



A hybrid three-term conjugate gradient projection method for constrained nonlinear monotone equations with applications

Jianghua Yin¹ · Jinbao Jian² · Xianzhen Jiang² · Meixing Liu³ · Lingzhi Wang⁴

Received: 18 May 2020 / Accepted: 9 November 2020 / Published online: 4 January 2021
© Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In this paper, based on the three-term conjugate gradient method and the hybrid technique, we propose a hybrid three-term conjugate gradient projection method by incorporating the adaptive line search for solving large-scale nonlinear monotone equations with convex constraints. The search direction generated by the proposed method is close to the one yielded by the memoryless BFGS method, and has the sufficient descent property and the trust region property independent of line search technique. Under some mild conditions, we establish the global convergence of the proposed method. Our numerical experiments show the effectiveness and robustness of the proposed method in comparison with two existing algorithms in the literature. Moreover, we show applicability and encouraging efficiency of the proposed method by extending it to solve sparse signal restoration and image de-blurring problems.

Keywords Derivative-free projection method · Three-term conjugate gradient method · Convergence property · Compressed sensing

Mathematics subject classification (2010) 65K10 · 68U10

✉ Jinbao Jian
jianjb@gxu.edu.cn

¹ School of Mathematical Sciences, Inner Mongolia University, Hohhot, 010021, China

² College of Mathematics and Physics, Guangxi University for Nationalities, Nanning, 530006, China

³ Department of Mathematics, Shanghai University, Shanghai, 200444, China

⁴ School of Mechanical and Electrical Engineering, Guangxi Science and Technology Normal University, Laibin, 546199, China

1 Introduction

In this paper, we consider numerically solving the following constrained nonlinear monotone equations of the form:

$$F(x) = 0, \quad x \in \mathcal{C}, \quad (1)$$

where $\mathcal{C} \subset \mathbb{R}^n$ is a nonempty closed convex set, and F is a continuous and monotone mapping from \mathbb{R}^n to itself. The monotonicity of the mapping F means that:

$$(F(x) - F(y))^T (x - y) \geq 0, \quad \forall x, y \in \mathbb{R}^n. \quad (2)$$

We call problem (1) the unconstrained nonlinear monotone equations if $\mathcal{C} = \mathbb{R}^n$.

The problem (1) arises in many important modern applications in applied mathematics, economics and engineering, and so on. Moreover, some mathematical problems can be converted into the form of (1), such as some nonlinear monotone complementarity problems, the compressed sensing problems, and the optimal power flow control problems in electricity (see [1–4] for references). The wide applications of nonlinear monotone equations have motivated many researchers to study the efficient methods for solving (1). Up to now, numerous iterative methods have been developed to solve (1). Among these methods, we only focus on the derivative-free projection-based methods (DFPMs for short); see, for example, [5–19]. This kind of methods are quite suitable for solving large-scale constrained nonlinear monotone equations due to their simplicity and low storage requirement.

Inspired by the projection technique [20] and the spectral gradient method [21], Yu et al. [5] first extended the spectral gradient projection method in [22] from unconstrained nonlinear monotone equations to constrained nonlinear monotone equations. The reported numerical results in [5] show that the proposed method works quite well even for large-scale problems. Since then, it has attracted much attention, and many DFPMs have sprung up, such as the spectral gradient-type projection method [6, 7], the conjugate gradient-type projection method [8–15], the spectral conjugate gradient-type projection method [16–19], to mention just a few. Recently, the spectral gradient-type projection method has been further developed. Abubakar et al. [23] proposed a positive spectral gradient projection method whose search direction is a convex combination of two different positive spectral coefficients multiplied with the residual vector. Awwal et al. [24] presented a two-step spectral gradient projection method. Mohammad and Abubakar [25] further designed an adaptive spectral gradient projection method. The numerical results in [23–25] show that these methods work well and are more suitable compared to the similar methods for solving (1).

In [26], Wang et al. proposed a three-term conjugate gradient projection method based on the HS method [27] for solving (1). Subsequently, motivated by [26], Gao and He [28] proposed another three-term conjugate gradient projection method by choosing a part of the LS method in [29] as the conjugate parameter. Recently, Gao et al. [30] developed an adaptive three-term conjugate gradient projection method, in which the search direction is close to the direction of the self-scaling memoryless BFGS method proposed by Oren and Luenberger [31] in the sense of the minimum Frobenius norm. And they successfully applied this algorithm to solve the compressed sensing problems in [30]. Quite recently, Awwal et al. [32] proposed a

Perry-type derivative-free method based on the BFGS quasi-Newton search direction. The reported numerical results in [32] show that the proposed method is effective and promising for solving (1) and signal recovery problem. On the other hand, based on the ideas of hybrid conjugate gradient method in [33], Sun and Liu [34] developed a hybrid conjugate gradient projection method for solving (1), in which the search direction satisfies the sufficient descent property and the trust region property independent of line search technique. Some large-scale numerical results in [34] show that the proposed method is quite competitive for some fixed initial points.

Recently, based on the memoryless BFGS method proposed by Nocedal [35] and Shanno [36], Li [37–39] developed a family of three-term conjugate gradient methods, in which the search directions are close to that of the memoryless BFGS method. A large number of numerical comparisons indicate that the proposed methods are quite efficient for solving unconstrained optimization problems.

To the best of our knowledge, no existing such a conjugate gradient method in the literature can combine the three-term conjugate gradient method, whose search direction is close to that of the memoryless BFGS method, with the ideas of hybrid methods. Hence, it is interesting to design a hybrid three-term conjugate gradient projection method for solving problem (1) such that its search direction is close to that of the memoryless BFGS method and possesses the sufficient descent property and the trust region property independent of line search technique.

In this paper, inspired by the hybrid conjugate gradient projection method in [34] and the three-term conjugate gradient methods in [37–39], we propose a hybrid three-term conjugate gradient projection method for solving (1). The rest of the paper is organized as follows. In the next section, we describe in detail the motivation of this paper and propose the algorithm. In Section 3, we establish the global convergence under some mild conditions. In Section 4, we perform numerical experiments to illustrate the performance of the proposed method and apply it to solve compressed sensing problems in Section 5. Some conclusions are summarized in the last section.

Notation Throughout this paper, the symbols $\| \cdot \|$ and $\| \cdot \|_1$ denote the Euclidean norm and ℓ_1 -norm on \mathbb{R}^n , respectively. For convenience, we abbreviate $F(x_k)$ to F_k , and similar abbreviations are also used. For a nonempty closed convex set $\mathcal{C} \subset \mathbb{R}^n$, $P_{\mathcal{C}}[\cdot]$ denotes the projection mapping from \mathbb{R}^n onto the set \mathcal{C} , namely, $P_{\mathcal{C}}[x] = \arg \min\{\|y - x\| \mid y \in \mathcal{C}\}$.

2 Motivation and algorithm

In this section, we are devoted to developing a hybrid three-term conjugate gradient projection method for solving problem (1). To this end, we first briefly review the three-term conjugate gradient methods for solving unconstrained optimization problems proposed in [37–39] and the hybrid conjugate gradient projection method for solving problem (1) in [34].

As we all know, for solving large-scale unconstrained optimization problems:

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

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function, one of the most effective methods is the three-term conjugate gradient method, which dates back to Beale [40] and later developed by Nazareth [41], Zhang et al. [42], etc. They are of the form:

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

where $\alpha_k > 0$ is the steplength determined by some line search and d_k is the search direction generated by:

$$d_k = -F_k + \beta_k d_{k-1} + \theta_k y_{k-1}, \quad k \geq 1, \quad d_0 = -F_0, \tag{3}$$

where $F_k := \nabla f(x_k)$, β_k and θ_k are scalar parameters, and $y_{k-1} := F_k - F_{k-1}$. Different choices for the parameters β_k and θ_k correspond to different three-term conjugate gradient methods. Clearly, the three-term conjugate gradient method reduces to the classical conjugate gradient method when $\theta_k \equiv 0$ in (3). There are at least four well-known formulas for β_k , such as:

$$\beta_k^{\text{PRP}} = \frac{F_k^\top y_{k-1}}{\|F_{k-1}\|^2}, \quad \beta_k^{\text{HS}} = \frac{F_k^\top y_{k-1}}{d_{k-1}^\top y_{k-1}}, \quad \beta_k^{\text{DY}} = \frac{\|F_k\|^2}{d_{k-1}^\top y_{k-1}}, \quad \beta_k^{\text{LS}} = \frac{F_k^\top y_{k-1}}{-d_{k-1}^\top F_{k-1}}.$$

We refer the readers to the monograph [43] and the comprehensive survey [44] for more details.

Based on the memoryless BFGS method proposed by Nocedal [35] and Shanno [36], in which the search direction can be written as:

$$d_k^{\text{LBFGS}} = -F_k + \left(\frac{F_k^\top y_{k-1}}{d_{k-1}^\top y_{k-1}} - \frac{\|y_{k-1}\|^2 F_k^\top d_{k-1}}{(d_{k-1}^\top y_{k-1})^2} \right) d_{k-1} + \frac{F_k^\top d_{k-1}}{d_{k-1}^\top y_{k-1}} (y_{k-1} - s_{k-1}),$$

where $s_{k-1} := x_k - x_{k-1}$, Li [37] proposed a three-term HS conjugate gradient method, in which:

$$\beta_k^{\text{THS}} = \frac{F_k^\top y_{k-1}}{d_{k-1}^\top y_{k-1}} - \frac{\|y_{k-1}\|^2 F_k^\top d_{k-1}}{(d_{k-1}^\top y_{k-1})^2}, \quad \theta_k = t_k \frac{F_k^\top d_{k-1}}{d_{k-1}^\top y_{k-1}}, \tag{4}$$

and $0 \leq t_k \leq \bar{t} < 1$. In practical computation, the parameter \bar{t} is set to 0.3 and then t_k is calculated by:

$$t_k = \min \left\{ \bar{t}, \max \left\{ 0, 1 - \frac{y_{k-1}^\top s_{k-1}}{\|y_{k-1}\|^2} \right\} \right\}, \tag{5}$$

where $1 - \frac{y_{k-1}^\top s_{k-1}}{\|y_{k-1}\|^2}$ is the solution of the univariate minimal problem:

$$\min_{t \in \mathbb{R}} \|(y_{k-1} - s_{k-1}) - t \cdot y_{k-1}\|^2.$$

It is not hard to know that the search direction defined in (3)–(5) is close to that of the memoryless BFGS method in the sense of the above univariate minimum problem. Subsequently, based on the fact that $d_{k-1}^\top y_{k-1} = \|F_{k-1}\|^2$ holds when the objective function f is a strongly convex quadratic and the exact line search is used, Li [38]

proposed a three-term PRP conjugate gradient method with the search direction of the form (3) being generated by replacing $d_{k-1}^\top y_{k-1}$ with $\|F_{k-1}\|^2$ in (4), i.e.:

$$\beta_k^{\text{TPRP}} = \frac{F_k^\top y_{k-1}}{\|F_{k-1}\|^2} - \frac{\|y_{k-1}\|^2 F_k^\top d_{k-1}}{\|F_{k-1}\|^4}, \quad \hat{\theta}_k = t_k \frac{F_k^\top d_{k-1}}{\|F_{k-1}\|^2}, \tag{6}$$

and $0 \leq t_k \leq \bar{t} < 1$. Recently, based on the comprehensive analysis of [37, 38], Li [39] proposed a family of three-term conjugate gradient methods for solving unconstrained optimization problems. Interestingly, a remarkable feature of these methods is that the search directions defined by [37–39] always satisfy the sufficient descent property:

$$F_k^\top d_k \leq -\left(1 - \frac{(1 + \bar{t})^2}{4}\right) \|F_k\|^2.$$

On the other hand, based on the ideas of hybrid conjugate gradient method in [33], Sun and Liu [34] developed a hybrid conjugate gradient method and then extended it to solve problem (1). In [34], the search direction is determined by the following scheme:

$$d_k = -F_k + \beta_k^{\text{HCG}} d_{k-1}, \quad k \geq 1, \quad d_0 = -F_0,$$

where:

$$\beta_k^{\text{HCG}} = \frac{\|F_k\|^2 - \max\left\{0, \frac{\|F_k\|}{\|F_{k-1}\|} F_k^\top F_{k-1}\right\}}{\max\{\kappa \|d_{k-1}\| \|F_k\|, d_{k-1}^\top y_{k-1}\}},$$

with $\kappa > 1$ being a constant. It is easy to show that the search direction defined in the above two relations satisfies the sufficient descent property and the trust region property independent of line search technique; see Lemma 3.1 and Lemma 3.2 of [34], respectively. Notice that the parameter β_k^{HCG} can be viewed as a hybrid of β_k^{HS} and β_k^{DY} .

Inspired by the three-term conjugate gradient methods and the hybrid conjugate gradient projection method introduced above, we propose a hybrid three-term conjugate gradient method by incorporating the projection technique to solve non-linear monotone equations with convex constraints. The search direction d_k for the proposed method is given by:

$$d_k = -F_k + \beta_k^{\text{HTP}} d_{k-1} + \tilde{\theta}_k y_{k-1}, \quad k \geq 1, \quad d_0 = -F_0, \tag{7}$$

where:

$$\begin{aligned} w_k &= \max\{\mu \|d_{k-1}\| \|y_{k-1}\|, d_{k-1}^\top y_{k-1}, \|F_{k-1}\|^2\}, \\ \beta_k^{\text{HTP}} &= \frac{F_k^\top y_{k-1}}{w_k} - \frac{\|y_{k-1}\|^2 F_k^\top d_{k-1}}{w_k^2}, \\ \tilde{\theta}_k &= t_k \frac{F_k^\top d_{k-1}}{w_k}, \end{aligned} \tag{8}$$

with $0 \leq t_k \leq \bar{t} < 1$ and $\mu > 0$. Clearly, due to the definition of w_k in (8), the parameter β_k^{HTP} is a hybrid of β_k^{THS} in (4) and β_k^{TPRP} in (6). Similarly, the parameter

$\tilde{\theta}_k$ is a hybrid of θ_k in (4) and $\hat{\theta}_k$ in (6). Thus, the search direction defined in (7) is close to that of the memoryless BFGS method when $t_k = 1 - \frac{y_{k-1}^\top s_{k-1}}{\|y_{k-1}\|^2}$. Notice that the first term of the maximum function from the first relation in (8) is chosen to make the search direction have a trust region property; see Lemma 2 below. It is easy to see that $w_k > 0$ due to the existence of $\|F_{k-1}\|^2$ in (8). Therefore, the parameters β_k^{HTP} and $\tilde{\theta}_k$ in (8) are both well defined.

We now give the steps of our algorithm which we call the hybrid three-term conjugate gradient projection (HTTCGP) method.

Algorithm 1 (HTTCGP)

Step 0. Given an initial point $x_0 \in \mathcal{C}$, the parameters $0 \leq \bar{t} < 1$, $\mu > 0$, $\varsigma > 0$, $0 < \rho < 1$, $\sigma > 0$, $\epsilon > 0$, $0 < \lambda \leq \nu$ and $0 < \gamma < 2$. Set $k := 0$.

Step 1. Compute F_k . If $\|F_k\| \leq \epsilon$, stop. Otherwise go to Step 2.

Step 2. Compute d_k using (7). Stop if $d_k = 0$; otherwise go to Step 3.

Step 3. Set $z_k = x_k + \alpha_k d_k$, where $\alpha_k = \varsigma \rho^{i_k}$ with i_k being the smallest nonnegative integer i such that

$$-F(z_k)^\top d_k \geq \sigma \alpha_k P_{[\lambda, \nu]}[\|F(z_k)\|] \cdot \|d_k\|^2, \tag{9}$$

where for any $u \in \mathbb{R}$, the projection operator $P_{[\lambda, \nu]}[\cdot]$ is defined as

$$P_{[\lambda, \nu]}[u] = \begin{cases} \nu, & \text{if } u \geq \nu, \\ \lambda, & \text{if } u \leq \lambda, \\ u, & \text{otherwise.} \end{cases}$$

Step 4. Compute the next iteration by

$$x_{k+1} = P_{\mathcal{C}} [x_k - \gamma \cdot \xi_k F(z_k)], \tag{10}$$

where

$$\xi_k = \frac{F(z_k)^\top (x_k - z_k)}{\|F(z_k)\|^2} \geq 0. \tag{11}$$

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

Remark 1 (i) The adaptive line search (9) was proposed by our recent work in [45]. From the definition of $P_{[\lambda, \nu]}[\cdot]$, we have that $\lambda \leq P_{[\lambda, \nu]}[u] \leq \nu$ for any $u \in \mathbb{R}$. Introducing the parameter ν is to prevent the right-hand side of (9) from being too large, especially when x_k is far from the solution set of (1) and $\|F(z_k)\|$ is also large. This can effectively reduce the computational cost of the line search. The parameter λ is chosen to guard against the worst possible outcome, i.e., $F(z_k) = 0$ and $z_k \notin \mathcal{C}$. Indeed, from step 2 of Algorithm 1, we have $\|d_k\| > 0$. Combining this with the fact that $P_{[\lambda, \nu]}[\|F(x_k + \varsigma \rho^i d_k)\|] \geq \lambda > 0$, we know that the right-hand side of (9) is positive for any finite nonnegative integer i . Thus, we see immediately from Lemma 3 below that $F(z_k) \neq 0$ holds.

(ii) The adaptive line search (9) is a derivative-free line search and contains two well-known line search procedures in [22, 46] which were used in many of works. In fact, if $\lambda = \nu = 1$, then $P_{[\lambda, \nu]}[\|F(z_k)\|] \equiv 1$. Thus, (9) reduces to the line search

used in [22]. If $\lambda = 0$ and ν is sufficiently large, such as $\nu \geq \|F_0\|$, then we have $P_{[\lambda, \nu]}[\|F(z_k)\|] \equiv \|F(z_k)\|$. Therefore, (9) is equivalent to the line search used in [46]. Notice that when $\lambda = 0$ and $\nu = 1$, (9) is equivalent to the line search recently proposed by Sun and Tian in [47]. Quite recently, Awwal et al. [24] proposed another adaptive line search, that is, $\alpha_k = \varsigma \rho^{i_k}$ with i_k being the smallest nonnegative integer i such that:

$$-F(x_k + \alpha_k d_k)^\top d_k \geq \sigma \alpha_k \|F(x_k + \alpha_k d_k)\|^{1/c} \cdot \|d_k\|^2, \quad c \geq 1. \tag{12}$$

Clearly, if $c = 1$, (12) can be viewed as a special case of (9), provided that $\lambda = 0$ and ν is sufficiently large. On the other hand, if c is sufficiently large, then (12) behaves like the adaptive line search (9) with $\lambda = \nu = 1$.

(iii) Notice that $\gamma \in (0, 2)$ in (10) is a relaxation factor whose suitable value may speed up convergence for the projection and contraction methods as stated in [48]. In practical computation, one can take a larger relaxation factor $\gamma \in [1, 2)$.

3 Convergence property

In this section, we analyze the global convergence of the proposed method. To this end, we need the following standard assumptions used in most DFPMs.

(A1) The solution set of problem (1), denoted by \mathcal{S} , is nonempty.

(A2) The mapping F is Lipschitz continuous on \mathbb{R}^n , i.e., there is a constant $L > 0$ such that:

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{13}$$

We now start our convergence analysis by giving a well-known nonexpansive property of the projection operator [49].

Lemma 1 *Let C be a nonempty closed convex subset of \mathbb{R}^n . Then:*

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

The next lemma shows that the search direction defined in (7) has some nice properties independent of line search technique.

Lemma 2 *For all $k \geq 0$, the search direction sequence $\{d_k\}$ generated by the HTTCGP always satisfies the sufficient descent property:*

$$F_k^\top d_k \leq -\left(1 - \frac{(1 + \bar{t})^2}{4}\right) \|F_k\|^2 \tag{14}$$

and the trust region property:

$$\left(1 - \frac{(1 + \bar{t})^2}{4}\right) \|F_k\| \leq \|d_k\| \leq \left(1 + \frac{1 + \bar{t}}{\mu} + \frac{1}{\mu^2}\right) \|F_k\| \tag{15}$$

with $0 \leq t_k \leq \bar{t} < 1$ and $\mu > 0$.

Proof Since $d_0 = -F_0$, we have $F_0^\top d_0 = -\|F_0\|^2$, which satisfies (14) for $k = 0$. From the definition of d_k in (7), we conclude that for all $k \geq 1$,

$$\begin{aligned} F_k^\top d_k &= -\|F_k\|^2 + \beta_k^{\text{HTP}} F_k^\top d_{k-1} + \tilde{\theta}_k F_k^\top y_{k-1} \\ &= -\|F_k\|^2 + \left(\frac{F_k^\top y_{k-1}}{w_k} - \frac{\|y_{k-1}\|^2 F_k^\top d_{k-1}}{w_k^2} \right) F_k^\top d_{k-1} + t_k \frac{F_k^\top d_{k-1}}{w_k} F_k^\top y_{k-1} \\ &= -\|F_k\|^2 + (1 + t_k) \frac{F_k^\top y_{k-1} F_k^\top d_{k-1}}{w_k} - \frac{\|y_{k-1}\|^2 (F_k^\top d_{k-1})^2}{w_k^2} \\ &= -\|F_k\|^2 + 2 \left(\frac{1 + t_k}{2} F_k^\top \right) \frac{y_{k-1} F_k^\top d_{k-1}}{w_k} - \frac{\|y_{k-1}\|^2 (F_k^\top d_{k-1})^2}{w_k^2} \\ &\leq -\|F_k\|^2 + \frac{(1 + t_k)^2}{4} \|F_k\|^2 + \frac{\|y_{k-1}\|^2 (F_k^\top d_{k-1})^2}{w_k^2} - \frac{\|y_{k-1}\|^2 (F_k^\top d_{k-1})^2}{w_k^2} \\ &= -\|F_k\|^2 + \frac{(1 + t_k)^2}{4} \|F_k\|^2 \leq -\left(1 - \frac{(1 + \bar{t})^2}{4}\right) \|F_k\|^2, \end{aligned}$$

which implies that (14) holds.

Again, from $d_0 = -F_0$, we obtain $\|d_0\| = \|F_0\|$, which satisfies (15) for $k = 0$. We next prove that the relation (15) holds for all $k \geq 1$. Note from step 2 of Algorithm 1 that $d_k \neq 0$ for all k . If $y_{k-1} = 0$, then the desired result immediately follows from the fact that $d_k = -F_k$ due to the definition of d_k in (7). Therefore, for the remainder of the proof, we assume that $y_{k-1} \neq 0$ for all $k \geq 1$. From (14) and the Cauchy-Schwarz inequality, we know that the first inequality in (15) holds. To proceed, we note from (8) that:

$$\begin{aligned} |\beta_k^{\text{HTP}}| &= \left| \frac{F_k^\top y_{k-1}}{w_k} - \frac{\|y_{k-1}\|^2 F_k^\top d_{k-1}}{w_k^2} \right| \\ &\leq \frac{\|F_k\| \|y_{k-1}\|}{w_k} + \frac{\|y_{k-1}\|^2 \|F_k\| \|d_{k-1}\|}{w_k^2} \\ &\leq \frac{\|F_k\| \|y_{k-1}\|}{\mu \|d_{k-1}\| \|y_{k-1}\|} + \frac{\|y_{k-1}\|^2 \|F_k\| \|d_{k-1}\|}{(\mu \|d_{k-1}\| \|y_{k-1}\|)^2} \\ &= \left(\frac{1}{\mu} + \frac{1}{\mu^2} \right) \frac{\|F_k\|}{\|d_{k-1}\|} \end{aligned}$$

and

$$\begin{aligned} |\tilde{\theta}_k| &= t_k \left| \frac{F_k^\top d_{k-1}}{w_k} \right| \\ &\leq \bar{t} \frac{\|F_k\| \|d_{k-1}\|}{w_k} \end{aligned}$$

$$\begin{aligned} &\leq \frac{\bar{\tau} \|F_k\| \|d_{k-1}\|}{\mu \|d_{k-1}\| \|y_{k-1}\|} \\ &= \frac{\bar{\tau} \|F_k\|}{\mu \|y_{k-1}\|}. \end{aligned}$$

Combining the above two relations with the definition of d_k in (7) yields:

$$\begin{aligned} \|d_k\| &= \|-F_k + \beta_k^{\text{HTP}} d_{k-1} + \tilde{\theta}_k y_{k-1}\| \\ &\leq \|F_k\| + |\beta_k^{\text{HTP}}| \|d_{k-1}\| + |\tilde{\theta}_k| \|y_{k-1}\| \\ &\leq \left(1 + \frac{1 + \bar{\tau}}{\mu} + \frac{1}{\mu^2}\right) \|F_k\|. \end{aligned}$$

This completes the proof. □

Remark 2 The first inequality in (15) implies that we find a solution of problem (1) when $d_k = 0$. This is the reason why we choose $d_k = 0$ as the stopping criterion in step 2 of Algorithm 1.

The following lemma not only shows that the adaptive line search (9) is well-defined but also provides a uniform lower bound of the sequence $\{\alpha_k\}$.

Lemma 3 (i) *Let $\{x_k\}$ and $\{d_k\}$ be the sequences generated by the HTTCGP. If the mapping F is continuous on \mathbb{R}^n , then for all $k \geq 0$, there exists a nonnegative integer i_k satisfying (9).*

(ii) *Suppose that assumption (A2) holds, then the steplength α_k yielded by (9) satisfies:*

$$\alpha_k \geq \alpha := \min \left\{ \varsigma, \frac{\rho \mu^4 [4 - (1 + \bar{\tau})^2]}{4(L + \sigma \nu) [\mu^2 + \mu(1 + \bar{\tau}) + 1]^2} \right\}. \tag{16}$$

Proof (i) We proceed by contradiction and assume that there exists an integer $k_0 \geq 0$ such that the adaptive line search (9) does not hold for any nonnegative integer i . Hence, for all $i \geq 0$, we have:

$$\begin{aligned} -F(x_{k_0} + \varsigma \rho^i d_{k_0})^\top d_{k_0} &< \sigma \varsigma \rho^i P_{[\lambda, \nu]}[\|F(x_{k_0} + \varsigma \rho^i d_{k_0})\|] \cdot \|d_{k_0}\|^2 \\ &\leq \sigma \nu \varsigma \rho^i \|d_{k_0}\|^2, \end{aligned}$$

where the second inequality follows from the fact that $P_{[\lambda, \nu]}[\cdot] \leq \nu$. Taking the limit in the above inequality, using the continuity of the mapping F and rearranging terms in the resulting relation, we obtain:

$$F(x_{k_0})^\top d_{k_0} \geq 0,$$

which contradicts (14) since $F(x_{k_0})^\top d_{k_0} \leq -\left(1 - \frac{(1+\bar{\tau})^2}{4}\right) \|F(x_{k_0})\|^2 < 0$. Thus, the desired result holds.

(ii) We now prove part (ii). Clearly, if $\alpha_k = \varsigma$, then (16) holds. Otherwise, $\widehat{\alpha}_k := \rho^{-1} \alpha_k$ does not satisfy the adaptive line search (9), namely:

$$-F(\widehat{z}_k)^\top d_k < \sigma \widehat{\alpha}_k P_{[\lambda, \nu]}[F(\widehat{z}_k)] \|d_k\|^2 \leq \sigma \nu \widehat{\alpha}_k \|d_k\|^2,$$

where $\widehat{z}_k = x_k + \widehat{\alpha}_k d_k$. This together with (13) and (14) yields:

$$\begin{aligned} \left(1 - \frac{(1 + \bar{t})^2}{4}\right) \|F_k\|^2 &\leq -F_k^T d_k = (F(\widehat{z}_k) - F_k)^T d_k - F(\widehat{z}_k)^T d_k \\ &< \widehat{\alpha}_k (L + \sigma v) \|d_k\|^2 = \rho^{-1} \alpha_k (L + \sigma v) \|d_k\|^2. \end{aligned}$$

Therefore:

$$\begin{aligned} \alpha_k &> \frac{\rho[4 - (1 + \bar{t})^2]}{4(L + \sigma v)} \frac{\|F_k\|^2}{\|d_k\|^2} \\ &\geq \frac{\rho[4 - (1 + \bar{t})^2]}{4(L + \sigma v)} \frac{\|F_k\|^2}{\left(1 + \frac{1+\bar{t}}{\mu} + \frac{1}{\mu^2}\right)^2 \|F_k\|^2} \\ &= \frac{\rho\mu^4[4 - (1 + \bar{t})^2]}{4(L + \sigma v)[\mu^2 + \mu(1 + \bar{t}) + 1]^2}, \end{aligned}$$

where the second inequality follows from the second inequality in (15). This yields the desired inequality (16). □

The following result is adapted from [20, Lemma 2.1] to show that for any $x^* \in \mathcal{S}$, the whole sequence $\{\|x_k - x^*\|\}$ generated by the HTTCGP is convergent.

Lemma 4 *Suppose that the assumptions (A1)-(A2) hold. Let $\{x_k\}$ and $\{z_k\}$ be the sequences generated by the HTTCGP. Then for any $x^* \in \mathcal{S}$ the sequence $\{\|x_k - x^*\|\}$ is convergent, therefore the sequence $\{x_k\}$ is bounded. Furthermore, it holds that:*

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0. \tag{17}$$

Proof We start our proof by noticing from the definition of z_k and (9) that:

$$\begin{aligned} F(z_k)^T(x_k - z_k) &\geq \sigma \alpha_k^2 P_{[\lambda, v]}[\|F(z_k)\|] \|d_k\|^2 \\ &\geq \sigma \lambda \alpha_k^2 \|d_k\|^2 = \sigma \lambda \|x_k - z_k\|^2 > 0. \end{aligned} \tag{18}$$

From (2) and $x^* \in \mathcal{S}$, we obtain that:

$$\begin{aligned} F(z_k)^T(x_k - x^*) &= F(z_k)^T(x_k - z_k) + F(z_k)^T(z_k - x^*) \\ &\geq F(z_k)^T(x_k - z_k) + F(x^*)^T(z_k - x^*) \\ &= F(z_k)^T(x_k - z_k). \end{aligned}$$

Combining the above two relations with (10), Lemma 1 and (11), we conclude that:

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|P_C[x_k - \gamma \cdot \xi_k F(z_k)] - P_C[x^*]\|^2 \\ &\leq \|x_k - \gamma \cdot \xi_k F(z_k) - x^*\|^2 \\ &= \|x_k - x^*\|^2 - 2\gamma \cdot \xi_k F(z_k)^T(x_k - x^*) + \gamma^2 \xi_k^2 \|F(z_k)\|^2 \\ &\leq \|x_k - x^*\|^2 - 2\gamma \cdot \xi_k F(z_k)^T(x_k - z_k) + \gamma^2 \xi_k^2 \|F(z_k)\|^2 \end{aligned}$$

$$\begin{aligned} &= \|x_k - x^*\|^2 - \gamma(2 - \gamma) \frac{(F(z_k)^T(x_k - z_k))^2}{\|F(z_k)\|^2} \\ &\leq \|x_k - x^*\|^2 - \gamma(2 - \gamma) \frac{\sigma^2 \lambda^2 \|x_k - z_k\|^4}{\|F(z_k)\|^2}, \end{aligned} \tag{19}$$

which implies $0 \leq \|x_{k+1} - x^*\| \leq \|x_k - x^*\|$ since $0 < \gamma < 2$ from the choice of γ in the HTTCGP. That is, the sequence $\{\|x_k - x^*\|\}$ is non-increasing and bounded below, and therefore is convergent. So, the boundedness of $\{x_k\}$ is at hand.

We next derive an upper bound of $\|F(z_k)\|$. To proceed, notice from the boundedness of $\{x_k\}$ and the continuity of F that there exists a constant $M > 0$ such that:

$$\|x_k\| \leq M, \|F_k\| \leq M, \forall k \geq 0.$$

Hence, we see from the Cauchy-Schwarz inequality, (2) and (18) that:

$$M \geq \|F_k\| \geq \frac{F_k^T(x_k - z_k)}{\|x_k - z_k\|} \geq \frac{F(z_k)^T(x_k - z_k)}{\|x_k - z_k\|} \geq \sigma \lambda \|x_k - z_k\| \geq \sigma \lambda \|z_k\| - \sigma \lambda M,$$

which shows that $\{z_k\}$ is bounded. Therefore, it again follows from the continuity of F that there exists a constant $\widehat{M} > 0$ such that:

$$\|F(z_k)\| \leq \widehat{M}, \forall k \geq 0.$$

This together with (19) implies:

$$\begin{aligned} \gamma(2 - \gamma) \frac{\sigma^2 \lambda^2}{\widehat{M}^2} \sum_{k=0}^{\infty} \|x_k - z_k\|^4 &\leq \sum_{k=0}^{\infty} \left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \right) \\ &= \|x_0 - x^*\|^2 - \lim_{k \rightarrow \infty} \|x_{k+1} - x^*\|^2 < \infty, \end{aligned}$$

which shows $0 = \lim_{k \rightarrow \infty} \|x_k - z_k\| = \lim_{k \rightarrow \infty} \alpha_k \|d_k\|$ since $z_k = x_k + \alpha_k d_k$ and $0 < \gamma < 2$ from the HTTCGP. This completes the proof. □

Remark 3 Notice that the proof of the above lemma relies heavily on the projection technique and the line search technique independent of the specific structure of the search direction. It should be mentioned that although the whole sequence $\{\|x_k - x^*\|\}$ is convergent from the above lemma, this does not guarantee that $\{x_k\}$ converges to $x^* \in \mathcal{S}$, and even it is not necessarily convergent.

We now combine these results to prove the global convergence of the HTTCGP under some mild conditions.

Theorem 1 *Suppose that the assumptions (A1)-(A2) hold. Then the whole sequence $\{x_k\}$ generated by the HTTCGP converges to a solution of proplem (1).*

Proof We start our proof by noticing from (16) and (17) that $0 \leq \alpha \|d_k\| \leq \alpha_k \|d_k\| \rightarrow 0$. This shows $\lim_{k \rightarrow \infty} \|d_k\| = 0$. Combining this with (15), we have:

$$0 \leq \left(1 - \frac{(1 + \bar{t})^2}{4}\right) \|F_k\| \leq \|d_k\| \rightarrow 0,$$

which implies $\lim_{k \rightarrow \infty} \|F_k\| = 0$. Since the sequence $\{x_k\} \subset \mathcal{C}$ is bounded from the fact that $x_0 \in \mathcal{C}$, (10) and Lemma 4, there exists at least one cluster point of $\{x_k\}$. Suppose that \bar{x} is a cluster point of the sequence $\{x_k\} \subset \mathcal{C}$ and let $\mathcal{K} \subset \{0, 1, 2, \dots\}$ be an infinite index set such that:

$$\lim_{k \rightarrow \infty, k \in \mathcal{K}} x_k = \bar{x} \in \mathcal{C},$$

where the inclusion follows from the closedness of \mathcal{C} . Thus, it follows from these and the continuity of F that:

$$0 = \lim_{k \rightarrow \infty} \|F_k\| = \lim_{k \rightarrow \infty, k \in \mathcal{K}} \|F_k\| = \|F(\bar{x})\|,$$

which implies that \bar{x} is a solution of problem (1), i.e., $\bar{x} \in \mathcal{S}$. Therefore, setting $x^* := \bar{x}$ in Lemma 4, we know that the sequence $\{\|x_k - \bar{x}\|\}$ is convergent and then:

$$\lim_{k \rightarrow \infty} \|x_k - \bar{x}\| = \lim_{k \rightarrow \infty, k \in \mathcal{K}} \|x_k - \bar{x}\| = 0.$$

Consequently, we conclude that the whole sequence $\{x_k\}$ converges to $\bar{x} \in \mathcal{S}$. This completes the proof. □

4 Numerical experiments

In this section, we perform numerical experiments to illustrate the effectiveness and robustness of the proposed method (HTTCGP), in comparison with two existing algorithms—Algorithm 2.1 in [30] (denoted by ATTCGP) and Algorithm 3.1 in [34] (denoted by HCGP)—in terms of the total number of iterations (Itr), the number of function evaluations (NF), and the CPU time in second (Tcpu) in the same calculating environment. All codes are written in MATLAB R2007b and run on a 64-bit Dell PC with Intel(R) Core(TM) i7-9700 CPU (3.00 GHz), 16.00 GB RAM and Windows 10.0 operating system.

The parameters in the HTTCGP are chosen as follows: $\rho = 0.5$, $\zeta = 1$, $\sigma = 0.01$, $\lambda = 0.001$, $\nu = 0.8$, $\mu = 0.2$, $\gamma = 1.6$, $\bar{t} = 0.3$ with t_k being calculated by:

$$t_k = \min \left\{ \bar{t}, \max \left\{ 0, 1 - \frac{y_{k-1}^\top s_{k-1}}{\|y_{k-1}\|^2} \right\} \right\}.$$

The parameters of ATTCGP are taken from Section 5 of [30], i.e., $\rho = 0.4$, $\zeta = 1$, $\sigma = 0.0001$, $r = 0.1$, $\mu_1 = 0.01$, $\mu_2 = 0.1$, $t_k = t_k^b$. The parameters of HCGP are taken from Section 4 of [34], i.e., $\rho = 0.5$, $\zeta = 1$, $\sigma = 0.0001$, $\mu = 1.4$, $\gamma = 1.5$. For each algorithm and each test problem, we stop the program when one of the following three conditions is satisfied:

- (i) $\|F_k\| \leq 10^{-6}$, (ii) $\|d_k\| \leq 10^{-7}$, or (iii) Itr > 2000.

Although Lemma 3 shows that the adaptive line search (9) terminates finitely, we impose a restriction on the steplength computation. That is, for some integer $i > 0$, if the trial steplength $\zeta\rho^i \leq 10^{-10}$ does not satisfy (9), then we set $\alpha_k = \zeta\rho^i$. We now give the following popular test problems in which the mapping F is taken as:

$$F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T.$$

Problem 4.1 [50] Set $f_i(x) = e^{x_i} - 1$, for $i = 1, 2, \dots, n$ and $\mathcal{C} = \mathbb{R}_+^n$. Clearly, this problem has a unique solution $x^* = (0, 0, \dots, 0)^T$.

Problem 4.2 [7] Set

$$\begin{aligned} f_1(x) &= x_1 - e^{\cos(\frac{x_1+x_2}{n+1})}, \\ f_i(x) &= x_i - e^{\cos(\frac{x_{i-1}+x_i+x_{i+1}}{n+1})}, \text{ for } i = 2, 3, \dots, n-1, \\ f_n(x) &= x_n - e^{\cos(\frac{x_{n-1}+x_n}{n+1})}, \end{aligned}$$

and $\mathcal{C} = \mathbb{R}_+^n$.

Problem 4.3 [5] Set $f_i(x) = x_i - \sin|x_i - 1|$, for $i = 1, 2, \dots, n$ and $\mathcal{C} = \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i \leq n, x_i \geq -1, i = 1, 2, \dots, n\}$. Clearly, Problem 4.3 is nonsmooth at $x = (1, \dots, 1)^T$.

Problem 4.4 [28] Set

$$\begin{aligned} f_1(x) &= 2x_1 + \sin x_1 - 1, \\ f_i(x) &= 2x_{i-1} + 2x_i + \sin x_i - 1, \text{ for } i = 2, 3, \dots, n-1, \\ f_n(x) &= 2x_n + \sin x_n - 1, \end{aligned}$$

and $\mathcal{C} = \mathbb{R}_+^n$.

Problem 4.5 [28] Set

$$\begin{aligned} f_1(x) &= x_1 - e^{\cos(\frac{x_1+x_2}{2})}, \\ f_i(x) &= x_i - e^{\cos(\frac{x_{i-1}+x_i+x_{i+1}}{i})}, \text{ for } i = 2, 3, \dots, n-1, \\ f_n(x) &= x_n - e^{\cos(\frac{x_{n-1}+x_n}{n})}, \end{aligned}$$

and $\mathcal{C} = \mathbb{R}_+^n$.

Problem 4.6 [28] Set $f_i(x) = (e^{x_i})^2 + 3 \sin x_i \cdot \cos x_i - 1$, for $i = 1, 2, \dots, n$ and $\mathcal{C} = \mathbb{R}_+^n$.

Problem 4.7 This problem is Problem 2 in [51] with additional $\mathcal{C} = \mathbb{R}_+^n$, i.e., $f_i(x) = 2x_i - \sin x_i$, for $i = 1, 2, \dots, n$ and $\mathcal{C} = \mathbb{R}_+^n$.

Problem 4.8 This problem is Problem 10 in [52] with additional $\mathcal{C} = \mathbb{R}_+^n$, i.e., $f_i(x) = \ln(x_i + 1) - \frac{x_i}{n}$, for $i = 1, 2, \dots, n$ and $\mathcal{C} = \mathbb{R}_+^n$.

For Problem 4.3, we use the quadratic program solver `quadprog.m` from the MATLAB optimization toolbox to compute a new iterate x_{k+1} in step 4 of HTTCGP. For the above test problems, we use the following different initial points: $x_1 = (1, 1, \dots, 1)^T$, $x_2 = (\frac{1}{3}, \frac{1}{3^2}, \dots, \frac{1}{3^n})^T$, $x_3 = (\frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^n})^T$, $x_4 = (0, \frac{1}{n}, \dots, \frac{n-1}{n})^T$, $x_5 = (1, \frac{1}{2}, \dots, \frac{1}{n})^T$, $x_6 = (\frac{1}{n}, \frac{2}{n}, \dots, 1)^T$, $x_7 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0)^T$. The numerical results are listed in Tables 1, 2, 3, 4, 5, 6, 7, and 8, where “Init” means the initial point, “n” is the problem dimension multiplied by 10^4 , and “ $\|F^*$ ” stands for the final value of $\|F(x_k)\|$ when the program is stopped.

In order to compare the performance of these methods clearly, we adopt the performance profiles introduced by Dolan and Moré [53] based on Tcpu, NF, and Itr, respectively. From Tables 1–8 and Figs. 1, 2, and 3, we conclude that for these fixed initial points, the proposed method in general is the most effective and robust in terms of Tcpu, NF, and Itr, respectively. The reason may lie in twofold, one is that the search direction defined by (7) inherits the effectiveness and robustness of the memoryless BFGS method, and the other is that the hybrid technique may numerically effectively improve the performance of the three-term conjugate gradient projection method.

Table 1 Numerical results on Problem 4.1

Init(n)	HTTCGP	HCGP	ATTCGP
	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $
$x_1(1)$	1/4/0.012/0.00e+00	1/4/0.007/0.00e+00	35/106/0.034/6.90e−07
$x_2(1)$	2/6/0.003/2.22e−16	9/28/0.008/6.21e−07	25/76/0.018/7.76e−07
$x_3(1)$	7/19/0.005/4.92e−07	9/28/0.007/7.77e−07	26/79/0.019/6.83e−07
$x_4(1)$	17/48/0.014/5.55e−07	12/37/0.009/9.98e−07	34/103/0.026/8.53e−07
$x_5(1)$	9/24/0.007/4.42e−07	10/31/0.008/5.69e−07	27/82/0.019/6.76e−07
$x_6(1)$	12/32/0.009/7.01e−09	12/37/0.009/9.98e−07	34/103/0.026/8.53e−07
$x_7(1)$	17/48/0.014/5.55e−07	12/37/0.009/9.98e−07	34/103/0.026/8.53e−07
$x_1(5)$	1/4/0.006/0.00e+00	1/4/0.006/0.00e+00	36/109/0.133/9.25e−07
$x_2(5)$	2/6/0.007/2.22e−16	9/28/0.028/6.21e−07	25/76/0.074/7.76e−07
$x_3(5)$	7/19/0.023/4.92e−07	9/28/0.030/7.77e−07	26/79/0.079/6.83e−07
$x_4(5)$	16/44/0.059/2.18e−07	13/40/0.048/5.58e−07	36/109/0.131/6.86e−07
$x_5(5)$	9/24/0.031/4.39e−07	10/31/0.035/5.69e−07	27/82/0.089/6.76e−07
$x_6(5)$	14/38/0.053/7.49e−07	13/40/0.048/5.58e−07	36/109/0.130/6.86e−07
$x_7(5)$	16/44/0.061/2.18e−07	13/40/0.047/5.58e−07	36/109/0.131/6.86e−07
$x_1(10)$	1/4/0.013/0.00e+00	1/4/0.012/0.00e+00	37/112/0.354/7.85e−07
$x_2(10)$	2/6/0.018/2.22e−16	9/28/0.074/6.21e−07	25/76/0.201/7.76e−07
$x_3(10)$	7/19/0.060/4.92e−07	9/28/0.075/7.77e−07	26/79/0.211/6.83e−07
$x_4(10)$	16/43/0.157/1.33e−15	13/40/0.124/7.89e−07	36/109/0.341/9.71e−07
$x_5(10)$	9/24/0.083/4.39e−07	10/31/0.089/5.69e−07	27/82/0.237/6.76e−07
$x_6(10)$	15/40/0.146/3.45e−07	13/40/0.124/7.89e−07	36/109/0.342/9.71e−07
$x_7(10)$	16/43/0.156/1.33e−15	13/40/0.124/7.89e−07	36/109/0.343/9.71e−07

Table 2 Numerical results on Problem 4.2

Init(<i>n</i>)	HTTCGP	HCGP	ATTCGP
	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $
$x_1(1)$	15/39/0.029/2.51e−07	14/43/0.028/6.40e−07	38/115/0.075/6.38e−07
$x_2(1)$	15/39/0.027/3.97e−07	15/46/0.030/2.53e−07	39/118/0.076/6.06e−07
$x_3(1)$	15/39/0.027/3.97e−07	15/46/0.030/2.53e−07	39/118/0.076/6.06e−07
$x_4(1)$	15/39/0.027/3.26e−07	14/43/0.028/8.33e−07	38/115/0.075/8.31e−07
$x_5(1)$	15/39/0.027/3.97e−07	15/46/0.030/2.53e−07	39/118/0.076/6.05e−07
$x_6(1)$	15/39/0.027/3.26e−07	14/43/0.028/8.33e−07	38/115/0.074/8.31e−07
$x_7(1)$	15/39/0.027/3.26e−07	14/43/0.028/8.33e−07	38/115/0.074/8.31e−07
$x_1(5)$	15/39/0.135/5.61e−07	15/46/0.146/3.58e−07	39/118/0.376/8.56e−07
$x_2(5)$	15/39/0.135/8.87e−07	15/46/0.147/5.66e−07	40/121/0.384/8.13e−07
$x_3(5)$	15/39/0.135/8.87e−07	15/46/0.146/5.66e−07	40/121/0.385/8.13e−07
$x_4(5)$	15/39/0.135/7.30e−07	15/46/0.146/4.66e−07	40/121/0.383/6.69e−07
$x_5(5)$	15/39/0.134/8.87e−07	15/46/0.146/5.66e−07	40/121/0.385/8.12e−07
$x_6(5)$	15/39/0.134/7.30e−07	15/46/0.145/4.66e−07	40/121/0.383/6.69e−07
$x_7(5)$	15/39/0.135/7.30e−07	15/46/0.147/4.66e−07	40/121/0.385/6.69e−07
$x_1(10)$	15/39/0.308/7.93e−07	15/46/0.326/5.06e−07	40/121/0.860/7.26e−07
$x_2(10)$	16/41/0.322/4.32e−07	15/46/0.325/8.01e−07	41/124/0.882/6.89e−07
$x_3(10)$	16/41/0.322/4.32e−07	15/46/0.325/8.01e−07	41/124/0.883/6.89e−07
$x_4(10)$	16/41/0.323/3.55e−07	15/46/0.325/6.59e−07	40/121/0.860/9.46e−07
$x_5(10)$	16/41/0.322/4.32e−07	15/46/0.324/8.01e−07	41/124/0.881/6.89e−07
$x_6(10)$	16/41/0.323/3.55e−07	15/46/0.326/6.59e−07	40/121/0.860/9.46e−07
$x_7(10)$	16/41/0.324/3.55e−07	15/46/0.325/6.59e−07	40/121/0.861/9.46e−07

To further explore the influence of initial points on the algorithm, we test each problem by using the initial points randomly generated from the interval (0,1) with a gradually increasing number of dimensions. For the given dimension, we run the same test 20 times to obtain the average CPU time in second (\overline{Tcpu}), the average number of function evaluations (\overline{NF}), the average number of iterations (\overline{Itr}), and the average value of $\|F^*\|$ ($\overline{\|F^*\|}$). The numerical results are listed in Table 9, where “*P*” stands for the test problems used and “*n*” is still the problem dimension multiplied by 10^4 . We also plot the performance profiles of \overline{Tcpu} , \overline{NF} , and \overline{Itr} , respectively. It is not hard to see from Table 9 and Figs. 4, 5, and 6 that (i) all the methods can solve all the test problems successfully; (ii) our method outperforms the other two methods about 68%, 72%, and 70% of the experiments in terms of \overline{Tcpu} , \overline{NF} , and \overline{Itr} , respectively. In most of the experiments, our method needs less number of function evaluations to obtain an approximate solution compared to the other two methods. The reason may be that the former benefits from the adaptive line search technique proposed by Yin et al. [45]. In a word, the proposed method is more efficient and competitive than the other two methods for the given test problems.

Table 3 Numerical results on Problem 4.3

Init(n)	HTTCGP	HCGP	ATTCGP
	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $
$x_1(1)$	8/25/0.006/2.83e-07	7/22/0.004/4.50e-07	14/43/0.008/3.10e-07
$x_2(1)$	14/50/0.009/4.36e-07	14/46/0.008/2.23e-08	14/43/0.008/4.90e-07
$x_3(1)$	16/58/0.010/1.55e-07	15/49/0.008/6.53e-07	14/43/0.008/4.90e-07
$x_4(1)$	21/79/0.014/3.10e-07	17/57/0.010/2.83e-07	13/40/0.007/9.40e-07
$x_5(1)$	15/54/0.010/2.08e-07	17/59/0.010/2.40e-07	14/43/0.008/4.89e-07
$x_6(1)$	20/75/0.013/6.29e-07	20/67/0.011/5.01e-07	13/40/0.007/9.40e-07
$x_7(1)$	21/79/0.014/3.10e-07	17/57/0.010/2.83e-07	13/40/0.007/9.40e-07
$x_1(5)$	8/25/0.024/6.32e-07	8/25/0.020/1.70e-09	14/43/0.035/6.94e-07
$x_2(5)$	11/38/0.033/2.78e-07	18/61/0.048/9.43e-07	15/46/0.037/2.75e-07
$x_3(5)$	16/56/0.048/6.84e-07	14/46/0.036/1.12e-07	15/46/0.037/2.75e-07
$x_4(5)$	20/71/0.061/8.47e-07	20/67/0.052/5.15e-07	14/43/0.035/5.28e-07
$x_5(5)$	13/45/0.041/4.00e-07	15/51/0.040/7.65e-07	15/46/0.038/2.75e-07
$x_6(5)$	21/79/0.066/3.38e-07	18/61/0.047/2.55e-07	14/43/0.034/5.28e-07
$x_7(5)$	20/71/0.061/8.47e-07	20/67/0.052/5.15e-07	14/43/0.035/5.28e-07
$x_1(10)$	8/25/0.065/8.94e-07	8/25/0.056/2.41e-09	14/43/0.097/9.81e-07
$x_2(10)$	16/55/0.134/3.71e-07	13/42/0.093/3.94e-07	15/46/0.105/3.89e-07
$x_3(10)$	14/46/0.114/3.50e-07	17/56/0.122/8.24e-09	15/46/0.104/3.89e-07
$x_4(10)$	23/85/0.200/2.84e-07	20/68/0.146/2.48e-08	14/43/0.098/7.46e-07
$x_5(10)$	15/51/0.125/9.24e-07	15/50/0.109/1.47e-08	15/46/0.107/3.89e-07
$x_6(10)$	21/78/0.183/7.88e-07	18/61/0.132/8.89e-08	14/43/0.097/7.46e-07
$x_7(10)$	23/85/0.200/2.84e-07	20/68/0.145/2.48e-08	14/43/0.098/7.46e-07

5 Applications in compressed sensing

In this section, we apply our method to solve compressed sensing problems and compare it with two existing solvers: CGD [9] and ATTCGP [30].

5.1 Problem description

In compressed sensing problems, the task is to find/recover sparse solutions/signals of underdetermined linear systems $Ax = b$, where $A \in \mathbb{R}^{m \times n}$ ($m \ll n$) is a sampling matrix, and $b \in \mathbb{R}^m$ is the observation signal. To this end, a popular approach is to solve the ℓ_1 -norm regularized optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \|Ax - b\|^2 + \tau \|x\|_1, \quad (20)$$

where $\tau > 0$ is a constant balancing the data-fitting and sparsity. Notice that the objective function $f(x)$ in (20) is nonsmooth convex. Interestingly, one can reformulate (20) as a convex quadratic programming by Figueiredo, Nowak, and Wright [54].

Table 4 Numerical results on Problem 4.4

Init(n)	HTTCGP	HCGP	ATTCGP
	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $
$x_1(1)$	35/155/0.043/9.05e−07	39/179/0.045/6.14e−07	19/77/0.020/6.54e−07
$x_2(1)$	41/178/0.048/9.64e−07	40/183/0.046/6.09e−07	18/73/0.019/9.49e−07
$x_3(1)$	42/181/0.048/8.29e−07	38/174/0.044/9.60e−07	18/73/0.019/9.82e−07
$x_4(1)$	29/132/0.035/9.06e−07	42/192/0.048/7.01e−07	20/81/0.021/5.70e−07
$x_5(1)$	36/160/0.042/8.28e−07	40/183/0.046/8.04e−07	19/77/0.020/8.48e−07
$x_6(1)$	29/132/0.035/8.11e−07	42/192/0.048/7.01e−07	20/81/0.021/5.70e−07
$x_7(1)$	30/133/0.035/7.70e−07	41/187/0.047/6.06e−07	19/77/0.020/4.57e−07
$x_1(5)$	34/151/0.197/5.33e−07	39/179/0.214/5.87e−07	19/77/0.096/7.93e−07
$x_2(5)$	36/160/0.206/8.56e−07	42/191/0.228/6.07e−07	18/73/0.091/8.85e−07
$x_3(5)$	37/163/0.210/6.42e−07	39/178/0.213/9.27e−07	19/77/0.096/4.77e−07
$x_4(5)$	32/143/0.184/7.82e−07	44/200/0.238/6.69e−07	19/77/0.096/8.04e−07
$x_5(5)$	34/152/0.195/7.89e−07	42/191/0.228/7.65e−07	19/77/0.096/8.34e−07
$x_6(5)$	36/158/0.203/5.18e−07	44/200/0.240/6.69e−07	19/77/0.097/8.04e−07
$x_7(5)$	24/111/0.142/7.97e−07	39/179/0.213/6.08e−07	19/77/0.096/5.36e−07
$x_1(10)$	33/148/0.481/6.67e−07	41/187/0.563/5.87e−07	19/77/0.242/8.80e−07
$x_2(10)$	39/170/0.564/6.01e−07	41/187/0.562/5.89e−07	18/73/0.229/9.19e−07
$x_3(10)$	43/185/0.609/6.24e−07	39/178/0.533/9.00e−07	19/77/0.240/4.93e−07
$x_4(10)$	30/135/0.437/8.71e−07	44/200/0.599/6.54e−07	19/77/0.240/7.76e−07
$x_5(10)$	33/149/0.484/6.34e−07	41/187/0.560/7.51e−07	19/77/0.240/8.67e−07
$x_6(10)$	30/135/0.438/9.04e−07	44/200/0.602/6.54e−07	19/77/0.240/7.76e−07
$x_7(10)$	27/124/0.399/9.79e−07	39/179/0.538/5.91e−07	19/77/0.244/5.97e−07

Here we provide the concrete process for completeness. By introducing the auxiliary variables $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$, we can reformulate x as:

$$x = u - v, \quad u \geq 0, \quad v \geq 0,$$

where $u_i = \max\{x_i, 0\}$ and $v_i = \max\{-x_i, 0\}$ for all $i = 1, 2, \dots, n$. Then, $\|x\|_1 = e_n^T u + e_n^T v$, where $e_n = (1, 1, \dots, 1)^T \in \mathbb{R}^n$. Mathematically, the ℓ_1 -norm regularized optimization problem (20) can be equivalently expressed as:

$$\begin{aligned} \min_{u,v} \quad & \frac{1}{2} \|A(u - v) - b\|^2 + \tau e_n^T u + \tau e_n^T v, \\ \text{s.t.} \quad & u^T v = 0, \quad u \geq 0, \quad v \geq 0. \end{aligned}$$

However, due to the special structure of the objective function above as analyzed in [54], we know that the above optimization problem is equivalent to:

$$\begin{aligned} \min_{u,v} \quad & \frac{1}{2} \|A(u - v) - b\|^2 + \tau e_n^T u + \tau e_n^T v, \\ \text{s.t.} \quad & u \geq 0, \quad v \geq 0. \end{aligned} \tag{21}$$

Table 5 Numerical results on Problem 4.5

Init(n)	HTTCGP	HCGP	ATTCGP
	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $
$x_1(1)$	20/65/0.044/5.42e-08	28/98/0.061/7.40e-07	26/79/0.051/5.73e-07
$x_2(1)$	27/83/0.055/5.92e-07	31/108/0.068/4.50e-07	27/82/0.053/6.56e-07
$x_3(1)$	26/81/0.053/9.51e-07	31/108/0.068/4.50e-07	26/79/0.051/9.83e-07
$x_4(1)$	19/62/0.040/2.36e-07	34/117/0.073/4.69e-07	28/85/0.055/5.45e-07
$x_5(1)$	25/79/0.052/4.04e-07	31/108/0.067/4.50e-07	26/79/0.051/6.53e-07
$x_6(1)$	19/61/0.040/5.84e-07	34/117/0.073/4.69e-07	27/82/0.053/5.92e-07
$x_7(1)$	23/73/0.048/7.65e-07	35/120/0.075/4.89e-07	27/82/0.053/3.47e-07
$x_1(5)$	22/79/0.248/3.04e-07	31/108/0.331/4.04e-07	28/85/0.268/9.13e-07
$x_2(5)$	22/73/0.233/1.75e-07	31/108/0.332/3.48e-07	28/85/0.270/6.05e-07
$x_3(5)$	23/74/0.240/8.18e-07	31/108/0.329/3.48e-07	31/94/0.299/4.43e-07
$x_4(5)$	25/91/0.283/3.14e-07	31/108/0.330/3.60e-07	29/88/0.280/3.04e-07
$x_5(5)$	23/77/0.245/4.91e-07	31/108/0.330/3.48e-07	28/85/0.271/7.01e-07
$x_6(5)$	22/80/0.250/6.79e-07	31/108/0.332/3.60e-07	29/88/0.280/3.04e-07
$x_7(5)$	30/94/0.306/3.28e-07	31/108/0.330/3.66e-07	28/85/0.270/2.73e-07
$x_1(10)$	21/67/0.489/4.40e-07	31/108/0.734/3.58e-07	29/88/0.628/3.59e-07
$x_2(10)$	24/80/0.575/2.56e-07	33/114/0.778/4.69e-07	28/85/0.603/9.75e-07
$x_3(10)$	21/71/0.508/9.25e-07	33/114/0.778/4.69e-07	29/88/0.626/7.21e-07
$x_4(10)$	19/63/0.454/7.36e-07	33/114/0.777/4.70e-07	29/88/0.625/3.19e-07
$x_5(10)$	22/73/0.524/8.93e-07	33/114/0.777/4.69e-07	28/85/0.607/6.09e-07
$x_6(10)$	19/63/0.454/7.16e-07	33/114/0.779/4.70e-07	29/88/0.626/3.19e-07
$x_7(10)$	21/68/0.494/8.69e-07	33/114/0.778/4.94e-07	30/91/0.649/2.42e-07

Then by expanding the quadratic term $\|A(u - v) - b\|^2$ in (21) and rearranging terms in the resulting relation, (21) can be written as the following simple bound constrained quadratic programming:

$$\begin{aligned} \min_{z \in \mathbb{R}^{2n}} \quad & \frac{1}{2} z^T H z + c^T z, \\ \text{s.t.} \quad & z \geq 0, \end{aligned} \tag{22}$$

where:

$$z = \begin{bmatrix} u \\ v \end{bmatrix}, \quad c = \begin{bmatrix} \tau e_n - A^T b \\ \tau e_n + A^T b \end{bmatrix}, \quad H = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}.$$

It is easy to show that H is positive semi-definite. Then, (22) is a convex quadratic programming. Hence, it follows from the first-order optimality conditions of (22) that z is a minimizer of (22) if and only if z is a solution of the following unconstrained equations:

$$F(z) = \min\{z, H z + c\} = 0,$$

where the “min” is interpreted as componentwise minimum. We then have from [55, Lemma 3] and [2, Lemma 2.2] that $F : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$ is Lipschitz continuous

Table 6 Numerical results on Problem 4.6

Init(n)	HTTCGP	HCGP	ATTCGP
	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $
$x_1(1)$	1/6/0.005/0.00e+00	1/6/0.004/0.00e+00	1/4/0.003/0.00e+00
$x_2(1)$	1/6/0.003/2.22e-16	1/6/0.003/2.22e-16	9/37/0.015/7.62e-07
$x_3(1)$	1/6/0.003/3.14e-16	1/6/0.003/3.14e-16	10/41/0.018/3.85e-07
$x_4(1)$	12/59/0.024/0.00e+00	10/51/0.022/1.56e-07	14/58/0.021/8.70e-07
$x_5(1)$	10/49/0.019/3.55e-08	10/46/0.018/1.54e-07	13/52/0.018/3.82e-07
$x_6(1)$	12/59/0.024/0.00e+00	10/51/0.021/1.49e-07	14/58/0.021/8.73e-07
$x_7(1)$	12/59/0.024/0.00e+00	10/51/0.021/1.56e-07	14/58/0.021/8.70e-07
$x_1(5)$	1/6/0.018/0.00e+00	1/6/0.017/0.00e+00	1/4/0.014/0.00e+00
$x_2(5)$	1/6/0.011/2.22e-16	1/6/0.010/2.22e-16	9/37/0.063/7.62e-07
$x_3(5)$	1/6/0.011/3.14e-16	1/6/0.010/3.14e-16	10/41/0.071/3.85e-07
$x_4(5)$	12/59/0.117/0.00e+00	10/51/0.104/3.42e-07	15/62/0.111/3.90e-07
$x_5(5)$	10/49/0.090/2.97e-08	10/46/0.089/4.68e-07	13/52/0.091/3.82e-07
$x_6(5)$	12/59/0.118/0.00e+00	10/51/0.104/3.39e-07	15/62/0.111/3.90e-07
$x_7(5)$	12/59/0.117/0.00e+00	10/51/0.104/3.42e-07	15/62/0.111/3.90e-07
$x_1(10)$	1/6/0.037/0.00e+00	1/6/0.036/0.00e+00	1/4/0.029/0.00e+00
$x_2(10)$	1/6/0.023/2.22e-16	1/6/0.022/2.22e-16	9/37/0.147/7.62e-07
$x_3(10)$	1/6/0.023/3.14e-16	1/6/0.022/3.14e-16	10/41/0.163/3.85e-07
$x_4(10)$	12/59/0.271/6.00e-15	10/51/0.236/4.83e-07	15/62/0.260/5.51e-07
$x_5(10)$	10/49/0.211/2.89e-08	10/46/0.204/6.10e-07	13/52/0.211/3.82e-07
$x_6(10)$	12/59/0.272/5.62e-15	10/51/0.236/4.81e-07	15/62/0.259/5.52e-07
$x_7(10)$	12/59/0.273/6.00e-15	10/51/0.237/4.83e-07	15/62/0.260/5.51e-07

and monotone. Thus, we can apply the proposed method to solve the unconstrained equations above.

5.2 Numerical results

In this subsection, we perform two types of numerical experiments to verify the effectiveness and efficiency of the proposed method.

In our first experiment, the main goal is to restore a one-dimensional sparse signal of length n from m observations, where $m \ll n$. For fairness, the parameters for our method HTTCGP are taken as the same as those in Section 4 except for $\mu = 2$; the parameters of CGD are taken from Section 5.2 of [9]; the parameters of ATTCGP are chosen from Section 6.2 of [30] with $t_k = t_k^b$. As usual, the quality of recovery is measured by the mean of squared error (MSE, for succinctness):

$$\text{MSE} = \frac{1}{n} \|\bar{x} - x\|^2,$$

where \bar{x} is the restored signal and x is the original sparse one. From the definition of MSE, the capability of achieving a lower MSE value reflects a better quality of

Table 7 Numerical results on Problem 4.7

Init(n)	HTTCGP	HCGP	ATTCGP
	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $
$x_1(1)$	2/6/0.002/0.00e+00	13/40/0.007/7.74e-07	36/109/0.018/8.79e-07
$x_2(1)$	2/6/0.002/1.48e-323	10/31/0.008/3.19e-07	25/76/0.018/9.88e-07
$x_3(1)$	2/6/0.002/9.88e-324	10/31/0.009/4.94e-07	26/79/0.022/9.52e-07
$x_4(1)$	7/18/0.004/2.59e-07	13/40/0.007/6.16e-07	35/106/0.017/8.99e-07
$x_5(1)$	3/8/0.002/0.00e+00	10/31/0.005/8.74e-07	28/85/0.014/7.08e-07
$x_6(1)$	7/18/0.004/2.61e-07	13/40/0.007/6.16e-07	35/106/0.017/8.99e-07
$x_7(1)$	7/18/0.004/2.59e-07	13/40/0.007/6.16e-07	35/106/0.017/8.99e-07
$x_1(5)$	2/6/0.007/0.00e+00	14/43/0.033/4.33e-07	38/115/0.086/7.07e-07
$x_2(5)$	2/6/0.007/1.48e-323	10/31/0.026/3.19e-07	25/76/0.062/9.88e-07
$x_3(5)$	2/6/0.007/9.88e-324	10/31/0.028/4.94e-07	26/79/0.069/9.52e-07
$x_4(5)$	7/18/0.018/5.81e-07	14/43/0.033/3.44e-07	37/112/0.084/7.24e-07
$x_5(5)$	3/8/0.008/0.00e+00	10/31/0.024/8.74e-07	28/85/0.064/7.08e-07
$x_6(5)$	7/18/0.018/5.82e-07	14/43/0.032/3.44e-07	37/112/0.084/7.24e-07
$x_7(5)$	7/18/0.018/5.81e-07	14/43/0.032/3.44e-07	37/112/0.083/7.24e-07
$x_1(10)$	2/6/0.016/0.00e+00	14/43/0.096/6.12e-07	39/118/0.263/6.00e-07
$x_2(10)$	2/6/0.016/1.48e-323	10/31/0.070/3.19e-07	25/76/0.173/9.88e-07
$x_3(10)$	2/6/0.016/9.88e-324	10/31/0.073/4.94e-07	26/79/0.184/9.52e-07
$x_4(10)$	7/18/0.050/8.22e-07	14/43/0.095/4.87e-07	38/115/0.257/6.14e-07
$x_5(10)$	3/8/0.022/0.00e+00	10/31/0.068/8.74e-07	28/85/0.189/7.08e-07
$x_6(10)$	7/18/0.050/8.22e-07	14/43/0.095/4.87e-07	38/115/0.257/6.14e-07
$x_7(10)$	7/18/0.050/8.22e-07	14/43/0.095/4.87e-07	38/115/0.256/6.14e-07

the restored signal for one method. Therefore, the lower the asymptotically optimal MSE value is, the better the method is. All the algorithms start their iterations with $x_0 = A^\top b$, and all take the stopping criterion as:

$$\frac{|f(x_k) - f(x_{k-1})|}{|f(x_{k-1})|} \leq 10^{-5},$$

where $f(x)$ is defined in (20). The parameter τ defined in (20) for three algorithms is obtained by the same continuation technique used in [54].

In our experiments, let r denote the number of nonzero entries of x . For a given triple (m, n, r) , we use the following MATLAB codes to generate the test data:

```

x=zeros(n,1); % initialize the original sparse signal
q=randperm(n);
x(q(1:r))=randn(r,1);
A=randn(m,n);
A = orth(A')'; % orthonormalize the rows of A
b=A* x+0.001* randn(m,1);
    
```

Table 8 Numerical results on Problem 4.8

Init(n)	HTTCGP	HCGP	ATTCGP
	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $	Itr/NF/Tcpu/ $\ F^*\ $
$x_1(1)$	1/3/0.003/0.00e+00	1/3/0.001/0.00e+00	5/11/0.006/3.62e-09
$x_2(1)$	3/7/0.003/2.22e-16	5/11/0.005/1.23e-14	7/15/0.007/6.85e-08
$x_3(1)$	5/11/0.006/3.79e-13	7/15/0.007/2.49e-15	7/15/0.007/8.49e-07
$x_4(1)$	5/11/0.006/0.00e+00	7/15/0.008/0.00e+00	8/17/0.008/1.91e-07
$x_5(1)$	3/7/0.004/0.00e+00	6/13/0.006/0.00e+00	8/17/0.008/7.58e-09
$x_6(1)$	5/11/0.006/0.00e+00	7/15/0.008/0.00e+00	8/17/0.008/1.91e-07
$x_7(1)$	5/11/0.006/0.00e+00	7/15/0.008/0.00e+00	8/17/0.008/1.91e-07
$x_1(5)$	1/3/0.007/0.00e+00	1/3/0.007/0.00e+00	13/36/0.068/6.69e-07
$x_2(5)$	3/7/0.013/2.22e-16	5/11/0.020/1.23e-14	16/44/0.070/8.86e-07
$x_3(5)$	5/11/0.022/1.04e-13	6/13/0.024/1.22e-15	15/40/0.066/8.75e-07
$x_4(5)$	5/11/0.028/0.00e+00	7/15/0.036/0.00e+00	24/67/0.119/8.27e-07
$x_5(5)$	3/7/0.019/0.00e+00	6/13/0.029/0.00e+00	24/67/0.130/6.28e-07
$x_6(5)$	5/11/0.029/0.00e+00	7/15/0.036/0.00e+00	24/67/0.120/8.28e-07
$x_7(5)$	5/11/0.028/0.00e+00	7/15/0.037/0.00e+00	24/67/0.120/8.27e-07
$x_1(10)$	1/3/0.015/0.00e+00	1/3/0.014/0.00e+00	13/36/0.162/9.21e-07
$x_2(10)$	3/7/0.032/2.22e-16	5/11/0.047/1.23e-14	16/44/0.171/9.20e-07
$x_3(10)$	5/11/0.052/2.37e-14	6/13/0.057/1.22e-15	15/40/0.158/9.69e-07
$x_4(10)$	5/11/0.066/0.00e+00	7/15/0.085/0.00e+00	25/70/0.301/7.45e-07
$x_5(10)$	3/7/0.043/0.00e+00	6/13/0.067/0.00e+00	24/67/0.310/7.53e-07
$x_6(10)$	5/11/0.066/0.00e+00	7/15/0.085/0.00e+00	25/70/0.301/7.45e-07
$x_7(10)$	5/11/0.067/0.00e+00	7/15/0.086/0.00e+00	25/70/0.303/7.45e-07

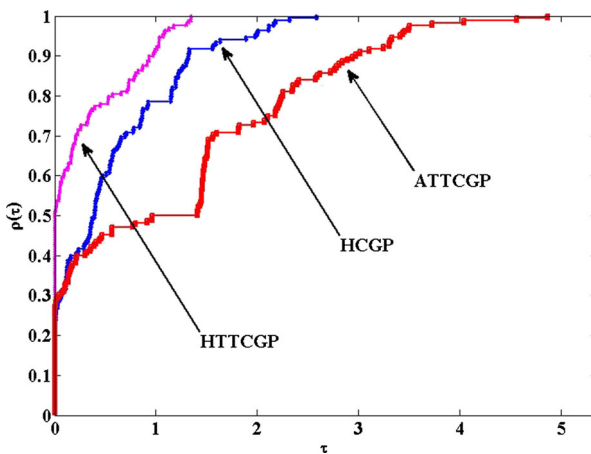


Fig. 1 Performance profiles on Tcpu

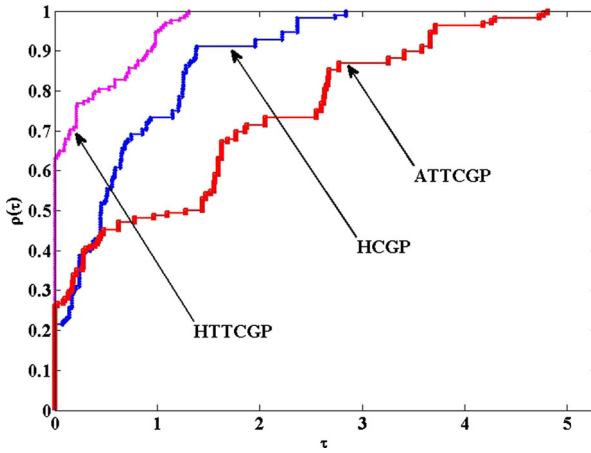


Fig. 2 Performance profiles on NF

From Fig. 7, we observe that the original signal is almost exactly restored from the disturbed one by the above-mentioned three algorithms, while our algorithm outperforms the other two algorithms in terms of MSE, Itr, and Tcpu. To further demonstrate the performance of our algorithm, we set $(m, n, r) = (256i, 1024i, 32i)$ with $i = 1 : 0.5 : 8$. For each $i = 1 : 0.5 : 8$, we generate 20 random instances as described above. The detailed numerical results are reported in Table 10, where we report the average number of iterations (\bar{Itr}), the average CPU time in second (\bar{Tcpu}), and the average MSE (\bar{MSE}) over the 20 instances. We see from Table 10 that HTTCGP clearly outperforms the other two algorithms in terms of \bar{Itr} , \bar{Tcpu} and \bar{MSE} . Moreover, ATTCGP is always faster than CGD, and its solution quality is also better than CGD in terms of MSE. In summary, the reported numerical

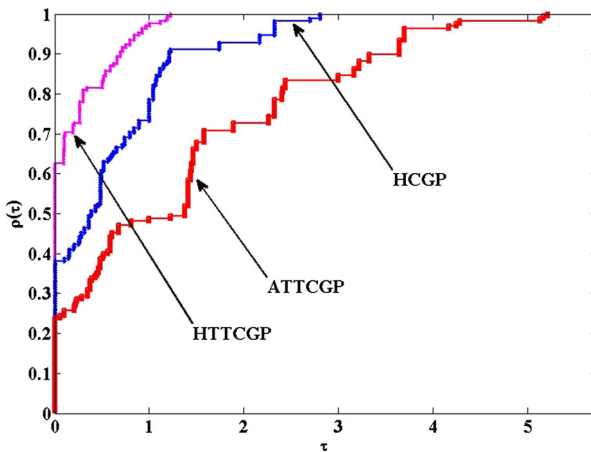


Fig. 3 Performance profiles on Itr

Table 9 Numerical results on Problems 4.1–4.8 with the randomly initial points

$P(n)$	HTTCGP	HCGP	ATTCGP
	$\overline{\text{Itr}}/\overline{\text{NF}}/\overline{\text{Tcpu}}/\ F^*\ $	$\overline{\text{Itr}}/\overline{\text{NF}}/\overline{\text{Tcpu}}/\ F^*\ $	$\overline{\text{Itr}}/\overline{\text{NF}}/\overline{\text{Tcpu}}/\ F^*\ $
4.1(1)	10/27/0.009/2.47e−07	12/38/0.010/8.46e−07	34/103/0.027/8.54e−07
4.1(3)	10/27/0.024/5.13e−07	13/40/0.030/4.32e−07	35/106/0.080/8.86e−07
4.1(5)	10/26/0.037/3.20e−07	13/40/0.050/5.57e−07	36/109/0.136/6.86e−07
4.1(8)	10/26/0.082/3.01e−07	13/40/0.103/7.05e−07	36/109/0.285/8.68e−07
4.1(10)	11/28/0.109/3.33e−07	13/40/0.129/7.89e−07	36/109/0.356/9.71e−07
4.2(1)	15/39/0.028/3.26e−07	14/43/0.028/8.33e−07	38/115/0.075/8.31e−07
4.2(3)	15/39/0.081/5.66e−07	15/46/0.089/3.61e−07	39/118/0.227/8.64e−07
4.2(5)	15/39/0.135/7.30e−07	15/46/0.147/4.66e−07	40/121/0.388/6.68e−07
4.2(8)	15/39/0.247/9.24e−07	15/46/0.264/5.89e−07	40/121/0.697/8.46e−07
4.2(10)	16/41/0.326/3.55e−07	15/46/0.329/6.59e−07	40/121/0.871/9.46e−07
4.3(1)	20/73/0.014/4.33e−07	18/62/0.011/5.80e−07	13/40/0.008/9.40e−07
4.3(3)	21/75/0.041/5.94e−07	18/62/0.032/5.11e−07	14/43/0.023/4.09e−07
4.3(5)	21/76/0.069/5.54e−07	19/63/0.054/5.08e−07	14/43/0.038/5.28e−07
4.3(8)	22/80/0.161/5.47e−07	18/61/0.112/3.54e−07	14/43/0.083/6.68e−07
4.3(10)	22/81/0.206/4.28e−07	18/63/0.145/4.31e−07	14/43/0.104/7.47e−07
4.4(1)	57/248/0.067/7.61e−07	60/274/0.070/6.24e−07	76/288/0.077/9.21e−07
4.4(3)	59/256/0.198/8.09e−07	63/285/0.208/6.04e−07	80/304/0.232/9.00e−07
4.4(5)	60/261/0.334/7.78e−07	63/283/0.341/7.87e−07	82/311/0.393/9.27e−07
4.4(8)	61/263/0.697/8.38e−07	63/285/0.692/9.62e−07	84/316/0.812/9.61e−07
4.4(10)	63/270/0.899/8.04e−07	65/293/0.891/6.26e−07	86/323/1.039/8.82e−07
4.5(1)	22/72/0.047/5.74e−07	34/117/0.074/4.72e−07	27/81/0.053/4.76e−07
4.5(3)	27/84/0.165/6.50e−07	31/108/0.201/3.91e−07	27/82/0.158/6.50e−07
4.5(5)	23/83/0.260/5.48e−07	31/108/0.333/3.60e−07	28/85/0.271/5.52e−07
4.5(8)	23/73/0.433/5.04e−07	31/108/0.596/3.47e−07	28/85/0.490/5.45e−07
4.5(10)	23/75/0.550/5.76e−07	33/114/0.786/4.71e−07	28/86/0.620/4.75e−07
4.6(1)	13/63/0.026/3.33e−07	11/54/0.023/2.25e−07	14/58/0.022/8.79e−07
4.6(3)	12/61/0.075/2.10e−07	10/51/0.065/3.72e−07	15/62/0.070/3.03e−07
4.6(5)	13/62/0.127/1.14e−07	10/51/0.108/3.48e−07	15/62/0.115/3.88e−07
4.6(8)	12/60/0.229/1.26e−07	10/51/0.195/4.55e−07	15/62/0.215/4.94e−07
4.6(10)	12/61/0.293/7.50e−08	10/51/0.244/5.00e−07	15/62/0.270/5.55e−07
4.7(1)	7/18/0.004/2.54e−07	13/40/0.007/6.16e−07	35/106/0.019/8.99e−07
4.7(3)	7/18/0.011/4.53e−07	14/43/0.022/2.67e−07	36/109/0.055/9.35e−07
4.7(5)	7/18/0.018/5.69e−07	14/43/0.036/3.45e−07	37/112/0.092/7.24e−07
4.7(8)	7/18/0.042/7.37e−07	14/43/0.082/4.36e−07	37/112/0.215/9.16e−07
4.7(10)	7/18/0.053/8.19e−07	14/43/0.102/4.87e−07	38/115/0.276/6.14e−07
4.8(1)	5/11/0.006/0.00e+00	7/15/0.008/0.00e+00	8/17/0.009/1.89e−07
4.8(3)	5/11/0.017/0.00e+00	7/15/0.022/0.00e+00	23/64/0.074/9.85e−07
4.8(5)	5/11/0.028/0.00e+00	7/15/0.037/0.00e+00	24/67/0.127/8.27e−07
4.8(8)	5/11/0.053/0.00e+00	7/15/0.069/0.00e+00	25/70/0.254/6.56e−07
4.8(10)	5/11/0.067/0.00e+00	7/15/0.087/0.00e+00	25/70/0.319/7.46e−07

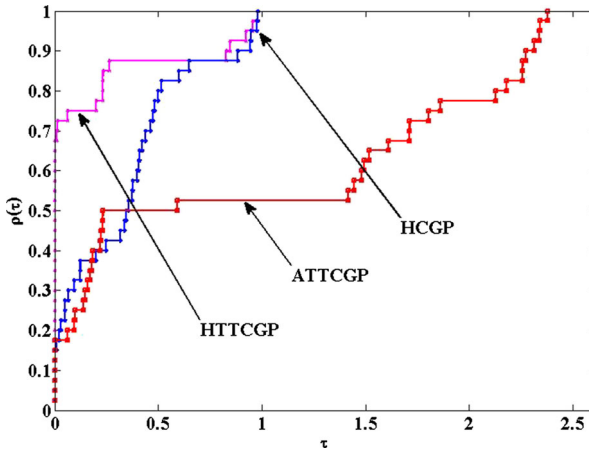


Fig. 4 Performance profiles on \overline{T}_{cpu}

results indicate that the proposed method is promising for recovering a sparse signal in compressive sensing.

In our second experiment, we concentrate on the image de-blurring problem and apply the proposed method to restore the original image from its blurred image. We refer the readers to the monograph [56] for the background of the digital image restoration problem. In the literature, the quality of restoration is usually measured by the peak signal-to-noise ratio (PSNR):

$$\text{PSNR} = 10 \times \log_{10} \frac{V^2}{\text{MSE}}$$

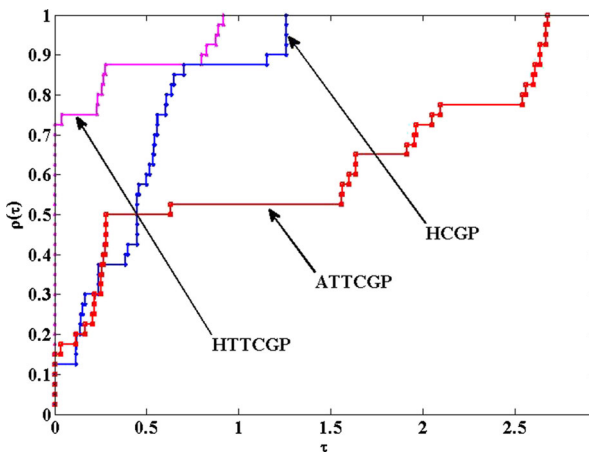


Fig. 5 Performance profiles on \overline{NF}

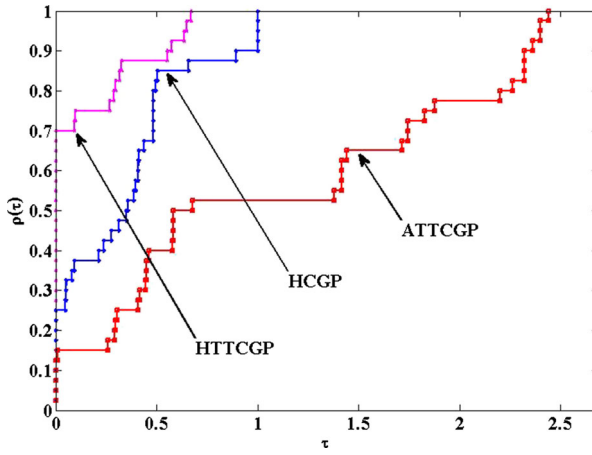


Fig. 6 Performance profiles on $\overline{\text{Itr}}$

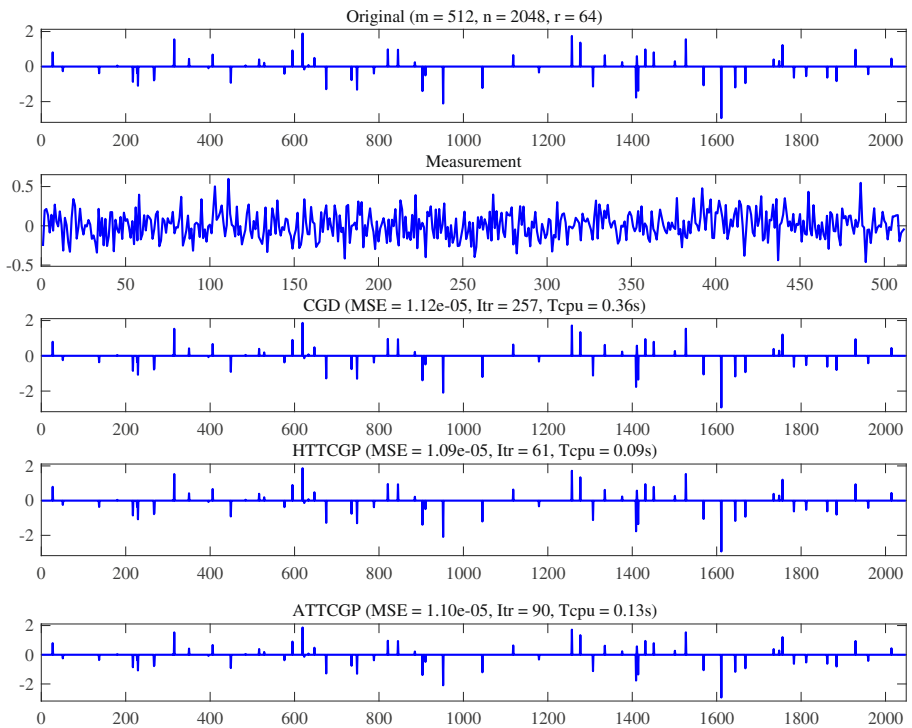


Fig. 7 From top to bottom: the original signal, the measurement, and the recovered signals by three algorithms

and the structured similarity (SSIM) index which measures the similarity between the original image and the restored one; see [57] for more details. Here V is the maximum absolute value of the reconstruction. The MATLAB implementation of the SSIM index can be obtained at <http://www.cns.nyu.edu/~lcv/ssim/>. The larger the PSNR and SSIM are, the better the corresponding method is. The initialization and stopping criterion of all three algorithms are exactly the same as the first experiment above. The parameters for HTTCGP are chosen from the first experiment. For CGD and ATTCGP, we empirically determine appropriate values for their involved parameters. The tuned parameters are as follows: $\rho = 0.1$ for CGD; $\zeta = 0.8$ for ATTCGP and the other parameters are taken from their original papers. Here, the matrix A defined in (20) is a partial discrete Walsh-Hadamard transform (DWT) matrix. We refer the readers to [2] and references therein for a discussion about the computational cost associated with matrix A .

The tested images are Chart.tiff (256×256), Barbara.png (512×512), Lena.png (512×512) and Man.bmp (1024×1024). See Fig. 8 for the original, blurred, and restored images by three algorithms. We report the corresponding results in Table 11. According to Table 11, we know that our method is attractive and efficient in terms of the solution quality. Although the proposed method requires more number of iterations, we observe that its CPU time is usually less than CGD. In a word, all the numerical results show that our method provides a effective approach for solving compressed sensing problems.

Table 10 Numerical results for sparse signal restoration

Problem size (m, n, r)	CGD $\overline{\text{Itr}}/\overline{\text{Tcpu}}/\overline{\text{MSE}}$	HTTCGP $\overline{\text{Itr}}/\overline{\text{Tcpu}}/\overline{\text{MSE}}$	ATTCGP $\overline{\text{Itr}}/\overline{\text{Tcpu}}/\overline{\text{MSE}}$
(256,1024,32)	229/0.10/1.313e−05	67/0.03/6.915e−06	92/0.03/7.007e−06
(384,1536,48)	244/0.18/1.137e−05	64/0.04/6.854e−06	92/0.05/6.993e−06
(512,2048,64)	234/0.31/1.132e−05	61/0.07/7.737e−06	89/0.11/7.829e−06
(640,2560,80)	249/0.60/9.893e−06	62/0.14/8.355e−06	89/0.20/8.683e−06
(768,3072,96)	246/0.86/1.482e−05	63/0.21/1.081e−05	92/0.30/1.081e−05
(896,3584,112)	236/1.15/1.835e−05	63/0.30/9.409e−06	88/0.41/9.609e−06
(1024,4096,128)	241/1.64/1.448e−05	62/0.41/9.971e−06	88/0.57/1.021e−05
(1152,4608,144)	249/1.98/1.078e−05	62/0.48/9.833e−06	88/0.67/9.833e−06
(1280,5120,160)	241/2.37/2.629e−05	64/0.62/1.005e−05	89/0.84/1.056e−05
(1408,5632,176)	244/2.86/1.708e−05	63/0.73/9.862e−06	87/0.99/9.923e−06
(1536,6144,192)	254/4.18/1.147e−05	59/0.97/9.633e−06	90/1.45/9.702e−06
(1664,6656,208)	250/4.06/1.470e−05	60/0.97/9.969e−06	89/1.41/9.698e−06
(1792,7168,224)	256/4.95/1.498e−05	61/1.17/1.124e−05	90/1.70/1.153e−05
(1920,7680,240)	249/5.25/2.763e−05	59/1.24/9.961e−06	87/1.80/1.007e−05
(2048,8192,256)	250/7.37/1.642e−05	62/1.83/1.099e−05	87/2.51/1.113e−05



Fig. 8 The original images (first row), the blurred images (second row), and the restored images by CGD (third row), HTTCGP (fourth row), and ATTCGP (last row)

Table 11 Numerical results for the image de-blurring problem

Image	CGD	HTTCGP	ATTCGP
	Itr/Tcpu/PSNR/SSIM	Itr/Tcpu/PSNR/SSIM	Itr/Tcpu/PSNR/SSIM
Chart	19/2.39/20.77/0.84	72/2.95/24.95/0.94	13/0.59/20.27/0.82
Barbara	21/11.72/23.98/0.86	51/9.28/24.77/0.90	23/4.03/24.37/0.87
Lena	21/11.41/30.33/0.95	32/5.83/32.00/0.97	15/2.59/30.99/0.96
Man	18/42.05/28.53/0.98	36/30.27/30.74/0.99	35/28.94/30.26/0.99

6 Conclusions

In this paper, we propose a hybrid three-term conjugate gradient projection method for solving large-scale constrained nonlinear monotone equations, in which the search direction is close to that of the memoryless BFGS method in the sense of the univariate minimum problem and has the sufficient descent property and the trust region property independent of line search technique. With the help of the adaptive line search technique, the global convergence of the proposed method is established under some mild conditions. A large number of numerical experiments about the constrained nonlinear monotone equations indicate that our method not only inherits the good properties of the three-term conjugate gradient methods and the hybrid conjugate gradient methods but also benefits from the adaptive line search technique, thus greatly improves its computational efficiency compared to the two existing algorithms. Moreover, we apply the proposed method to solve sparse signal restoration and image de-blurring problems, and compare it with two existing methods. The numerical results show that our method is practical and promising.

Acknowledgments The authors wish to thank the three anonymous referees for their very professional comments and quite useful suggestions, which greatly helped us to improve the original version of this paper.

Funding This work was supported by the National Natural Science Foundation of China (11771383), the Natural Science Foundation of Guangxi Province (2018GXNSFAA281099), the Research Project of Guangxi University for Nationalities (2018KJQD02), and the Middle-aged and Young Teachers' Basic Ability Promotion Project of Guangxi Province (2018KY0700) of China.

References

1. Zhao, Y.B., Li, D.: Monotonicity of fixed point and normal mapping associated with variational inequality and its applications. *SIAM J. Optim.* **11**(4), 962–973 (2001)
2. Xiao, Y.H., Wang, Q.Y., Hu, Q.J.: Non-smooth equations based method for ℓ_1 -norm problems with applications to compressed sensing. *Nonlinear Anal.* **74**(11), 3570–3577 (2011)
3. Dirkse, S.P., Ferris, M.C.: MCPLIB: A collection of nonlinear mixed complementarity problems. *Optim. Methods Softw.* **5**(4), 319–345 (1995)
4. Wood, A.J., Wollenberg, B.F.: *Power Generation, Operation, and Control*. Wiley, New York (1996)
5. Yu, Z.S., Lin, J., Sun, J., Xiao, Y.H., Liu, L.Y., Li, Z.H.: Spectral gradient projection method for monotone nonlinear equations with convex constraints. *Appl. Numer. Math.* **59**(10), 2416–2423 (2009)

6. Liu, J., Duan, Y.R.: Two spectral gradient projection methods for constrained equations and their linear convergence rate. *J. Inequal. Appl.* **2015**(1), 1–13 (2015)
7. 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**(1), 117–129 (2013)
8. Zhang, L.: A modified PRP projection method for nonlinear equations with convex constraints. *Int. J. Pure Appl. Math.* **79**(1), 87–96 (2012)
9. 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**(1), 310–319 (2013)
10. Liu, S.Y., Huang, Y.Y., Jiao, H.W.: Sufficient descent conjugate gradient methods for solving convex constrained nonlinear monotone equations. *Abstr. Appl. Anal.* **2014**, 1–12 (2014)
11. Sun, M., Liu, J.: A modified Hestenes-Stiefel projection method for constrained nonlinear equations and its linear convergence rate. *J. Appl. Math. Comput.* **49**(1-2), 145–156 (2015)
12. Ding, Y.Y., Xiao, Y.H., Li, J.W.: A class of conjugate gradient methods for convex constrained monotone equations. *Optimization* **66**(12), 2309–2328 (2017)
13. Liu, J.K., Feng, Y.M.: A derivative-free iterative method for nonlinear monotone equations with convex constraints. *Numer. Algorithms* **82**, 245–262 (2019)
14. Abubakar, A.B., Kumam, P.: A descent Dai-Liao conjugate gradient method for nonlinear equations. *Numer. Algorithms* **81**(1), 197–210 (2019)
15. Ibrahim, A.H., Kumam, P., Abubakar, A.B., Jirakitpuwapat, W., Abubakar, J.: A hybrid conjugate gradient algorithm for constrained monotone equations with application in compressive sensing. *Heliyon* **6**(3), e03466 (2020)
16. Wang, S., Guan, H.B.: A scaled conjugate gradient method for solving monotone nonlinear equations with convex constraints. *J. Appl. Math.* **2013**(1), 1–7 (2013)
17. Liu, J.K., Li, S.J.: Multivariate spectral DY-type projection method for convex constrained nonlinear monotone equations. *J. Ind. Manag. Optim.* **13**(1), 283–295 (2017)
18. Abubakar, A.B., Kumam, P., Mohammad, H., Awwal, A.M., Sitthithakerngkiet, K.: A modified Fletcher-Reeves conjugate gradient method for monotone nonlinear equations with some applications. *Mathematics* **7**(8), 745 (2019)
19. Abubakar, A.B., Kumam, P., Mohammad, H., Awwal, A.M.: An efficient conjugate gradient method for convex constrained monotone nonlinear equations with applications. *Mathematics* **7**(9), 767 (2019)
20. Solodov, M.V., Svaiter, B.F.: A globally convergent inexact newton method for systems of monotone equations. In: Fukushima, M., Qi, L. (eds.) *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pp.355–369. Kluwer Academic (1998)
21. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. *IMA J. Numer. Anal.* **8**(1), 141–148 (1988)
22. Zhang, L., Zhou, W.J.: Spectral gradient projection method for solving nonlinear monotone equations. *J. Comput. Appl. Math.* **196**, 478–484 (2006)
23. Abubakar, A.B., Kumam, P., Mohammad, H.: A note on the spectral gradient projection method for nonlinear monotone equations with applications. *Comput. Appl. Math.* **39**, 129 (2020)
24. Awwal, A.M., Wang, L., Kumam, P., Mohammad, H.: A two-step spectral gradient projection method for system of nonlinear monotone equations and image deblurring problems. *Symmetry* **12**(6), 874 (2020)
25. Mohammad, H., Abubakar, A.B.: A descent derivative-free algorithm for nonlinear monotone equations with convex constraints. *RAIRO-oper Res.* **54**(2), 489–505 (2020)
26. Wang, X.Y., Li, S.J., Kou, X.P.: A self-adaptive three-term conjugate gradient method for monotone nonlinear equations with convex constraints. *Calcolo* **53**(2), 133–145 (2016)
27. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **49**, 409–436 (1952)
28. Gao, P.T., He, C.J.: An efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints. *Calcolo* **55**(4), 53 (2018)
29. Liu, Y., Storey, C.: Efficient generalized conjugate gradient algorithms, part 1: Theory. *J. Optim. Theory Appl.* **69**, 177–182 (1991)
30. Gao, P.T., He, C.J., Liu, Y.: An adaptive family of projection methods for constrained monotone nonlinear equations with applications. *Appl. Math. Comput.* **359**, 1–16 (2019)
31. Oren, S.S., Luenberger, D.G.: Self-scaling variable metric (SSVM) algorithms, Part i: Criteria and sufficient conditions for scaling a class of algorithms. *Manag. Sci.* **20**(5), 845–862 (1974)

32. Awwal, A.M., Kumam, P., Mohammad, H., Wathayu, W., Abubakar, A.B.: A Perry-type derivative-free algorithm for solving nonlinear system of equations and minimizing ℓ_1 regularized problem. Optimization. (2020). <https://doi.org/10.1080/02331934.2020.1808647>
33. Jian, J.B., Han, L., Jiang, X.Z.: A hybrid conjugate gradient method with descent property for unconstrained optimization. Appl. Math. Model. **39**(3-4), 1281–1290 (2015)
34. Sun, M., Liu, J.: New hybrid conjugate gradient projection method for the convex constrained equations. Calcolo **53**, 399–411 (2016)
35. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. **35**(151), 773–782 (1980)
36. Shanno, D.F.: Conjugate gradient methods with inexact searches. Math. Oper. Res. **3**, 244–256 (1978)
37. Li, M.: A modified Hestense-Stiefel conjugate gradient method close to the memoryless BFGS quasi-Newton method. Optim. Methods Softw. **33**(2), 336–353 (2018)
38. Li, M.: A three term polak-ribière-polyak conjugate gradient method close to the memoryless BFGS quasi-Newton method. J. Ind. Manag. Optim. **13**(5), 1–16 (2017)
39. Li, M.: A family of three-term nonlinear conjugate gradient methods close to the memoryless BFGS method. Optim. Lett. **12**, 1911–1927 (2018)
40. Beale, E.M.L.: A derivation of conjugate gradients. In: Lootsma, F.A. (ed.) Numerical Methods for Nonlinear Optimization. Academic Press, London (1972)
41. Nazareth, L.: A conjugate direction algorithm without line search. J. Optim. Theory Appl. **23**(3), 373–387 (1997)
42. Zhang, L., Zhou, W.J., Li, D.H.: A descent modified polak-ribière-polyak conjugate gradient method and its global convergence. IMA J. Numer. Anal. **26**(4), 629–640 (2006)
43. Dai, Y.H., Yuan, Y.X.: Nonlinear Conjugate Gradient Methods. Shanghai Science and Technology Publisher, Shanghai (2000)
44. Hager, W.W., Zhang, H.C.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**(1), 35–58 (2006)
45. Yin, J.H., Jian, J.B., Jiang, X.Z.: A spectral gradient projection algorithm for convex constrained nonsmooth equations based on an adaptive line search. Math. Numer. Sin. (Chinese) **42**(4), 457–471 (2020)
46. Li, Q., Li, D.H.: A class of derivative-free methods for large-scale nonlinear monotone equations. IMA J. Numer. Anal. **31**(4), 1625–1635 (2011)
47. Sun, M., Tian, M.Y.: A class of derivative-free CG projection methods for nonsmooth equations with an application to the LASSO problem. Bull. Iran. Math. Soc. **46**, 183–205 (2020)
48. Cai, X.J., Gu, G.Y., He, B.S.: On the $O(1/t)$ convergence rate of the projection and contraction methods for variational inequalities with Lipschitz continuous monotone operators. Comput. Optim. Appl. **57**(2), 339–363 (2014)
49. Zarantonello, E.H.: Projections on Convex Sets in Hilbert Space and Spectral Theory. Academic Press, New York (1971)
50. Wang, C.W., Wang, Y.J., Xu, C.L.: A projection method for a system of nonlinear monotone equations with convex constraints. Math. Meth. Oper. Res. **66**(1), 33–46 (2007)
51. Zhou, W.J., Li, D.H.: Limited memory BFGS method for nonlinear monotone equations. J. Comput. Math. **25**, 89–96 (2007)
52. La Cruz, W., Raydan, M.: Nonmonotone spectral methods for large-scale nonlinear systems. Optim. Methods Softw. **18**(5), 583–599 (2003)
53. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
54. Figueiredo, M.A.T., Nowak, R.D., Wright, S.J.: Gradient projection for sparse reconstruction, application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Sign. Proces. **1**(4), 586–597 (2008)
55. Pang, J.S.: Inexact Newton methods for the nonlinear complementary problem. Math. Program. **36**(1), 54–71 (1986)
56. Gonzales, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice Hall (2008)
57. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)