Malaya Journal of Matematik, Vol. 8, No. 2, 502-509, 2020

https://doi.org/10.26637/MJM0802/0031

A derivative-free conjugate gradient projection method based on the memoryless BFGS update

M. Koorapetse[1] and P. Kaelo[1]*

Abstract

Conjugate gradient-based projection methods are widely used for solving large-scale nonlinear monotone equations. This is due to their simplicity and that they are derivative-free. In this paper, we propose another conjugate gradient-based projection method for large-scale nonlinear monotone equations. We show that the method satisfies the descent condition independent of line searches and that the method is globally convergent. Numerical results show that the method is both efficient and effective.

Keywords

Global convergence, Nonlinear monotone equations, Derivative-free.

AMS Subject Classification

90C06, 90C56, 65K05, 65K10.

[1] *Department of Mathematics, University of Botswana, Private Bag UB00704, Gaborone, Botswana.*

***Corresponding author**: kaelop@mopipi.ub.bw;

Article History: Received **16** September **2019**; Accepted **21** March **2020**

Contents

1. Introduction

Consider the constrained nonlinear monotone equations

$$F(x) = 0, \quad x \in \Omega, \tag{1.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and satisfies the monotonicity condition

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

and $\Omega \subseteq \mathbb{R}^n$ is a nonempty closed convex set.

Nonlinear monotone equations arise in many applications such as subproblems in the generalized proximal algorithms with Bregman distances [7]. Some monotone variational inequality problems can also be converted into systems of nonlinear monotone equations by means of fixed point maps or normal maps if the underlying function satisfies some coercive conditions [21].

The study of iterative methods for solving Problem (1.1) with $\Omega = \mathbb{R}^n$ has received much attention. For instance, Solodov and Svaiter [15], proposed an inexact Newton method which is a combination of Newton method and hyperplane projection strategy. By the monotonicity of F, for any x^* such that $F(x^*) = 0$, we have

$$F(z_k)^T (x^* - z_k) \leq 0,$$

where $z_k = x_k + \alpha_k d_k$, x_k is the current iterate, α_k is the step length and d_k is the search direction. Thus, by performing some kind of line search procedure along the direction d_k, a point z_k can be computed such that

$$F(z_k)^T (x_k - z_k) > 0.$$

The above two inequalities indicate that the hyperplane

$$H_k = \{x \in \mathbb{R}^n \mid F(z_k)^T(x - z_k) = 0\}$$

strictly separates the current iterate x_k from the solution set of Problem (1.1). Using the hyperplane H_k, the next iterate is obtained by

$$x_{k+1} = x_k - \frac{F(z_k)^T(x_k - z_k)}{\|F(z_k)\|^2} F(z_k), \tag{1.3}$$

which is a projection of x_k onto H_k.

Conjugate gradient-based projection methods [1, 2, 4–6, 8, 9, 11, 12, 17–20, 22] are probably the most popular methods for solving nonlinear monotone equations (1.1). These methods are motivated by the hyperplane projection method in [15]. Recently, Ou and Li [14] presented a new derivative-free SCG-type projection method for nonlinear monotone equations with convex constraints in which

$$d_k = \begin{cases} -F_k, & \text{if } k = 0, \\ -\tilde{Q}_k F_k, & \text{if } k \geq 1, \end{cases}$$

where the matrix $\tilde{Q}_k \in \mathbb{R}^{n\times n}$ is defined by

$$\tilde{Q}_k = \tilde{\theta}_k I - \tilde{\theta}_k \frac{w_k s_k^T + s_k w_k^T}{w_k^T s_k} + \left(1 + \tilde{\theta}_k \frac{w_k^T w_k}{w_k^T s_k}\right) \frac{s_k s_k^T}{w_k^T s_k},$$

with

$$\tilde{\theta}_k = \frac{\|s_k\|^2}{w_k^T s_k},$$

$F_k = F(x_k)$, $s_k = x_k - x_{k-1}$ and $w_k = F_k - F_{k-1} + t s_k$, where $t > 0$ is a constant. The next iterate x_{k+1} in [14] is computed by projecting x_k onto the hyperplane H_k and then onto the feasible set Ω as

$$x_{k+1} = P_\Omega\left[x_k - \frac{F(z_k)^T(x_k - z_k)}{\|F(z_k)\|^2} F(z_k)\right], \tag{1.4}$$

where $P_\Omega[x] : \mathbb{R}^n \to \Omega$ is a projection operator

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

which is nonexpansive, i.e.

$$\|P_\Omega[x] - P_\Omega[y]\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{1.5}$$

This method was shown to be globally convergent and efficient.

In this paper, we present a new derivative-free conjugate gradient-based projection method for solving convex constrained nonlinear monotone equations and perform some numerical experiments to test its efficiency and effectiveness. This proposed method is presented in the next section and the rest of this paper is organized as follows. In Section 3, we show that the proposed method satisfies the descent property and also establish its global convergence. We also show the method converges R-linearly in Section 4. Numerical results follow in Section 5 and conclusion in Section 6.

2. Motivation and the algorithm

The method we propose is motivated by the work of Livieris et al. [13], Stanimirović et al. [16] and Liu and Feng [10]. Livieris et al. [13] recently proposed a hybrid conjugate gradient method based on the memorylesss BFGS update for solving the unconstrained optimization problem

$$\min\{f(x) \mid x \in \mathbb{R}^n\}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. This is an iterative method that generates a sequence of points $\{x_k\}$, starting from an initial point $x_0 \in \mathbb{R}^n$, using the recurrence

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \ldots,$$

where $\alpha_k > 0$ is the stepsize obtained by some line search, and d_k is the search direction defined by

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -\left(1 + \beta_k^{HCG+} \frac{g_k^T d_{k-1}}{\|g_k\|^2}\right) g_k + \beta_k^{HCG+} d_{k-1}, & \text{if } k \geq 1, \end{cases}$$

where

$$\beta_k^{HCG+} = \lambda_k \beta_k^{DY} + (1 - \lambda_k)\beta_k^{HS+},$$

with

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}}, \quad \beta_k^{HS+} = \max\{\beta_k^{HS}, 0\},$$

and

$$\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}.$$

The parameter $\lambda_k \in [0, 1]$ is given by

$$\lambda_k = \frac{s_k^T g_{k-1}}{\|g_{k-1}\|^2} \left[\frac{s_k^T y_{k-1}}{\|s_k\|^2} - \frac{1}{\vartheta_k} \frac{\|y_{k-1}\|^2}{s_k^T y_{k-1}} - 1\right] + \left(\frac{1}{\vartheta_k} - 1\right) \frac{y_{k-1}^T g_{k-1}}{\|g_{k-1}\|^2},$$

where $s_k = x_k - x_{k-1}$, $y_{k-1} = g_k - g_{k-1}$ and $g_k = \nabla f(x_k)$ is the gradient of f at x_k. Two different parameters of ϑ_k are presented, $\vartheta_k = \max\{\theta_k^{OL}, 1\}$ and $\vartheta_k = \max\{\theta_k^{OS}, 1\}$, in order to give two methods $ADHCG1$ and $ADHCG2$ respectively, with

$$\theta_k^{OL} = \frac{s_k^T y_{k-1}}{\|s_k\|} \quad \text{and} \quad \theta_k^{OS} = \frac{\|y_{k-1}\|^2}{s_k^T y_{k-1}}. \tag{2.1}$$

These methods satisfy the sufficient descent property

$$d_k^T g_k \leq -\|g_k\|^2, \quad \forall k \geq 0.$$

The methods were shown to perform well numerically as compared to other methods in the literature and global convergence was established by means of the strong Wolfe line search technique.

Stanimirović et al. [16], on the other hand, suggested a hybridization

$$d_k = \begin{cases} -g_k, & \text{if } k=0,\\ -\left(1+\beta_k^{LSCD}\frac{g_k^T d_{k-1}}{\|g_k\|^2}\right)g_k + \beta_k^{LSCD} d_{k-1}, & \text{if } k\geq 1, \end{cases}$$

where

$$\beta_k^{LSCD} = \max\{0, \min\{\beta_k^{LS}, \beta_k^{CD}\}\},$$

with

$$\beta_k^{LS} = -\frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}} \quad \text{and} \quad \beta_k^{CD} = -\frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}}.$$

This method was shown to be efficient and convergent.

In another recent work, Liu and Feng [10] presented a derivative-free method for nonlinear monotone equations (1.1) with

$$d_k = \begin{cases} -F_k, & \text{if } k=0,\\ -\theta_k F_k + \beta_k^{PDY} d_{k-1}, & \text{if } k\geq 1, \end{cases}$$

where

$$\beta_k^{PDY} = \frac{\|F_k\|^2}{d_{k-1}u_{k-1}}, \ \theta_k = c - \frac{F_k^T d_{k-1}}{d_{k-1}^T u_{k-1}},$$

with $u_{k-1} = y_{k-1} + t_{k-1}d_{k-1}$, $y_{k-1} = F_k - F_{k-1}$, $t_{k-1} = 1 + \max\left\{0, -\frac{d_{k-1}^T y_{k-1}}{d_{k-1}^T d_{k-1}}\right\}$ and $c>0$ a constant. The global convergence of the method was established and its efficacy was tested against other competing methods.

Now, inspired by the work of Livieris et al. [13], Liu and Feng [10] and that of Stanimirović [16] , we define our proposed method as

$$d_k = \begin{cases} -F_k, & \text{if } k=0,\\ -\left(1+\beta_k\frac{F_k^T s_{k-1}}{\|F_k\|^2}\right)F_k + \beta_k s_{k-1}, & \text{if } k\geq 1, \end{cases} \tag{2.2}$$

where

$$\beta_k = \max\{\beta_k^{HCG+}, \beta_k^{LSCD}\}, \tag{2.3}$$

with

$$\beta_k^{HCG+} = \lambda_k \beta_k^{DY} + (1-\lambda_k)\beta_k^{HS+},$$

$$\beta_k^{DY} = \frac{\|F_k\|^2}{d_{k-1}^T w_k} \ \text{and} \ \beta_k^{HS+} = \max\{\beta_k^{HS}, 0\},$$

where

$$\beta_k^{HS} = \frac{F_k^T w_k}{d_{k-1}^T w_k},$$

and

$$\beta_k^{LSCD} = \max\{0, \min\{\beta_k^{LS}, \beta_k^{CD}\}\},$$

$$\beta_k^{LS} = -\frac{F_k^T w_k}{d_{k-1}^T F_{k-1}}, \ \text{and} \ \beta_k^{CD} = -\frac{\|F_k\|^2}{d_{k-1}^T F_{k-1}}.$$

The parameter $\lambda_k \in [0,1]$ is given by

$$\lambda_k = \frac{s_{k-1}^T F_{k-1}}{\|F_{k-1}\|^2}\left[\frac{s_{k-1}^T w_k}{\|s_{k-1}\|^2} - \frac{1}{\theta_k^M}\frac{\|w_k\|^2}{s_{k-1}^T w_k} - 1\right] + \left(\frac{1}{\theta_k^M} - 1\right)\frac{w_k^T F_{k-1}}{\|F_{k-1}\|^2},$$

where

$$\theta_k^M = c - \frac{F_k^T s_{k-1}}{s_{k-1}^T w_k}$$

with c being a positive constant. Here, $w_k = F(z_{k-1}) - F_{k-1} + r s_{k-1}$, $s_k = z_k - x_k = \alpha_k d_k$ and $r \in (0,1)$. We state the algorithm as follows.

Algorithm 2.1. *Memoryless BFGS Conjugate Gradient-based Method (MBCG)*

1: *Give* $x_0 \in \Omega$ *and the parameters* $\sigma, r, \rho \in (0,1)$. *Set* $k=0$.

2: ***for*** $k = 0,1,\ldots$ ***do***
3: *If* $\|F_k\| = 0$, *then stop. Otherwise, go to Step 4.*
4: *Compute* d_k *by (2.2) and (2.3).*
5: *Compute* $z_k = x_k + \alpha_k d_k$ *where* $\alpha_k = \max\{\rho^i : i = 0,1,2,\ldots\}$ *such that the inequality*

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha_k \|F(z_k)\| \|d_k\|^2 \tag{2.4}$$

is satisfied.
6: *If* $z \in \Omega$ *and* $\|F(z_k)\| = 0$, *then stop. Otherwise, compute* x_{k+1} *using (1.4).*
7: *Set* $k = k+1$ *and go to Step 3.*
8: ***end for***

3. Global convergence

In this section, we analyze the global convergence of Algorithm 2.1. For this purpose, we first make the following assumptions.

Assumption 3.1. **(i)** *The function* $F(\cdot)$ *is monotone on* $\mathbb{R}^n$, *i.e.* $(F(x) - F(y))^T(x-y) \geq 0$, $\forall x, y \in \mathbb{R}^n$.

(ii) *The solution set* Ω^* *is nonempty.*

(iii) *The function $F(\cdot)$ is Lipschitz continuous on $\mathbb{R}^n$, i.e. there exists a positive constant L such that*

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

Lemma 3.2. *Let the sequences $\{d_k\}$ and $\{F_k\}$ be generated by Algorithm 2.1. Then we have*

$$F_k^T d_k = -\|F_k\|^2, \quad \forall k \geq 0. \tag{3.2}$$

Proof. Since $d_0 = -F_0$, we have $F_0^T d_0 = -\|F_0\|^2$, which satisfies (3.2). For $k \geq 1$, by taking the inner product of (2.2) with the vector F_k, we have

$$\begin{aligned} F_k^T d_k &= -\left(1 + \beta_k \frac{F_k^T s_{k-1}}{\|F_k\|^2}\right) \|F_k\|^2 + \beta_k F_k^T s_{k-1} \\ &= -\|F_k\|^2. \end{aligned}$$

Thus (3.2) holds. □

Lemma 3.3. *Let $\{x_k\}$ and $\{z_k\}$ be generated by Algorithm 2.1. Then*

$$\alpha_k \geq \min\left\{1, \frac{\rho \|F_k\|^2}{(L + \sigma \|F(z_k')\|)\|d_k\|^2}\right\}, \tag{3.3}$$

where $z_k' = x_k + \rho^{-1}\alpha_k d_k$.

Lemma 3.4. *Suppose Assumption 3.1 holds and sequences $\{x_k\}$ and $\{z_k\}$ are generated by Algorithm 2.1. Then $\{x_k\}$ and $\{z_k\}$ are both bounded. Furthermore, it holds that*

$$\lim_{k\to\infty} \|x_k - z_k\| = 0. \tag{3.4}$$

Proof. From (2.4), we have

$$F(z_k)^T (x_k - z_k) \geq \sigma \|F(z_k)\| \|x_k - z_k\|^2 > 0. \tag{3.5}$$

For $x^* \in \Omega$ we have from (1.4) and (1.5) that

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|P_\Omega[x_k - \xi_k F(z_k)] - x^*\|^2 \\ &\leq \|x_k - \xi_k F(z_k) - x^*\|^2 \\ &= \|x_k - x^*\|^2 - 2\xi_k F(z_k)^T (x_k - x^*) \\ &\quad + \xi_k^2 \|F(z_k)\|^2, \end{aligned} \tag{3.6}$$

where $\xi_k = \frac{F(z_k)^T (x_k - z_k)}{\|F(z_k)\|^2}$. By the monotonicity of F, it follows that

$$\begin{aligned} F(z_k)^T (x_k - x^*) &= F(z_k)^T (x_k - z_k) + F(z_k)^T (z_k - x^*) \\ &\geq F(z_k)^T (x_k - z_k) + F(x^*)^T (z_k - x^*) \\ &= F(z_k)^T (x_k - z_k). \end{aligned} \tag{3.7}$$

From (3.5)-(3.7), we obtain

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &\leq \|x_k - x^*\|^2 - 2\xi_k F(z_k)^T (x_k - z_k) \\ &\quad + \xi_k^2 \|F(z_k)\|^2 \\ &= \|x_k - x^*\|^2 - \frac{(F(z_k)^T (x_k - z_k))^2}{\|F(z_k)\|^2} \\ &\leq \|x_k - x^*\|^2 - \sigma^2 \|x_k - z_k\|^4. \end{aligned} \tag{3.8}$$

Hence the sequence $\{x_k - x^*\}$ is decreasing and convergent, thus $\{x_k\}$ is bounded. From (3.5), we get

$$\begin{aligned} \sigma \|F(z_k)\| \|x_k - z_k\|^2 &\leq F(z_k)^T (x_k - z_k) \\ &\leq \|F(z_k)\| \|x_k - z_k\|, \end{aligned} \tag{3.9}$$

which shows that

$$\sigma \|x_k - z_k\| \leq 1,$$

indicating that $\{z_k\}$ is bounded. It then follows from (3.8) that

$$\sigma^2 \sum_{k=0}^{\infty} \|x_k - z_k\|^4 \leq \sum_{k=0}^{\infty} (\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2) < \infty,$$

which implies

$$\lim_{k\to\infty} \|x_k - z_k\| = 0.$$

□

Note that $\{x_k\}$ and $\{z_k\}$ bounded imply that there exist constants $M > 0$ and $M_0 > 0$ such that $\|s_k\| = \|\alpha_k d_k\| \leq M$, and that both $\|F_k\| \leq M_0$ and $\|F(z_k)\| \leq M_0$. That is, the sequences $\{s_k\}$ and $\{F_k\}$ are bounded.

Theorem 3.5. *Suppose that Assumption 3.1 holds, and the sequence $\{x_k\}$ is generated by Algorithm 2.1. Then*

$$\liminf_{k\to\infty} \|F_k\| = 0. \tag{3.10}$$

Proof. Suppose (3.10) does not hold. Then there is a constant $\varepsilon_0 > 0$ such that

$$\|F_k\| \geq \varepsilon_0, \ \forall k \geq 0.$$

By (3.2) we have that

$$\|d_k\| \geq \|F_k\| \geq \varepsilon_0, \ \forall k \geq 0.$$

By definition of w_k we have that there exist constants γ and M_1 such that

$$\|w_k\| \leq \gamma \ \text{ and } d_{k-1}^T w_k \geq M_1 \|d_{k-1}\|, \ \forall k \geq 0.$$

Now, if $\beta_k = \beta_k^{HCG+}$, we have that

$$\beta_k^{HCG+} \leq \frac{\|F_k\|^2 + \|F_k\| \|w_k\|}{d_{k-1}^T w_k} \tag{3.11}$$

$$\leq \frac{M_0(M_0 + \gamma)}{M_1 \|d_{k-1}\|}. \tag{3.12}$$

This gives that

$$\begin{aligned} \|d_k\| &\leq \|F_k\| + 2\beta_k^{HCG+} \alpha_{k-1} \|d_{k-1}\| \\ &\leq M_0 + 2\frac{M_0(M_0 + \gamma)}{M_1} = \gamma_1. \end{aligned} \tag{3.13}$$

On the other hand, if $\beta_k = \beta_k^{LSCD}$ we obtain that $\beta_k \le \beta_k^{CD}$. Hence

$$\begin{aligned}\|d_k\| &\le \|F_k\| + 2\beta_k^{CD}\|s_{k-1}\| \\ &\le M_0 + 2\frac{\|F_k\|^2}{\|F_{k-1}\|^2}\|s_{k-1}\| \\ &\le M_0 + \frac{2M_0^2}{\varepsilon_0^2}\alpha_{k-1}\|d_{k-1}\| \qquad (3.14)\end{aligned}$$

for all $k \ge 0$.

Since (3.4) holds, we obtain that for every $\varepsilon_1 > 0$ there is a k_0 such that $\alpha_{k-1}\|d_{k-1}\| < \varepsilon_1$ for all $k > k_0$. Now, choosing $\varepsilon_1 = \varepsilon_0^2$ and $\varpi = \max\{\gamma_1, \|d_0\|, \|d_1\|, \cdots, \|d_{k_0}\|, \gamma_2\}$, where $\gamma_2 = M_0 + 2M_0^2$, it holds that

$$\|d_k\| \le \varpi, \ \forall k \ge 0.$$

From (3.3) we have that

$$\begin{aligned}\alpha_k\|d_k\| &\ge \min\left\{1, \frac{\rho\|F_k\|^2}{(L+\sigma\|F(z_k')\|)\|d_k\|^2}\right\}\|d_k\| \\ &= \min\left\{\|d_k\|, \frac{\rho\|F_k\|^2}{(L+\sigma\|F(z_k')\|)\|d_k\|}\right\} \\ &\ge \min\left\{\varepsilon_0, \frac{\rho\varepsilon_0^2}{(L+\sigma M_0)\varpi}\right\} > 0.\end{aligned}$$

This contradicts (3.4), therefore (3.10) holds. □

4. R-linear convergence rate

In this section, we discuss the R-linear convergence rate for *Algorithm* 2.1. From Theorem 3.5, we know that the sequence $\{x_k\}$ converges to a solution of Problem (1.1). Thus, we always assume that $x_k \to x^*$ as $k \to \infty$, where $x^* \in \Omega^*$. To prove the R-linear convergence of $\{x_k\}$, we need the following assumption.

Assumption 4.1. *For any $x^* \in \Omega^*$, there exist $\mu \in (0,1)$ and $\delta > 0$ such that*

$$\mu dist(x,\Omega^*) \le \|F(x)\|^2, \quad \forall x \in \mathcal{N}_\delta(x^*), \qquad (4.1)$$

where $\mathcal{N}_\delta(x^*)$ is the neighbourhood of x^* defined by $\mathcal{N}_\delta(x^*) = \{x \in \mathbb{R}^n : \|x - x^*\| \le \delta\}$ and $dist(x,\Omega^*)$ denotes the distance from x to the solution set Ω^*.

Theorem 4.2. *Suppose that Assumptions 3.1 and 4.1 hold. Let the sequence $\{x_k\}$ be generated by Algorithm 2.1. Then the sequence $\{dist(x_k,\Omega^*)\}$ is Q-linearly convergent to 0, and so the sequence $\{x_k\}$ is R-linearly convergent to x^*.*

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

$$\|x_k - \bar{x}_k\| = dist(x_k,\Omega^*).$$

From (3.2), (3.8) and (4.1), for $\bar{x}_k \in \Omega^*$ we have

$$\begin{aligned}dist(x_{k+1},\Omega^*)^2 &= \|x_{k+1} - \bar{x}_k\|^2 \\ &\le dist(x_k,\Omega^*)^2 - \sigma^2\|\alpha_k d_k\|^4 \\ &\le dist(x_k,\Omega^*)^2 - \sigma^2\alpha_k^4\|F_k\|^4 \\ &\le dist(x_k,\Omega^*)^2 - \mu^2\sigma^2\alpha_k^4 dist(x_k,\Omega^*)^2 \\ &= (1-\mu^2\sigma^2\alpha_k^4)dist(x_k,\Omega^*)^2.\end{aligned}$$

Since $\mu \in (0,1)$, $\sigma \in (0,1)$ and $\alpha_k \in (0,1]$, we have that $(1-\mu^2\sigma^2\alpha_k^4) \in (0,1)$. Therefore, we obtain that the sequence $\{dist(x_k,\Omega^*)\}$ Q-linearly converges to 0. Therefore, the whole sequence $\{x_k\}$ converges to x^* R-linearly. □

5. Numerical Experiments

In this section, numerical results are given to substantiate the efficacy of the proposed Algorithm 2.1, herein denoted as *MBCG*. We compare it with two other methods from the literature, namely, an efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints [4], herein denoted as *ETT*, and a derivative-free iterative method for nonlinear monotone equations with convex constraints, denoted as *PDY* [10]. The methods are compared using *NI*, *NFE* and *CPU*, where *NI* presents the number of iterations, *NFE* is the number of function evaluations and *CPU* is the time in seconds. All codes are written in MATLAB R2016a and are tested using the following test problems with different initial starting points and various dimensions.

Problem 1. [10].

$$F_i(x) = e^{x_i} - 1, \ i = 1,2,3,\dots,n,$$

and $\Omega = \mathbb{R}^n_+$.

Problem 2. [10].

$$\begin{aligned}F_1(x) &= x_1 - e^{\cos(\frac{x_1+x_2}{n+1})}, \\ F_i(x) &= x_i - e^{\cos(\frac{x_{i-1}+x_i+x_{i+1}}{n+1})}, \ i = 2,3,\dots,n-1, \\ F_n(x) &= 2x_n - e^{\cos(\frac{x_{n-1}+x_n}{n+1})},\end{aligned}$$

and $\Omega = \mathbb{R}^n_+$.

Problem 3. [2].

$$F_i(x) = x_i - \sin(|x_i - 1|), \ i = 1,2,3,\dots,n,$$

and $\Omega = \{x \in \mathbb{R} : \sum_{i=1}^n x_i \le n, x_i \ge 0\}$.

Problem 4. [10].

$$\begin{aligned}F_1(x) &= 2x_1 + 0.5h^2(x_1+h)^3 - x_2, \\ F_i(x) &= 2x_i + 0.5h^2(x_i+ih)^3 - x_{i-1} + x_{i+1}, \\ &\quad i = 2,3,\dots,n-1, \\ F_n(x) &= 2x_n + 0.5h^2(x_n+nh)^3 - x_{n-1},\end{aligned}$$

where $h = \frac{1}{n+1}$ and $\Omega = \mathbb{R}^n_+$.

Problem 5. [4].

$$F_i(x) = x_i - \sin(|x_i| - 1),\ i = 1,2,3,...,n,$$

where $\Omega = \{x \in \mathbb{R} : \sum_{i=1}^n x_i \leq n, x_i \geq -1\}$.

Problem 6. [5].

$$F_i(x) = e^{2x_i} + 3\sin(x_i)\cos(x_i) - 1,\ i = 1,2,3,...,n,$$

and $\Omega = \mathbb{R}^n_+$.

In our experiments, all the algorithms are stopped whenever the inequality $\|F_k\| \leq 10^{-5}$ is satisfied, or the total number of iterations exceeds 5000. The parameters used in *ETT* and *PDY* methods are set as in respective papers. The parameters in *MBCG* are selected as $\sigma = 10^{-4}$, $\rho = 0.5$, $r = 10^{-2}$ and $c = 1$. The results are listed in Table 1, where DIM stands for the dimension of the test problems. We tested the given problems with initial points $x_0^1 = (10,10,...,10)^T$, $x_0^2 = (-10,-10,...,-10)^T$, $x_0^3 = (0.1,0.1,...,0.1)^T$ and $x_0^4 = (-0.1,-0.1,...,-0.1)^T$.

We see in Table 1 that the proposed *MBCG* method performs generally better than the other two methods. In order to further make detailed comparison of the proposed method with the other methods, we use the performance profiles tool proposed by Dolan and Moré [3]. We show the performance profiles in Figures 1-3, where Figure 1 shows performance profile of number of iterations, Figure 2 gives performance profile of number of function evaluations and Figure 3 is the performance profile of CPU time. From Figures 1-3, it can be readily seen that the proposed *MBCG* method out-performed both the two methods in all the comparable characteristics, hence the proposed method is both effective and efficient.

6. Conclusion

In this paper, we proposed a derivative-free conjugate gradient-based projection method based on the memoryless BFGS update. The proposed method is free from derivative evaluations, and therefore, is suitable for solving large-scale nonlinear monotone equations with convex constraints. The method also satisfies the descent condition independent of any line search. Global convergence of the proposed method was established and numerical results from a number of benchmark test problems from the literature validate the efficacy of the method.

Table 1. Numerical results for Problems 1-6.

Prob	x_0	DIM	NI			NFE			CPU		
			MBCG	ETT	PDY	MBCG	ETT	PDY	MBCG	ETT	PDY
1	x_0^1	50000	24	31	43	85	102	402	0.3379	0.0893	0.3123
		100000	24	32	56	85	105	577	0.1367	0.1476	0.8077
		150000	25	32	68	88	105	740	0.3053	0.3072	2.0467
	x_0^2	50000	1	1	1	3	3	5	0.0026	0.0023	0.0037
		100000	1	1	1	3	3	5	0.0056	0.0057	0.0080
		150000	1	1	1	3	3	5	0.0090	0.0086	0.0139
	x_0^3	50000	19	25	12	54	72	33	0.0420	0.0496	0.0239
		100000	20	26	13	57	75	36	0.1052	0.1025	0.0535
		150000	20	27	13	57	78	36	0.2110	0.2280	0.1125
	x_0^4	50000	1	1	1	3	3	3	0.0016	0.0016	0.0016
		100000	1	1	1	3	3	3	0.0033	0.0032	0.0032
		150000	1	1	1	3	3	3	0.0083	0.0075	0.0079
2	x_0^1	50000	25	34	36	72	99	179	0.3151	0.3953	0.6906
		100000	26	35	47	75	102	267	0.6156	0.8341	2.1092
		150000	26	35	53	75	102	315	1.0008	1.2912	3.7593
	x_0^2	50000	18	23	21	50	65	76	0.2236	0.2759	0.3326
		100000	2	2	26	2	2	109	0.0184	0.0159	0.8587
		150000	2	2	28	2	2	121	0.0247	0.0222	1.4422
	x_0^3	50000	24	32	20	69	93	69	0.2718	0.3640	0.2954
		100000	24	33	25	69	96	101	0.5701	0.7771	0.8108
		150000	25	33	26	72	96	107	0.9508	1.1478	1.3061
	x_0^4	50000	24	32	21	69	93	74	0.2523	0.3776	0.2711
		100000	25	33	25	72	96	101	0.5704	0.7538	0.7864
		150000	25	33	27	72	96	114	0.9425	1.1584	1.4236
3	x_0^1	50000	7	13	17	18	36	67	0.0750	0.0393	0.0600
		100000	8	15	19	21	42	75	0.0512	0.0768	0.1235
		150000	8	15	21	21	42	86	0.0954	0.1534	0.2696
	x_0^2	50000	10	16	23	26	44	96	0.0271	0.0430	0.0827
		100000	10	17	23	26	47	96	0.0577	0.0858	0.1580
		150000	10	17	26	26	47	118	0.1100	0.1565	0.3636
	x_0^3	50000	8	14	17	21	39	63	0.0209	0.0410	0.0622
		100000	8	14	18	21	39	68	0.0451	0.0639	0.1013
		150000	8	14	18	21	39	68	0.0826	0.1254	0.2031
	x_0^4	50000	8	14	18	21	39	68	0.0209	0.0349	0.0571
		100000	8	14	18	21	39	68	0.0407	0.0673	0.1099
		150000	8	14	20	21	39	77	0.0869	0.1254	0.2366
4	x_0^1	50000	27	255	49	104	783	316	0.9618	7.3459	3.1238
		100000	27	256	60	104	786	420	1.9393	14.3162	7.8144
		150000	28	256	71	108	786	528	3.0205	21.2274	14.5829
	x_0^2	50000	1	109	1	5	328	9	0.0406	2.9721	0.0725
		100000	1	109	1	5	328	9	0.0811	5.9591	0.1522
		150000	1	109	1	5	328	10	0.1257	9.2021	0.2518
	x_0^3	50000	21	243	15	79	737	62	0.7953	6.5503	0.5490
		100000	22	244	14	83	740	57	1.5362	13.2444	1.0798
		150000	22	244	14	83	740	57	2.3883	20.5879	1.5491
	x_0^4	50000	1	81	1	4	244	5	0.0373	2.2469	0.0492
		100000	1	81	1	5	244	5	0.0789	4.4413	0.0797
		150000	1	81	1	5	244	5	0.1233	6.7274	0.1332
5	x_0^1	50000	9	14	19	23	38	76	0.0262	0.0347	0.0682
		100000	10	16	24	26	44	100	0.0527	0.0785	0.1618
		150000	10	16	26	26	44	113	0.1113	0.1795	0.3531
	x_0^2	50000	8	15	19	21	42	75	0.0347	0.0413	0.0627
		100000	8	15	19	21	42	76	0.0462	0.0825	0.1236
		150000	8	15	21	21	42	86	0.1048	0.1499	0.2912
	x_0^3	50000	8	14	18	21	39	67	0.0207	0.0373	0.0563
		100000	8	15	20	21	42	77	0.0404	0.0713	0.1174
		150000	8	15	20	21	42	77	0.0844	0.1351	0.2348
	x_0^4	50000	8	14	17	21	39	63	0.0198	0.0354	0.0501
		100000	8	14	18	21	39	68	0.0367	0.0656	0.0979
		150000	8	14	18	21	39	68	0.0892	0.1312	0.2091
6	x_0^1	50000	23	9	40	110	50	629	0.2870	0.1101	1.1184
		100000	23	9	53	110	50	889	0.4219	0.2140	3.0590
		150000	24	9	59	114	50	1008	0.7554	0.3450	5.8129
	x_0^2	50000	19	9	14	68	22	56	0.1164	0.0361	0.0893
		100000	19	9	14	68	22	56	0.2292	0.0712	0.1821
		150000	20	9	17	72	22	74	0.4294	0.1282	0.4235
	x_0^3	50000	16	10	11	75	36	50	0.0789	0.0363	0.0501
		100000	16	10	11	75	36	50	0.1590	0.0720	0.1004
		150000	16	10	11	75	36	50	0.3070	0.1338	0.1919
	x_0^4	50000	16	10	10	75	36	45	0.0758	0.0363	0.0445
		100000	16	10	11	75	36	50	0.1647	0.0735	0.1064
		150000	16	11	11	75	40	50	0.3029	0.1515	0.1902

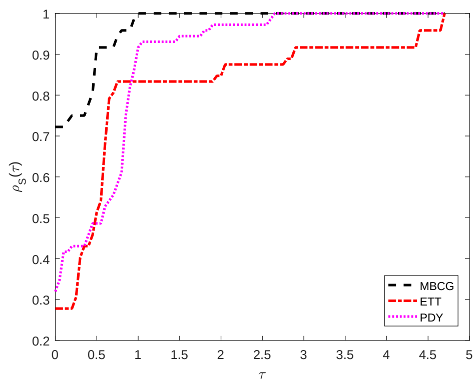

Figure 1. Iterations performance profile

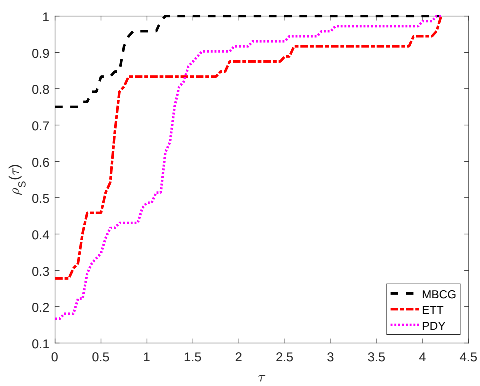

Figure 2. Function evaluations performance profile

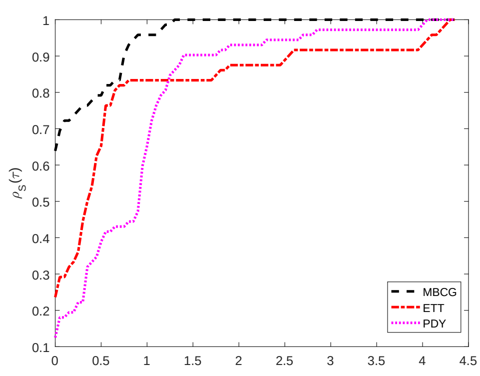

Figure 3. Cpu time performance profile

References

[1] A. B. Abubakar, P. Kumam, H. Mohammad and A. M. Awwal, An efficient conjugate gradient method for convex constrained monotone nonlinear equations with applications, *Mathematics*, 7:767 (2019), https://doi.org/10.3390/math7090767.

[2] Y. Ding, Y. Xiao and J. Li, A class of conjugate gradient methods for convex constrained monotone equations, *Optim.*, 66(12) (2017), 2309–2328 .

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

[4] P. Gao and C. He, An effecient three-term conjugate gradient method for nonlinear monotone equations with convex constraints, *Calcolo*, 55:53 (2018), https://dx.doi.org/10.1007/s10092-018-0291-2.

[5] P. Gao, C. He and Y. Liu, An adaptive family of projection methods for constrained monotone equations with applications, *Appl. Math. Comput.*, 359 (2019), 1–16.

[6] J. Guo and Z. Wan, A modified spectral PRP conjugate gradient projection method for solving large-scale monotone equations and its application in compressed sensing,*Math. Prob. Eng.*, 2019 Article ID 5261830 (2019), 17 pages.

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

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

[9] 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.

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

[11] S.Y. Liu, Y.Y. Huang and H.W. Jiao, Sufficient descent conjugate gradient methods for solving convex constrined nonlinear monotone equations, *Abstr. Appl. Anal.*, 2014 Article ID 305643 (2014), 12 pages.

[12] Z. Liu, S. Du and R. Wang, A new conjugate gradient projection method for solving stochastic generalized linear complementarity problems, *J. Appl. Math. Phys.*, 4 (2016), 1024–1031.

[13] I.E. Livieris, V. Tampakas and P. Pintelas, A descent hybrid conjugate gradient method based on the memoryless BFGS update, *Numer. Algor.*, 79(4) (2018), 1169–1185.

[14] 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.

[15] M.V. Solodov and B.F. Svaiter, A globally convergent inexact newton method for systems of monotone equations, In Fukushima M., Qi L. (eds) Reformulation: Nonsmooth,

Piecewise Smooth, Semismoothing methods, Applied Optimization, it J. (eds). Springer, Boston, MA, (1998), 355–369.

[16] P.S. Stanimirović, B. Ivanov, S. Djordjević and I. Brajević, New hybrid conjugate gradient and Broyden-Fletcher-Goldfarb-Shanno conjugate gradient methods, *J. Optim. Theory. Appl.*, 178 (2018), 860–884.

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

[18] S. Wang and H. Guan, A scaled conjugate gradient method for solving monotone nonlinear equations with convex constraints, *J. Appl. Math.*, 2013 Article ID 286486 (2013), 7 pages.

[19] 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.

[20] 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.

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

[22] L. Zheng, A modified PRP projection method for nonlinear equations with convex constraints, *Int. J. Pure. Appl. Math.*, 79 (2012), 87–96 .

⋆⋆⋆⋆⋆⋆⋆⋆⋆
ISSN(P):2319 − 3786
Malaya Journal of Matematik
ISSN(O):2321 − 5666
⋆⋆⋆⋆⋆⋆⋆⋆⋆