R}^n$,

$$F(y)^T(x - y) \geq 0 \Rightarrow F(x)^T(x - y) \geq 0, \ \forall x, y \in \mathbb{R}^n,$$

(15)

but not vice versa (see [21] for instance). Hence, Assumption A1 is weaker than monotonicity of $F$ used in the existing derivative-free projection methods for the problem (1).

**Remark 3.2.** It is clear that Assumption A2 is satisfied if the sequence $\{d_k\}$ is bounded. Hence, Assumption A2 can be replaced by the following assumption:

**A2'.** The sequence $\{d_k\}$ is bounded.

It should be pointed out that for the existing derivative-free projection methods for solving the problem (1), Assumption A2 or A2' can be verified to be always satisfied, see [13, 1, 12, 7, 17, 25, 27, 11, 3, 28, 2, 16] for example.

In what follows, we always assume that $F_k \neq 0$ and $F(z_k) \neq 0$ for some $k$, i.e., Algorithm AM generates two infinite sequences $\{x_k\}$ and $\{z_k\}$, otherwise, a solution to (1) is found.

**Lemma 3.1.** Let $x^* \in X^*$. If Assumption A1 holds, then we have

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \frac{\sigma^2\|x_k - z_k\|^4}{(1 + \|F(z_k)\|)^2}, \ \forall k. $$

(16)

Furthermore, the sequence $\{x_k\}$ is bounded.

**Proof.** From Step 3 of algorithm AM, it follows that

$$F(z_k)^T(x_k - z_k) = -\alpha_k F(z_k)^T d_k \geq \sigma \gamma_k \alpha^2_k \|d_k\|^2 = \sigma \gamma_k \|x_k - z_k\|^2,$$

(17)

where $z_k = x_k + \alpha_k d_k$, and $\gamma_k = \frac{\|F(z_k)\|}{1 + \|F(z_k)\|}$. By A1 and $x^* \in X^*$, we get

$$F(z_k)^T(x_k - z_k) = F(z_k)^T(x_k - z_k) + F(z_k)^T(z_k - x^*) \geq F(z_k)^T(z_k - z_k).$$

(18)

Combining (17) and (18) gives

$$\langle F(z_k), x_k - x^* \rangle \geq \sigma \gamma_k \|x_k - z_k\|^2.$$  

(19)

Then, by (9), (10), (17) and (18), we deduce that

$$\|x_{k+1} - x^*\|^2 = \|P_X [x_k - \xi_k F(z_k)] - P_X [x^*]\|^2$$

$$\leq \|x_k - x^*\|^2 - 2\xi_k F(z_k)^T(x_k - x^*) + \xi_k^2 \|F(z_k)\|^2$$

$$\leq \|x_k - x^*\|^2 - 2\xi_k F(z_k)^T(x_k - z_k) + \xi_k^2 \|F(z_k)\|^2$$

$$= \|x_k - x^*\|^2 - \left(\frac{\|F(z_k)\|^2}{1 + \|F(z_k)\|}\right)^2$$

$$\leq \|x_k - x^*\|^2 - \frac{\sigma^2\|x_k - z_k\|^4}{(1 + \|F(z_k)\|)^2}, \ \forall k,$$

(20)

which implies that the assertion (16) is true.

Furthermore, it follows from (16) that

$$\|x_k - x^*\| \leq \|x_{k-1} - x^*\| \leq \cdots \leq \|x_0 - x^*\|, \ \forall k.$$

This further implies that $\{x_k\}$ is bounded. This completes the proof. \qed
Lemma 3.2. Suppose that Assumptions A1 and A2 hold. If there exists \( \varepsilon > 0 \) such that
\[
\|F_k\| \geq \varepsilon, \tag{21}
\]
then the sequence \( \{z_k\} \) is bounded. Furthermore, we have
\[
\lim_{k \to +\infty} \|x_k - z_k\| = \lim_{k \to +\infty} \alpha_k \|d_k\| = 0. \tag{22}
\]
Proof. By A2 and (21), it follows that the sequence \( \{d_k\} \) is bounded. This fact together with the boundedness of \( \{x_k\} \) (see Lemma 3.1) and \( \alpha_k \leq \tau \) for all \( k \) implies that the sequence \( \{z_k\} | z_k = x_k + \alpha_k d_k, k = 0, 1, \cdots \} \) is also bounded.

Let \( x^* \in X^* \). Then, by the boundedness of \( \{z_k\} \) and the continuity of \( F \), we deduce that \( \{F(z_k)\} \) is bounded, i.e., there exists a constant \( c_2 > 0 \) such that
\[
\|F(z_k)\| \leq c_2, \quad \forall k, \tag{23}
\]
which, together with (16), implies that
\[
\sigma^2 \frac{1}{(1 + c_2)^2} \sum_{k=0}^{+\infty} \|x_k - z_k\|^4 \leq \sigma^2 \sum_{k=0}^{+\infty} \frac{\|x_k - z_k\|^4}{(1 + \|F(z_k)\|)^2} \leq \|x_0 - x^*\|^2 < +\infty. \tag{24}
\]
This further implies that the assertion (22) is true. This completes the proof. \( \square \)

Remark 3.3. In Lemma 3.2, if the condition (21) is removed and Assumption A2 is replaced by Assumption A2', the conclusion still holds.

Using these lemmas mentioned above, we obtain the following global convergence property of Algorithm AM.

Theorem 3.3. Suppose that Assumptions A1 and A2 hold. Then we have
\[
\lim_{k \to +\infty} \inf \|F_k\| = 0. \tag{25}
\]
Furthermore, the whole sequence \( \{x_k\} \) converges to a solution \( \bar{x} \) of problem (1).

Proof. The proof is similar to that of Theorem 3.1 in [28]. \( \square \)

Remark 3.4. From Theorem 3.3, we see that the global convergence of Algorithm AM is established without the assumptions of monotonicity and Lipschitz continuity. However, for the existing derivative-free projection methods for solving large-scale nonlinear equations (1), these assumptions are essential to prove the global convergence property (see [13, 1, 12, 7, 17, 25, 27, 11, 3, 28, 2, 16] for instance). Therefore, the assumptions in this paper are considerably weaker.

4. Analysis of convergence rate. In this section, we analyze the convergence rate of Algorithm AM when it is applied to problem (1). For this purpose, we need the following assumptions.

A3. The mapping \( F \) is Lipschitz continuous on the nonempty closed convex set \( X \), i.e., there exists a constant \( L > 0 \) such that
\[
\|F(x) - F(y)\| \leq L \|x - y\|, \quad \forall x, y \in X, \tag{26}
\]

A4. There exists a constant \( c_3 > 0 \) such that
\[
\|d_k\| \leq c_3 \|F_k\|, \quad \forall k. \tag{27}
\]
A5. $\|F(x)\|^2$ provides a local error bound on some neighborhood $\mathcal{N}(x^*, \delta)$ of $x^* \in X^*$, i.e., there exist constant $c_4 > 0$ and $\delta \in (0, 1)$ such that
\[
\|F(x)\|^2 \geq c_4 \text{dist}(x, X^*), \quad \forall x \in \mathcal{N}(x^*, \delta) \cap X,
\]
where $\text{dist}(x, X^*) = \inf_{y \in X^*} \|x - y\|$, and $\mathcal{N}(x^*, \delta) = \{x \in \mathbb{R}^n \|x - x^*\| \leq \delta\}$.

**Remark 4.1.** Assumption A3 is standard. For many of the existing derivative-free projection methods for solving the problem (1), Assumption A4 can be verified to be satisfied, see [13, 1, 7, 17, 25, 11, 3, 28, 16] for details. Of course, whether the direction $d_k$ satisfies the condition (27) or not depends entirely on its construction.

It is noted that for the existing derivative-free projection methods for problem (1), the sequence $\{x_k\}$ can be proved to be always bounded when $F$ is continuous and monotone, see [13, 1, 12, 7, 17, 25, 27, 11, 3, 28, 16] for instance. This fact together with A4 and the continuity of $F$ further implies that the sequence $\{d_k\}$ is also bounded (i.e., Assumption A2' is satisfied), so is the sequence $\{z_k\}$.

**Remark 4.2.** It follows from (16) that $x_{k+1} \in \mathcal{N}(x^*, \delta)$ if $x_k \in \mathcal{N}(x^*, \delta)$ for all $k$.

To analyze the convergence rate of Algorithm AM, we need the following lemma. Since its proof is similar to that of Lemma 2 in [2], we therefore omit it here.

**Lemma 4.1.** Suppose that Assumption A3 holds. Then there exists a constant $\gamma > 0$ such that the stepsize $\alpha_k$ defined in Step 3 of Algorithm AM satisfies
\[
\alpha_k \geq \min\{\tau, \gamma \frac{\|F_k\|^2}{\|d_k\|^2}\}, \quad \forall k,
\]
where $\gamma = \frac{\tau}{L + \sigma}$.

**Theorem 4.2.** Let $\{x_k\}$ be an infinite sequence generated by AM. Suppose that Assumptions A1, A3, A4 and A5 hold. Then the distance sequence $\{\text{dist}(x_k, X^*)\}$ Q-linearly converges to 0.

**Proof.** Let $\bar{x}_k \in X^*$ be such that
\[
\|x_k - \bar{x}_k\| = \text{dist}(x_k, X^*). \quad (30)
\]
Then, it follows from (16) and (30) that
\[
\text{dist}^2(x_{k+1}, X^*) \leq \|x_{k+1} - \bar{x}_k\|^2 \leq \text{dist}^2(x_k, X^*) - \frac{\sigma^2 \|x_k - z_k\|^4}{(1 + \|F(z_k)\|^2)^2}, \quad \forall k. \quad (31)
\]
From A4 and (29), it follows that there exists a positive constant $c_5 = \min\{\tau, \frac{\gamma}{L + \sigma}\}$ such that
\[
\alpha_k \geq \min\{\tau, \gamma \frac{\|F_k\|^2}{\|d_k\|^2}\} \geq c_5, \quad \forall k. \quad (32)
\]
Moreover, by (2) and the Cauchy-Schwarz inequality, we get
\[
\|d_k\| \geq c_1 \|F_k\|. \quad (33)
\]
Combining this inequality with (32) and (28) yields
\[
\|x_k - z_k\|^2 = c_4 \|d_k\|^2 \geq c_4 c_1^2 \|F_k\|^2 \geq (c_1 c_5)^2 c_4 \text{dist}(x_k, X^*), \quad \forall k. \quad (34)
\]
Note that $\{z_k\}$ is bounded by Remark 4.1, then it follows from the continuity of $F$ that $\{F(z_k)\}$ is also bounded, i.e., there exists a positive constant $c_6$ ($c_6 \geq \sigma (c_1 c_5)^2 c_4$) such that
\[
\|F(z_k)\| \leq c_6, \quad \forall k. \quad (35)
\]
This inequality (35) together with (31) and (34) implies that
\[
\text{dist}^2(x_{k+1}, X^*) \leq \text{dist}^2(x_k, X^*) - \frac{\sigma^2(c_1c_5)^4\sigma_1^2}{(1+c_6)^2}\text{dist}^2(x_k, X^*)
\]
where $c_7 = 1 - \frac{\sigma^2(c_1c_5)^4\sigma_1^2}{(1+c_6)^2} < (0, 1)$. Then, the desired conclusion directly follows from (36). The proof is completed. \(\square\)

Now, let us discuss the convergence rate of the sequence \(\{x_k\}\) generated by Algorithm AM. To this end, we need the following lemma, which is from Lemma 6 in Chapter 2 of [19].

**Lemma 4.3.** Let $u_k > 0$ and let
\[
u_{k+1} \leq u_k - a_k u_k^{1+p}, \quad a_k \geq 0, \quad p > 0.
\]
Then
\[
u_k \leq u_0 \left(1 + pu_0^{k-1} \sum_{i=0}^{k-1} a_i\right)^{-\frac{1}{p}}.
\]
In particular, if $a_k \equiv a$, $p = 1$, then
\[
u_k \leq \frac{u_0}{1 + au_0^k}.
\]

**Theorem 4.4.** Let \(\{x_k\}\) be an infinite sequence generated by AM. Suppose that Assumptions A3 and A4 hold. If the mapping $F$ is strongly monotone with modulus $\mu > 0$, i.e.,
\[\langle F(x) - F(y), x - y \rangle \geq \mu \|x - y\|^2, \quad \forall x, y \in X,\]
then the sequence \(\{x_k\}\) converges to a solution $x^* \in X$ at a $R$-sublinear rate, i.e.,
\[\lim_{k \to +\infty} \sup \|x_k - x^*\| \leq 1.
\]

**Proof.** By the strong monotonicity of $F$ and the Cauchy-Schwarz inequality, we get
\[\|F_k\| = \|F_k - F(\bar{x})\| \geq \mu \|x_k - \bar{x}\|,
\]
which, together (32) and (33), implies that
\[\|x_k - z_k\| = \alpha_k \|d_k\| \geq c_5 c_1 \|F_k\| \geq c_5 c_1 \mu \|x_k - \bar{x}\|.
\]
Combining this inequality with (35) and (16) yields
\[
\|x_{k+1} - \bar{x}\|^2 \leq \|x_k - \bar{x}\|^2 - \frac{\sigma^2 \|x_k - z_k\|^4}{(1 + \|F(x_k)\|^2)} \leq \|x_k - \bar{x}\|^2 - \frac{\sigma^2 (c_5 c_1 \mu)^4}{(1+c_6)^2} \|x_k - \bar{x}\|^4
\]
where $c_8 := \frac{\sigma^2 (c_5 c_1 \mu)^4}{(1+c_6)^62}$. Then, using Lemma 4.3 with $u_k = \|x_k - \bar{x}\|^2$ and $a_k \equiv c_8$, we obtain
\[
\|x_k - \bar{x}\|^2 \leq \frac{\|x_0 - \bar{x}\|^2}{1 + c_8 \|x_0 - \bar{x}\|^2 k},
\]
i.e.,
\[
\|x_k - \bar{x}\| \leq \frac{1}{\sqrt{c_8 k + \|x_0 - \bar{x}\|^2 k}} \leq \frac{1}{\sqrt{c_8 k}},
\]
which shows the truth of (40). The proof is completed. \(\square\)
5. **A practical algorithm.** In this section, a practical algorithm for solving the problem (1.1) is proposed to verify the feasibility and effectiveness of the model AM, based on a two-parameter scaled memoryless BFGS methods [4] (abbreviated as TPSMBFGS method).

5.1. **Algorithm.** Let us first simply recall the TPSMBFGS method for unconstrained optimization. Consider the following problem:

\[
\min_{x \in \mathbb{R}^n} f(x),
\]

where \( f : \mathbb{R}^n \to \mathbb{R} \) is a continuously differentiable function. The TPSMBFGS method has been successfully used to solve the problem (44), which consists of iteration of the form

\[
x_{k+1} = x_k + a_k d_k,
\]

where \( a_k > 0 \) is a stepsize obtained by a suitable line search scheme (see [22] for details), and \( d_k \) is the search direction defined by

\[
\begin{cases}
  d_k &= -H_k g_k, \quad (k \geq 1), \\
  d_0 &= -g_0,
\end{cases}
\]

where \( g_k \) is the gradient of \( f \) at \( x_k \), and \( H_k \) is updated by the following two-parameter scaled memoryless BFGS updating formula

\[
H_{k+1} = \theta_k I - \theta_k \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} + \left( 1 + \gamma_k \frac{\|y_k\|^2}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k}
\]

(47)

with

\[
s_k = x_{k+1} - x_k, \quad y_k = g_{k+1} - g_k,
\]

\[
\theta_k = \frac{1}{\frac{1}{2} - \eta \|y_k\|^2}, \quad \gamma_k = \eta \theta_k, \quad 1 \leq \eta < 2.
\]

As we know, \( \{H_k\} \) is a symmetric positive definite matrix sequence when \( s_k^T y_k > 0 \) for all \( k \) (see [4] for details).

To extend the idea of TPSMBFGS method to solving nonlinear equations (1), we define the search direction \( d_k \) as follows:

\[
\begin{cases}
  d_k &= -\overline{H}_k F_k, \quad (k \geq 1), \\
  d_0 &= -F_0,
\end{cases}
\]

(49)

where \( \overline{H}_k \) is updated by

\[
\overline{H}_{k+1} = \overline{\theta}_k I - \overline{\theta}_k \frac{s_k z_k^T + z_k s_k^T}{s_k^T z_k} + \left( 1 + \overline{\gamma}_k \frac{\|z_k\|^2}{s_k^T z_k} \right) \frac{s_k s_k^T}{s_k^T z_k}
\]

(50)

with

\[
s_k = x_{k+1} - x_k, \quad z_k = F_{k+1} - F_k + r s_k, \quad r > 0,
\]

\[
\overline{\theta}_k = \frac{1}{\frac{1}{2} - \eta \|z_k\|^2}, \quad \overline{\gamma}_k = \eta \overline{\theta}_k, \quad 1 \leq \eta < 2.
\]

**Remark 5.1.** If the mapping \( F \) is monotone, then

\[
z_k^T s_k = (F_{k+1} - F_k)^T (x_{k+1} - x_k) + r \|s_k\|^2 \geq r \|s_k\|^2 > 0, \quad \forall k,
\]

(52)

which ensure that \( \overline{H}_k \) is a symmetric positive definite matrix for all \( k \). This fact together with the Sherman-Morrison formula (see [22] for details) implies that

\[
\overline{B}_{k+1} := \overline{H}^{-1}_{k+1} = \frac{1}{\theta_k} - \frac{s_k s_k^T}{\theta_k \|s_k\|^2} + \frac{z_k z_k^T}{s_k^T z_k + (\overline{\gamma}_k - \theta_k) \|z_k\|^2}
\]

(53)

is also a symmetric positive definite matrix for all \( k \).
Now, we describe the detailed algorithm for solving problem (1) as follows.

**Algorithm 5.1**

**Step 0.** Given \( r > 0 \), \( \eta \in [1, 2) \), \( \sigma \in (0, 1) \), \( \rho \in (0, 1) \), \( \tau > 0 \), \( \epsilon \geq 0 \), and \( x_0 \in X \). Set \( k := 0 \).

**Step 1.** If \( \|F_k\| \leq \epsilon \), stop.

**Step 2.** Construct a direction \( d_k \) using (49)-(51).

**Step 3.** Compute the trial point \( z_k = x_k + \alpha_k d_k \), where \( \alpha_k \) is chosen to satisfy the line search scheme (5).

**Step 4.** If \( \|F(z_k)\| \leq \epsilon \), stop. Otherwise, compute the next iterate \( x_{k+1} \) using

\[
x_{k+1} = X_k \left[ x_k - \xi_k F(z_k) \right],
\]

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

**Step 5.** Set \( k := k + 1 \), and go to Step 1.

### 5.2. Convergence property

To analyze the convergence properties of Algorithm 5.1, we need the following results.

**Lemma 5.1.** Suppose that Assumption A3 holds. If the mapping \( F \) is monotone, then there exist two positive constants \( c_9 \) and \( c_{10} \) such that

\[
F_k^T d_k \leq -c_9 \|F_k\|^2, \quad \forall k,
\]

and

\[
\|d_k\| \leq c_{10} \|F_k\|, \quad \forall k.
\]

**Proof.** From (50) and (53), it follows that the trace of \( \bar{H}_{k+1} \) and \( \bar{H}_{k+1}^{-1} \) can be computed respectively by

\[
tr(\bar{H}_{k+1}) = (n - 2) \bar{\theta}_k + \left( 1 + \eta \bar{\theta}_k \right) \|s_k\|^2 / s_k^T z_k,
\]

and

\[
tr(\bar{H}_{k+1}^{-1}) = \frac{n - 1}{\bar{\theta}_k} + \frac{\|z_k\|^2}{s_k^T z_k} / (\eta - 1) \bar{\theta}_k / \|z_k\|^2.
\]

Then, it follows from (52) and A3 that

\[
\|s_k^T z_k\| / \|z_k\|^2 \leq \frac{\|s_k\|^2}{s_k^T z_k} \leq \frac{1}{r},
\]

and

\[
\|z_k\|^2 / s_k^T z_k \leq \frac{(\|F_{k+1} - F_k\| + r \|s_k\|)^2}{r \|s_k\|^2} \leq \frac{(L + r)^2 \|s_k\|^2}{r \|s_k\|^2} = \frac{(L + r)^2}{r},
\]

and thus

\[
\bar{\theta}_k = \frac{1}{\eta \|z_k\|^2} \leq \frac{1}{r(2 - \eta)},
\]

\[
\frac{1}{\bar{\theta}_k} = \frac{2 - \eta \|z_k\|^2}{s_k^T z_k} \leq \frac{(2 - \eta)(L + r)^2}{r}.
\]

Combining the above inequalities with (56) and (57) gives

\[
tr(\bar{H}_{k+1}) \leq \frac{(n - 2)}{(2 - \eta) r} + \left( 1 + \frac{r(2 - \eta)}{\eta(2 - \eta)} \right) \left( L + r \right)^2 / r.
\]

and

\[
tr(\bar{H}_{k+1}^{-1}) \leq \frac{(n - 1)}{\bar{\theta}_k} + \frac{1}{(\eta - 1) \bar{\theta}_k} \leq \left( n - 1 + \frac{1}{\eta - 1} \right) \frac{(2 - \eta)(L + r)^2}{r}.
\]
Now, we prove the inequalities (54) and (55). For \( k = 0, d_0 = -F_0 \), and thus
\[
F_0^T d_0 = -\|F_0\|^2, \quad \|d_0\| = \|F_0\|.
\] (64)
For \( k \geq 0 \), it follows from (62)-(63) and (49) that
\[
-F_{k+1}^T d_{k+1} = F_{k+1}^T \tilde{H}_{k+1} F_{k+1} \geq \frac{\|F_{k+1}\|^2}{\|\tilde{H}_{k+1}\|} \geq \frac{\|F_{k+1}\|^2}{\text{tr}(\tilde{H}_{k+1})} \geq \frac{\|F_{k+1}\|^2}{c_{11}},
\] (65)
where \( c_{11} = (n - 1 + \frac{1}{(q-1)}) \frac{(2-q)(L+r)^2}{r} \), and \( c_{12} = (n - 1 + \frac{1}{(q-1)}) \frac{(2-q)(L+r)^2}{r} \).

Taking \( \delta_9 = \min\{1, \frac{1}{3}\} \) and \( \delta_{10} = \max\{1, c_{12}\} \), then the desired conclusions follow directly from (64), (65) and (66). The proof is completed. \( \Box \)

Now, we give the convergence properties of Algorithm 5.1. Since they can be proven by using the same argument as that in Theorems 4.2 and 4.4, we therefore omit the proof.

**Theorem 5.2.** Let \( \{x_k\} \) be an infinite sequence generated by Algorithm 5.1. Suppose that Assumptions A3 and A5 hold. If the mapping \( F \) is monotone, then the distance sequence \( \{\text{dist}(x_k, X^*)\} \) Q-linearly converges to 0.

**Theorem 5.3.** Let \( \{x_k\} \) be an infinite sequence generated by Algorithm 5.1. Suppose that Assumption A3 holds. If the mapping \( F \) is strongly monotone with modulus \( \mu > 0 \), then the sequence \( \{x_k\} \) converges to a solution \( \hat{x} \in X^* \) at a sublinear rate.

At the end of this section, it should be mentioned that under common conditions as noted in Theorem 5.3, the sequence \( \{x_k\} \) generated by the algorithm [16] is R-linearly convergent to \( x^* \). Obviously, this conclusion is different from that of Theorem 4.4 in this paper. The reason why there are such different results lies in the different ways of choosing the stepsize \( \alpha_k \). Therefore, in the future research, we should discuss how to devise a more effective line search scheme to construct a derivative-free projection method for solving problem (1).

### 5.3. Numerical experiments.

In this subsection, we report some numerical results of Algorithm 5.1 on a set of test problems with four different initial points:
\[
x_{01} = (5,5,...,5)^T; \quad x_{02} = (1, \frac{1}{2}, ..., \frac{1}{n}); \quad x_{03} = (\frac{1}{n}, \frac{2}{n}, ..., 1)^T; \quad x_{04} = \text{rand}(n,1).
\]

Meanwhile, we compare it with some related algorithms including Ou-Li’s algorithm (OLA) [16] and Xiao-Zhu’s algorithm (XZA) [27]. All codes were written in Matlab 7.0 and run on a PC computer with CPU 2.60 GHZ and 2.00 GB memory.

The test problems are listed as follows (see [27, 2, 16] for example), where the mapping \( F \) is defined as \( F(x) = (F_1(x), F_2(x), \cdots, F_n(x))^T \).

**Problem 1.** Strictly convex function I:
\[
F_i(x) = \exp(x_i) - 1, i = 1, 2, \cdots, n,
\]
and \( X = \{ x \in \mathbb{R}^n | x_i \geq 0, i = 1, 2, \cdots, n \} \).

**Problem 2.** Strictly convex function II:
\[
F_i(x) = \frac{i}{n} \exp(x_i) - 1, i = 1, 2, \cdots, n,
\]
and \( X = \{ x \in \mathbb{R}^n | x_i \geq 0, i = 1, 2, \cdots, n \} \).
Problem 3. Non-smooth function

\[ F_i(x) = x_i - \sin(|x_i|), \quad i = 1, 2, \ldots, n, \]
and \( X = \{ x \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i \leq n, x_i \geq 0, \quad i = 1, 2, \ldots, n \} \).

Problem 4. Tridiagonal exponential function:

\[
\begin{align*}
F_i(x) &= x_1 - \exp(\cos(\frac{x_1 + x_2}{n+1})), \\
F_i(x) &= x_i - \exp(\cos(\frac{x_{i-1} + x_i + x_{i+1}}{n+1})), \quad i = 2, \ldots, n - 1, \\
F_n(x) &= x_n - \exp(\cos(\frac{x_{n-1} + x_n}{n+1})),
\end{align*}
\]
and \( X = \{ x \in \mathbb{R}^n \mid x_i \geq 0, \quad i = 1, 2, \ldots, n \} \).

Problem 5. Exponential function:

\[
\begin{align*}
F_1(x) &= \exp(x_1) - 1, \\
F_i(x) &= \exp(x_i) + x_{i-1} - 1, \quad i = 2, 3, \ldots, n, \\
F_n(x) &= 2x_n + \sin(x_n) - 1,
\end{align*}
\]
and \( X = \{ x \in \mathbb{R}^n \mid x_i \geq 0, \quad i = 1, 2, \ldots, n \} \).

Problem 6. Function:

\[
\begin{align*}
F_1(x) &= 2x_1 + \sin(x_1) - 1, \\
F_i(x) &= 2x_{i-1} + 2x_i + \sin(x_i) - 1, \quad i = 2, 3, \ldots, n - 1, \\
F_n(x) &= 2x_n + \sin(x_n) - 1,
\end{align*}
\]
and \( X = \{ x \in \mathbb{R}^n \mid x_i \geq 0, \quad i = 1, 2, \ldots, n \} \).

Problem 7. Function

\[ F(x) = Ax + g(x), \]

where

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

\[
A = \begin{pmatrix}
2 & -1 & 0 & \cdots & 0 \\
-1 & 2 & -1 & \cdots & 0 \\
0 & -1 & 2 & -1 & \cdots \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
0 & 0 & \cdots & -1 & 2 \\
\end{pmatrix},
\]
and \( X = \{ x \in \mathbb{R}^n \mid 1 \leq x_i \leq 5, \quad i = 1, 2, \ldots, n \} \).

Problem 8. Function

\[ F_i(x) = 2x_i - \sin(|x_i|), \quad i = 1, 2, \ldots, n, \]
and \( X = \{ x \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i \leq n, x_i \geq 0, \quad i = 1, 2, \ldots, n \} \).

Problem 9. Function

\[
\begin{align*}
F_1(x) &= 2.5x_1 + x_2 - 1, \\
F_i(x) &= x_{i-1} + 2.5x_i + x_{i+1} - 1, \quad i = 2, 3, \ldots, n - 1, \\
F_n(x) &= x_{n-1} + 2.5x_n - 1,
\end{align*}
\]
and \( X = \{ x \in \mathbb{R}^n \mid x_i \geq 0, \quad i = 1, 2, \ldots, n \} \).

Problem 10. Function:

\[
\begin{align*}
F_1(x) &= 2x_1 - x_2 + \exp(x_1) - 1, \\
F_i(x) &= -x_{i-1} + 2x_i - x_{i+1} + \exp(x_i) - 1, \quad i = 2, 3, \ldots, n - 1, \\
F_n(x) &= -x_{n-1} + 2x_n + \exp(x_n) - 1,
\end{align*}
\]
and \( X = \{ x \in \mathbb{R}^n \mid x_i \geq 0, \quad i = 1, 2, \ldots, n \} \).
Throughout the numerical experiments, the parameters used in Algorithm 5.1 are chosen as follows: $\eta = 1$, $r = 0.001$, $\sigma = 0.001$, $\rho = 0.5$ and $\tau = 1$; the parameters used in Algorithm XZA are chosen as follows: $\xi = 1$, $\rho = 0.5$ and $\sigma = 0.001$, while the parameters used in Algorithm OLA are the same as that in [16]. Furthermore, all runs are terminated whenever 
\[ \|F(x_k)\| \leq 10^{-5}, \text{ or } \|F(z_k)\| \leq 10^{-5}. \]
We also terminate the iteration when it exceeds the preset iteration limit 5000.

The numerical results are listed in Appendix 1-Appendix 3 with the form of $NI/NF/CPU$, where we report the test problems (P), the dimension of variables ($n$), the initial points $x_0$, the number of iterations ($NI$), the number of function evaluations ($NF$), and the cpu-time in seconds ($CPU$).

We use the performance profile proposed by Dolan and Moré [6] to display the performance of each implementation on the set of test problems with the number of variables $n =$5000, 10000 and 50000, respectively. Based on the testing results in Appendix 1-Appendix 3, we draw the performance profiles in Figs. 1, 2 and 3 for $NI$, $NF$ and $CPU$, respectively.
From Figs. 1, 2 and 3, we observe the following facts:

- For the test problems, Algorithm 5.1 performs worse than Algorithm OLA, and both of them performs better than Algorithm XZA in terms of the number of iterations (NI), since with the least number of iterations, Algorithm 5.1 and Algorithm OLA successfully solve about 36% and 62% of the test problems, respectively, while in the same situation, the percentage of XZA is 12%.

- For the test problems, Algorithm 5.1 performs better than Algorithms OLA and XZA in terms of function evaluations, since with the least number of function evaluations, Algorithm 5.1 successfully solves about 62.5% of the test problems, while in the same situation, the percentages of OLA and XZA are 47.5% and 1%, respectively.

- For the test problems, Algorithm 5.1 has great advantage over Algorithms OLA and XZA in terms of the CPU time, since with the least average CPU time, Algorithm 5.1 successfully solves about 74% of the test problems, while in the same situation, the percentages of Algorithms OLA and XZA are about 18% and 8%, respectively.

Therefore, we could say that Algorithm 5.1 is computationally preferable to Algorithms OLA and XZA, at least for the set of test problems.

While it would be unwise to draw some firm conclusions from the limited numerical results, they indicate some promise for the new method proposed in this paper, compared with the existing Algorithms OLA and XZA.

6. Application in obstacle and free boundary problems.

6.1. Description of the problem. In recent years, the obstacle problem has attracted much attention due to its wide application background (see [5] for instance). This problem is to find the equilibrium position of an elastic membrane that is held at a fixed position on its boundary and lies over an obstacle.

Consider stretching an elastic string fixed at the endpoints (0, 0) and (4, 0) over an obstacle defined by a parabola function $f(x) = 1 - (x - 2.2)^2$ (see Figure 4). Notice that the position of the string will be determined by $f(x)$ for $x$ between the unknown points $P$ and $Q$, and that in the intervals $0 \leq x \leq P$ and $Q \leq x \leq 4$, the string
A unified derivative-free projection method model

Figure 4. An elastic string stretched over an obstacle

will lie along straight line segments connecting $(0,0)$ to $(P,f(P))$ and $(Q,f(Q))
to $(4,0)$, respectively. Let the function $u$ represent the equilibrium position of the
string. Then $u$ satisfies the conditions as follows:

$$
\begin{aligned}
&u(0) = 0, \\ &u(4) = 0, \\ &u'(P) = f'(P), \\ &u'(Q) = f'(Q), \\ &u(x) = f(x), \text{ for } P \leq x \leq Q, \\ &u''(x) = 0, \text{ for } 0 < x < P \text{ or } Q < x < 4.
\end{aligned}
$$

(67)

It is difficult for us to solve problem (67) directly. Instead, it is transformed into a
linear complementarity problem (LCP) [5], which avoids requiring the free boundaries $P$ and $Q$.

As noted in [29], the function $u$ satisfies the following conditions:

$$
\begin{aligned}
&u(0) = 0, \\ &u(4) = 0, \\ &u(x) \geq f(x), \text{ for } 0 \leq x \leq 4, \\ &u''(x) \leq 0, \\ &(u(x) - f(x))u''(x) = 0.
\end{aligned}
$$

(68)

We can solve this system (68) numerically using a central difference scheme, i.e.,
given a regular mesh with stepsize $h = \frac{4}{n}$, the function $u$ is approximated discretely
by the vector $\tilde{u} = (u_1, u_2, \cdots, u_n)$, where $u_i = f(x_i)$ with $x_i = x_0 + ih$ ($i = 0, 1, \cdots, n$) and $x_0 = 0$. Then, the system (68) is approximated by

$$
\begin{aligned}
&u_0 = u_n = 0, \\ &u_{i-1} - 2u_i + u_{i+1} \leq 0, \\ &u_i - f(x_i) \geq 0, \\ &u_i - f(x_i) \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = 0, \\ &i = 1, 2, \cdots, n - 1.
\end{aligned}
$$

(69)

By the simple transformation $y_i = u_i - f(x_i)$, the system (69) is equivalent to the
following LCP: Find $y \in \mathbb{R}^{n-1}$ such that

$$
w \geq 0, y \geq 0, w^T y = 0,
$$

where $w = My + q$ with $M \in \mathbb{R}^{(n-1) \times (n-1)}$ and $q \in \mathbb{R}^{n-1}$ defined respectively by

$$
M = \begin{pmatrix}
2 & -1 & & \\
-1 & 2 & -1 & \\
& & \ddots & \ddots & \ddots \\
& & \ddots & -1 & \\
& & & -1 & 2
\end{pmatrix},
q = \begin{pmatrix}
-2f(x_1) + f(x_2) \\
f(x_1) - 2f(x_2) + f(x_3) \\
\vdots \\
f(x_{n-1}) - 2f(x_{n-2}) + f(x_{n-1}) \\
f(x_{n-2}) - 2f(x_{n-1})
\end{pmatrix}.
$$
It is clear that a vector $y$ is a solution of (70) if and only if it satisfies the following nonsmooth equations [18]

$$F(y) = \min\{y, My + q\} = 0,$$  \hspace{1cm} (71)

where the function $F$ is vector-valued with the $\min$ interpreted as componentwise minimum. Thus we can solve Eqs. (71) by Algorithm 5.1 effectively.

Let $y = (y_1, y_2, \cdots, y_{n-1})$ be the solution to Eqs. (71) (or LCP (70)). Then we obtain the discrete approximation $\hat{u}$ to $u$ at the interior grid points by the relation $u_i = y_i + f(x_i)$, $i = 1, 2, \cdots, n - 1$.

6.2. **Numerical results.** Using the above-mentioned algorithms to solve the obstacle problem, we report some numerical results in this subsection. Throughout the numerical experiments, the parameters used are chosen to be the same as that in Section 5, while the initial points are generated randomly in the interval $[0, 1]$. Moreover, We stopped the iterations when either the iteration number exceeded 10000 or the inequality $\|F(x_k)\| \leq 10^{-4}$ is satisfied.

The detailed numerical results are listed in Table 1 with the form of CPU/FN, where we report the dimension of variables ($n$), the cpu-time in seconds ($CPU$), and the final norm of $F$ at $y_k$ ($FN$).

From Table 1, we can see that for the obstacle problem, Algorithm 5.1 can be competitive with Algorithms OLA and XZA, in terms of the CPU time and the final objective accuracy.

| $n$  | Algorithm5.1 | OLA (CPU/FN) | XZA (CPU/FN) |
|------|--------------|--------------|--------------|
| 50   | 1.664626/9.7218e-06 | 1.814302/7.1023e-06 | 4.972539/9.9601e-06 |
| 100  | 10.632101/5.9094e-05 | 10.647692/8.0831e-05 | 39.520620/1.3001e-05 |
| 500  | 90.768111/0.0161 | 167.333069/0.0174 | 459.419363/0.0279 |

7. **Concluding remarks.** In this paper, we propose a unified derivative-free projection method model for large-scale nonlinear equations with convex constraints. The main property of the proposed method is that we establish the global convergence without the Lipschitz continuity and monotonicity assumptions. Furthermore, the convergence rate of the proposed method is also established under some reasonable assumptions. In order to verify the feasibility and effectiveness of the model, a practical algorithm is devised and the corresponding numerical experiments are reported, which show that the proposed practical method is efficient and can be applied to solve large-scale nonsmooth equations. Another contribution of this paper is the use of our practical method to solving the obstacle problem. The numerical experiments on obstacle problems show that our practical method is competitive with the compared ones.

Since the most computational cost of each algorithm for problem (1) is to determine the search direction $d_k$ and find the stepsize $\alpha_k$ in line search, we will study some more effective methods for constructing a descent direction $d_k$ and an inexpensive line search in our future research.

**Acknowledgments.** The authors would like to thank the anonymous referees and the associate editor for their patience and valuable comments and suggestions that greatly improved this paper.
## A unified derivative-free projection method model

### Appendix 1. Numerical Results (n=5000)

| P | x0 | Algorithm 5.1 (NI/NF/CPU) | OLA (NI/NF/CPU) | XZA (NI/NF/CPU) |
|---|----|---------------------------|-----------------|-----------------|
| **P1** | x01 | 7/20/0.106104 | 7/20/0.147386 | 9/57/0.171656 |
|      | x02 | 8/18/0.092664 | 8/18/0.110545 | 18/85/0.282896 |
|      | x03 | 8/18/0.107345 | 8/18/0.128645 | 21/100/0.316561 |
|      | x04 | 8/18/0.108259 | 8/18/0.121497 | 23/109/0.368111 |
| **P2** | x01 | 20/53/0.305684 | 22/53/0.329380 | 34/197/0.606628 |
|      | x02 | 19/55/0.276160 | 17/48/0.279440 | 18/76/0.269347 |
|      | x03 | 24/64/0.301659 | 18/51/0.253223 | 29/131/0.439021 |
|      | x04 | 20/56/0.266451 | 26/69/0.368655 | 29/150/0.469068 |
| **P3** | x01 | 8/24/0.108566 | 8/24/0.131143 | 14/76/0.218523 |
|      | x02 | 13/34/0.169848 | 16/40/0.275883 | 18/100/0.302666 |
|      | x03 | 9/21/0.120308 | 8/19/0.107343 | 23/129/0.374192 |
|      | x04 | 10/23/0.115714 | 8/19/0.146601 | 21/116/0.328151 |
| **P4** | x01 | 4/10/0.072528 | 5/12/0.120855 | 14/69/0.252921 |
|      | x02 | 4/10/0.075093 | 5/12/0.120413 | 18/89/0.338554 |
|      | x03 | 4/10/0.075465 | 5/12/0.084672 | 17/83/0.307171 |
|      | x04 | 4/10/0.062832 | 5/12/0.104822 | 18/90/0.309595 |
| **P5** | x01 | 19/48/0.254132 | 15/42/0.247090 | 18/105/0.310756 |
|      | x02 | 17/43/0.222260 | 14/35/0.205039 | 20/101/0.298898 |
|      | x03 | 16/42/0.176401 | 12/31/0.224047 | 19/113/0.320323 |
|      | x04 | 22/55/0.241117 | 25/65/0.330554 | 23/137/0.380518 |
| **P6** | x01 | 42/110/0.547386 | 36/100/0.574953 | 38/256/0.724176 |
|      | x02 | 58/145/0.638433 | 37/115/0.530955 | 36/248/0.653960 |
|      | x03 | 57/144/0.617889 | 33/99/0.485122 | 37/261/0.674588 |
|      | x04 | 80/196/0.837358 | 66/195/0.879832 | 112/939/2.271418 |
| **P7** | x01 | 58/139/21.062102 | 54/161/26.305742 | 66/525/55.153478 |
|      | x02 | 50/120/18.118033 | 45/125/21.350917 | 48/419/41.649118 |
|      | x03 | 64/152/23.787516 | 59/172/27.201032 | 82/615/2.802815 |
|      | x04 | 74/169/26.042134 | 64/175/29.736745 | 73/573/65.573356 |
| **P8** | x01 | 7/17/0.096286 | 8/19/0.120349 | 11/58/0.167877 |
|      | x02 | 6/14/0.084938 | 6/15/0.111063 | 16/84/0.241674 |
|      | x03 | 7/16/0.082715 | 6/14/0.085867 | 23/120/0.343313 |
|      | x04 | 7/16/0.097362 | 6/14/0.103871 | 22/109/0.347084 |
| **P9** | x01 | 48/118/0.576797 | 72/198/0.951439 | 28/185/0.456993 |
|      | x02 | 27/73/0.283217 | 50/145/0.750436 | 626/6478/13.990421 |
|      | x03 | 80/187/0.784093 | 84/203/0.893276 | 242/2464/5.289804 |
|      | x04 | 93/207/0.937570 | 95/229/1.041198 | 429/4558/9.745147 |
| **P10** | x01 | 77/176/0.910051 | 61/175/0.945501 | 92/709/1.844008 |
|      | x02 | 46/111/0.497183 | 46/125/0.737076 | 47/409/1.037076 |
|      | x03 | 69/161/0.717932 | 59/172/0.817132 | 90/738/1.856715 |
|      | x04 | 74/169/0.825584 | 62/169/0.857298 | 78/593/1.517098 |
## APPENDIX-2. Numerical Results (n=10000)

| P    | x0  | Algorithm5.1 (NI/NF/CPU/FN) | OLA (NI/NF/CPU/FN) | XZA (NI/NF/CPU/FN) |
|------|-----|-----------------------------|--------------------|--------------------|
| P1   | x01 | 7/20/0.198678               | 7/20/0.220884      | 9/57/0.305319      |
|      | x02 | 8/18/0.160055               | 8/18/0.182768      | 18/85/0.499904     |
|      | x03 | 9/21/0.212598               | 8/18/0.239102      | 24/118/0.693524    |
|      | x04 | 9/21/0.204122               | 8/18/0.216345      | 24/120/0.687622    |
| P2   | x01 | 26/68/0.717034              | 21/51/0.815161     | 28/150/0.927982    |
|      | x02 | 19/55/0.481576              | 18/50/0.493633     | 28/133/0.891733    |
|      | x03 | 25/67/0.643779              | 19/53/0.686125     | 27/120/0.760968    |
|      | x04 | 21/60/0.484723              | 29/82/0.682750     | 50/292/1.683411    |
| P3   | x01 | 8/24/0.194794               | 8/24/0.291192      | 14/76/0.419421     |
|      | x02 | 19/50/0.413520              | 14/37/0.343390     | 19/109/0.581618    |
|      | x03 | 9/21/0.204539               | 8/19/0.206205      | 20/106/0.590355    |
|      | x04 | 9/21/0.195874               | 8/19/0.183434      | 20/106/0.586361    |
| P4   | x01 | 4/10/0.124562               | 5/12/0.184362      | 18/89/0.613673     |
|      | x02 | 4/10/0.126524               | 5/12/0.158129      | 17/89/0.592864     |
|      | x03 | 4/10/0.160094               | 5/12/0.171573      | 21/109/0.731632    |
|      | x04 | 4/10/0.119214               | 5/12/0.162090      | 18/93/0.641634     |
| P5   | x01 | 9/26/0.228399               | 20/52/0.506135     | 11/74/0.390610     |
|      | x02 | 17/43/0.356433              | 14/35/0.353195     | 20/101/0.568257    |
|      | x03 | 11/29/0.247406              | 16/41/0.339989     | 21/131/0.685652    |
|      | x04 | 16/41/0.344421              | 9/23/0.212989      | 26/156/0.830173    |
| P6   | x01 | 62/156/1.352174             | 42/118/1.087519    | 150/1173/5.675434  |
|      | x02 | 65/161/1.349872             | 51/147/1.424847    | 35/257/1.306742    |
|      | x03 | 61/156/1.307122             | 42/123/0.956588    | 36/233/1.193513    |
|      | x04 | 82/200/1.695587             | 69/201/1.508617    | 65/531/2.554975    |
| P7   | x01 | 74/174/2.052570             | 71/201/2.788763    | 75/637/5.570837    |
|      | x02 | 43/104/1.209124             | 40/111/1.513102    | 46/374/3.268021    |
|      | x03 | 79/183/2.201611             | 63/187/2.423513    | 89/697/6.127831    |
|      | x04 | 71/166/2.027387             | 59/167/2.236851    | 77/587/5.224853    |
| P8   | x01 | 7/17/0.156426               | 8/19/0.232142      | 11/58/0.328779     |
|      | x02 | 6/14/0.138376               | 6/15/0.165895      | 16/82/0.463611     |
|      | x03 | 7/16/0.171131               | 6/14/0.121279      | 22/115/0.631024    |
|      | x04 | 7/16/0.158673               | 6/14/0.138015      | 19/95/0.550413     |
| P9   | x01 | 66/156/1.222693             | 68/185/1.545983    | 28/181/0.877929    |
|      | x02 | 30/80/0.665042              | 51/146/1.304443    | 304/3257/13.516349 |
|      | x03 | 52/124/1.011101             | 62/171/1.243855    | 25/161/0.775442    |
|      | x04 | 132/289/2.408427            | 85/247/1.740591    | 170/1829/7.603185  |
| P10  | x01 | 74/174/1.488389             | 65/187/1.657350    | 75/670/3.247027    |
|      | x02 | 43/104/0.886894             | 40/111/1.065424    | 47/380/1.882825    |
|      | x03 | 79/183/1.673381             | 60/178/1.413495    | 90/682/3.441846    |
|      | x04 | 70/158/1.415521             | 63/173/1.331438    | 77/608/3.039018    |
### APPENDIX-3. Numerical Results \( (n=50000) \)

| \( P \) | \( x_0 \) | Algorithm 5.1 \((N1/NCF/CPU/FN)\) | OLA \((N1/NCF/CPU/FN)\) | XZA \((N1/NCF/CPU/FN)\) |
|---|---|---|---|---|
| P1 | 7/20/0.657066 | 8/22/0.875999 | 8/56/1.517435 |
| 8/02 | 8/18/0.681896 | 8/18/0.938792 | 17/84/2.375818 |
| 9/03 | 9/20/0.741662 | 8/18/0.802363 | 19/94/2.694325 |
| 9/04 | 9/20/0.694816 | 8/18/0.811017 | 16/75/2.65480 |
| P2 | 26/68/2.337139 | 24/59/2.693146 | 27/142/4.356456 |
| 21/50/2.183263 | 19/54/2.467531 | 24/112/3.488163 |
| 26/70/2.410611 | 20/56/2.372021 | 30/145/4.452101 |
| 24/66/2.207827 | 33/101/3.974809 | 44/268/7.573123 |
| P3 | 8/24/0.750872 | 8/24/0.997774 | 13/75/2.018475 |
| 20/53/1.996364 | 13/32/1.478025 | 25/111/3.842478 |
| 10/23/0.795941 | 8/19/0.787655 | 21/112/2.965393 |
| 10/23/0.790232 | 8/19/0.853699 | 19/106/2.837264 |
| P4 | 4/10/0.422089 | 5/12/0.638775 | 12/63/2.100308 |
| 4/10/0.392918 | 5/12/0.670729 | 14/77/2.520254 |
| 4/10/0.404484 | 5/12/0.613995 | 15/80/2.644590 |
| 4/10/0.395571 | 5/12/0.609066 | 18/104/3.319601 |
| P5 | 13/36/1.196057 | 18/50/1.981564 | 12/84/2.182320 |
| 17/43/1.549754 | 14/35/1.649476 | 19/100/2.731096 |
| 18/46/1.486943 | 9/22/0.900211 | 20/117/3.099405 |
| 13/33/1.274044 | 18/46/1.799406 | 21/124/3.285520 |
| P6 | 37/99/3.259450 | 52/147/5.624482 | 35/237/6.105914 |
| 70/174/7.176086 | 60/183/7.304829 | 31/200/5.084945 |
| 40/111/3.555434 | 48/142/5.321046 | 37/257/6.352924 |
| 90/222/7.242280 | 74/220/7.669101 | 67/517/12.250560 |
| P7 | 74/180/10.428094 | 66/199/12.945232 | 97/750/33.050593 |
| 45/106/6.392777 | 44/120/8.705869 | 47/316/14.364987 |
| 76/174/10.662135 | 68/195/12.680697 | 105/880/38.275212 |
| 69/189/9.206876 | 62/177/11.380591 | 78/591/26.327068 |
| P8 | 7/17/0.589308 | 8/19/0.789388 | 11/62/1.684645 |
| 6/14/0.510422 | 6/14/0.658295 | 12/63/1.729378 |
| 7/16/0.568656 | 7/16/0.838928 | 18/92/2.539715 |
| 7/16/0.561441 | 7/16/0.649622 | 20/100/2.770224 |
| P9 | 75/175/5.741871 | 71/208/7.327877 | 27/179/4.192673 |
| 31/81/2.864117 | 47/144/5.329594 | 22/149/3.531229 |
| 89/204/6.876357 | 70/202/8.017477 | 22/145/3.432510 |
| 90/201/6.645084 | 91/264/8.725611 | 132/1009/23.211413 |
| P10 | 74/180/6.245719 | 67/200/7.682366 | 106/800/20.133220 |
| 45/106/6.495596 | 44/120/5.432655 | 47/319/8.291141 |
| 76/174/6.116757 | 68/196/7.978214 | 104/830/20.604733 |
| 69/156/5.530603 | 62/177/6.546336 | 78/591/14.942913 |

### REFERENCES

[1] A. B. Abubakar, P. Kumam and H. Mohammad, A note on the spectral gradient projection method for nonlinear monotone equations with applications, *Comput. Appl. Math.*, 39 (2020), Paper No. 129, 35 pp.
[2] K. Amini and A. Kamandi, A new line search strategy for finding separating hyperplane in projection-based methods, Numer. Algorithms, 70 (2015), 559–570.

[3] A. M. Awwal, P. Kumama and A. B. Abubakar, A modified conjugate gradient method for monotone nonlinear equations with convex constraints, Applied Numerical Mathematics, 145 (2019), 507–520.

[4] S. Babaie-Kafaki and Z. Aminifard, Two-parameter scaled memoryless BFGS methods with a nonmonotone choice for the initial step length, Numer. Algorithms, 82 (2019), 1345–1357.

[5] S. C. Billups and K. G. Murty, Complementarity problems, J. Comput. Appl. Math., 124 (2000), 303–318.

[6] E. D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, Math. Program., 91 (2002), 201–213.

[7] P. Gao and C. He, An efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints, Calcolo, 55 (2018), Paper No. 53, 17 pp.

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

[9] C.-X. Jia and D.-T. Zhu, Projected gradient trust-region method for solving nonlinear systems with convex constraints, Appl. Math. J. Chinese Univ. Ser. B, 26 (2011), 57–69.

[10] C. Kanzow, N. Yamashita and M. Fukushima, Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints, J. Comput. Appl. Math., 173 (2005), 321–343.

[11] M. Koorapetse, P. Kaelo and E. R. Offen, A scaled derivative-free projection method for solving nonlinear monotone equations, Bull. Iranian Math. Soc., 45 (2019), 755–770.

[12] J. Liu and Y. Feng, A derivative-free iterative method for nonlinear monotone equations with convex constraints, Numerical Algorithms, 82 (2019), 245–262.

[13] J. Liu and S. Li, Multivariate spectral DY-type projection method for convex constrained nonlinear monotone equations, J. Ind. Manag. Optim., 13 (2017), 283–295.

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

[15] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[16] Y. Ou and J. Li, A new derivative-free SCG-type projection method for nonlinear monotone equations with convex constraints, J. Appl. Math. Comput., 56 (2018), 195–216.

[17] Y. Ou and Y. Liu, Supermemory gradient methods for monotone nonlinear equations with convex constraints, Comput. Appl. Math., 36 (2017), 259–279.

[18] J.-S. Pang, Inexact Newton methods for the nonlinear complementary problem, Math. Programming, 36 (1986), 54–71.

[19] B. T. Polyak, Introduction to Optimization, Optimization Software Incorporation, Publications Division, New York, NY, USA, 1987.

[20] M. V. Solodov and B. F. Svaiter, A globally convergent inexact Newton method for system of monotone equations, in: M. Fukushima and L. Qi (Eds.), Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, Kluwer Academic Publishers, Dordrecht, (1999), 355–369.

[21] M. V. Solodov and B. F. Svaiter, A new projection method for variational inequality problems, SIAM J. Control Optim., 37 (1999), 765–776.

[22] W. Y. Sun and Y. X. Yuan, Optimization Theory and Methods: Nonlinear Programming, Springer, New York, 2006.

[23] Z. Wan, J. Guo, J. J. Liu and W. Y. Liu, A modified spectral conjugate gradient projection method for signal recovery, Signal Image Video Process., 12 (2018), 1455–1462.

[24] C. Wang, Y. Wang and C. Xu, A projection method for a system of nonlinear monotone equations with convex constraints, Math. Methods Oper. Res., 66 (2007), 33–46.

[25] X. Y. Wang, S. J. Li and X. P. Kou, A self-adaptive three-term conjugate gradient method for monotone nonlinear equations with convex constraints, Calcolo, 53 (2016), 133–145.

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

[27] Y. Xiao and H. Zhu, A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing, J. Math. Anal. Appl., 405 (2013), 310–319.
[28] Z. Yu, J. Lin, J. Sun, Y. Xiao, L. Liu and Z. Li, Spectral gradient projection method for monotone nonlinear equations with convex constraints, *Appl. Numer. Math.*, 59 (2009), 2416–2423.

[29] Y.-B. Zhao and D. Li, Monotonicity of fixed point and normal mapping associated with variational inequality and applications, *SIAM J. Optim.*, 11 (2001), 962–973.

[30] L. Zheng, A new projection algorithm for solving a system of nonlinear equations with convex constraints, *Bull. Korean Math. Soc.*, 50 (2013), 823–832.

Received August 2020; revised March 2021; early access August 2021.

E-mail address: ouyigui@126.com
E-mail address: 990661@hainanu.edu.cn