Mompati S. Koorapetse · P. Kaelo

Globally convergent three-term conjugate gradient projection methods for solving nonlinear monotone equations

Abstract In this paper, we propose two derivative-free conjugate gradient projection methods for systems of large-scale nonlinear monotone equations. The proposed methods are shown to satisfy the sufficient descent condition. Furthermore, the global convergence of the proposed methods is established. The proposed methods are then tested on a number of benchmark problems from the literature and preliminary numerical results indicate that the proposed methods can be efficient for solving large scale problems and therefore are promising.

Mathematics Subject Classification 90C30 · 90C56 · 65K05 · 65K10

1 Introduction

In this paper we consider the problem of finding a solution \( x^* \) of the constrained system of nonlinear equations

\[
F(x) = 0, \quad \text{subject to} \quad x \in \Omega, 
\]

where \( \Omega \subseteq \mathbb{R}^n \) is a nonempty closed convex set and \( F : \mathbb{R}^n \to \mathbb{R}^n \) is a continuous and monotone function, i.e.,

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

This problem has many important applications in applied mathematics, economics and engineering. For instance, the economic equilibrium problem [4] can be reformulated as Problem (1.1).

There exist a number of numerical methods for solving Problem (1.1). These include the trust region methods [20], Newton methods [10], quasi-Newton methods [3], the Levenberg-Marquardt methods [11], derivative-free methods [7,12,15], the subspace methods [23], gradient-based projection methods [1,13,14,21]
and conjugate gradient methods [8]. All these methods are iterative, that is, starting with \( x_k \) the next iterate is found by

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

where \( d_k \) is the search direction and \( \alpha_k \) is the step length.

The gradient projection method [1,6,12–15,19,21,25] is the most effective method for solving systems of large-scale nonlinear monotone equations. The projection concept was first proposed by Goldstein [6] for convex programming in Hilbert spaces. It was then extended by Solodov and Svaiter [17]. In their paper, Solodov and Svaiter constructed a hyperplane \( H_k \) which strictly separates the current iterate \( x_k \) from the solution set of Problem (1.1) as

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

where \( z_k = x_k + \alpha_k d_k \) is generated by performing some line search along the direction \( d_k \) such that

\[ F(z_k)^T(x - z_k) > 0. \]

The hyperplane \( H_k \) strictly separates \( x_k \) from the solutions of Problem (1.1) and from the monotonicity of \( F \) we have that for any \( x^* \) such that \( F(x^*) = 0 \),

\[ F(z_k)^T(x^* - z_k) \leq 0. \]

Now, after constructing this hyperplane, Solodov and Svaiter’s next iteration point, \( x_{k+1} \), is constructed by projecting \( x_k \) onto \( H_k \) as

\[ x_{k+1} = x_k - \frac{F(z_k)^T(x_k - z_k)}{\| F(z_k) \|^2} F(z_k). \] (1.3)

Recently, most research has been focussed on the conjugate gradient projection methods for solving Problem (1.1). For the conjugate gradient projection method, the search direction is found using

\[ d_k = \begin{cases} -F_k, & \text{if } k = 0, \\ -F_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \] (1.4)

where \( F_k = F(x_k) \) and \( \beta_k \) is a parameter, and \( x_{k+1} \) is obtained by (1.3). One such method is that of Sun and Liu [19] where the constrained nonlinear system (1.1) with convex constraints is solved by

\[ x_{k+1} = P_{\Omega} \left[ x_k - \frac{F(z_k)^T(x_k - z_k)}{\| F(z_k) \|^2} F(z_k) \right], \] (1.5)

where \( z_k = x_k + \alpha_k d_k \) and \( d_k \) is computed using the conjugate gradient scheme (1.4) with \( \beta_k \) defined as

\[ \beta_k^{NN} = \frac{\| F(x_k) \|^2 - \max\{0, \frac{\| F(x_k) \|^2}{\| F(x_k) \|^2 F(x_k)^T F(x_k) \}}}{\| F(x_k) \|^2 \| d_{k-1} \|^2}, \]

and \( \lambda > 1 \) is a constant. This method was shown to be globally convergent using the line search

\[ -F(x_k + \alpha_k d_k)^T d_k \geq \mu \alpha_k \| F(x_k + \alpha_k d_k) \| \| d_k \|^2, \] (1.6)

where \( \mu > 0, \alpha_k = \rho^m k \) and \( \rho \in (0, 1) \) with \( m_k \) being the smallest nonnegative integer \( m \) such that (1.6) holds.

In equation (1.5), \( P_{\Omega}[\cdot] \) denotes the projection mapping from \( \mathbb{R}^n \) onto the convex set \( \Omega \), i.e.,

\[ P_{\Omega}[x] = \arg\min_{y \in \Omega} \| x - y \|. \]

This is an optimization problem that minimizes the distance between \( x \) and \( y \), where \( y \in \Omega \). Also, the projection operator is such that it is nonexpansive, i.e.,

\[ \| P_{\Omega}[x] - P_{\Omega}[y] \| \leq \| x - y \|, \quad \forall x, y \in \mathbb{R}^n. \] (1.7)
Ahookhosh et al. [1] also extended the projection method by Solodov and Svaiter [17] to three-term conjugate gradient method where \( x_{k+1} \) is given by (1.3) and \( z_k \) is found using the direction

\[
d_k = \begin{cases} 
-F_k, & \text{if } k = 0, \\
-F_k + \beta_k^{PRP} w_{k-1} - \theta_k y_{k-1}, & \text{if } k \geq 1,
\end{cases}
\]

where \( \beta_k^{PRP} = \frac{F_k^T y_{k-1}}{\|F_{k-1}\|^2} \), \( y_k = F_k - F_{k-1} \), \( w_{k-1} = z_{k-1} - x_{k-1} = \alpha_{k-1} d_{k-1} \), and \( \theta_k \) is such that the sufficient descent condition

\[
F_k^T d_k \leq -\tau \|F_k\|^2, \quad \forall k \geq 0, \quad \tau > 0,
\]

holds. To satisfy Condition (1.8), the authors suggest using

\[
\theta_k = \frac{(F_k^T y_{k-1}) \|w_{k-1}\|^2}{\|F_{k-1}\|^2}
\]

or

\[
\theta_k = \frac{F_k^T w_{k-1}}{\|F_{k-1}\|^2} + \frac{(F_k^T y_{k-1}) \|y_{k-1}\|^2}{\|F_{k-1}\|^4}.
\]

This leads to two derivative-free algorithms DFPB1 and DFPB2, respectively. Other conjugate gradient projection methods can be found in [9,12–15,18,25]. In this paper we propose two globally convergent derivative-free conjugate gradient projection methods.

The rest of the paper is structured as follows: In the next section, motivation and the details of the proposed algorithm are given. The sufficient descent property and the global convergence of the proposed algorithm are presented in Sect. 3. Numerical results and conclusion are presented in Sects. 4 and 5, respectively.

2 Motivation and algorithm

Before presenting the proposed methods, we first present some work that motivated us. We start with the work of Hager and Zhang [8] where the unconstrained minimization problem

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

is solved, with \( f : \mathbb{R}^n \to \mathbb{R} \) being a continuously differentiable function. The iterations \( x_{k+1} = x_k + \alpha_k d_k \) are generated using the direction

\[
d_k = \begin{cases} 
-g_k, & \text{if } k = 0, \\
-g_k + \tilde{\beta}_N^k d_{k-1}, & \text{if } k \geq 1,
\end{cases}
\]

where \( g_k = \nabla f(x_k) \) is the gradient of \( f \) at \( x_k \), and

\[
\tilde{\beta}_N^k = \max\{\beta_N^k, \eta_k\}, \\
\eta_k = \frac{-1}{\|d_{k-1}\| \min\{\eta, \|g_{k-1}\|\}},
\]

and

\[
\beta_N^k = \frac{1}{d_{k-1}^T y_{k-1}} \left( y_{k-1} - \frac{2 \|y_{k-1}\|^2}{d_{k-1}^T y_{k-1}} d_{k-1} \right) ^T g_k,
\]

\( y_{k-1} = g_k - g_{k-1} \), and \( \eta > 0 \) is a constant. The parameter \( \tilde{\beta}_N^k \) satisfies the descent condition

\[
d_k^T g_k \leq -c \|g_k\|^2, \quad \forall k \geq 0,
\]

where \( c = \frac{7}{8} \) and its global convergence was established by means of the standard Wolfe line search technique.

In Yuan [22], a modified \( \beta_k^{MPRP} \) formula

\[
\beta_k^{MPRP} = \beta_k^{PRP} - \min \left\{ \beta_k^{PRP}, \frac{\sigma \|y_{k-1}\|^2}{\|g_{k-1}\| \|s_k^T d_{k-1}\|^2} \right\} = \max\{\beta_k^{DPRP}, 0\},
\]

where \( s_k = g_k - g_{k-1} - \frac{1}{\|g_{k-1}\|^2} g_{k-1}^T g_{k-1} \) and \( \sigma = \frac{1}{2} \frac{\|g_{k-1}\|^2}{\|g_{k-1}\| \|s_k^T d_{k-1}\|^2} \), and its global convergence was established by means of the standard Wolfe line search technique.
is presented where $\sigma > \frac{1}{4}$ is a constant, and

$$\beta_k^{\text{DPRP}} = \beta_k^{\text{PRP}} - \frac{\sigma}{\| g_{k-1} \|^2} g_k^T d_{k-1}$$

(2.6)

is an earlier modification of $\beta_k^{\text{DPRP}}$ (see [22] and reference therein). Other modified formulas similar in pattern to (2.5) were proposed in [22] using $\beta_k^{\text{CD}}, \beta_k^{\text{LS}}, \beta_k^{\text{DY}}$ and $\beta_k^{\text{HS}}$. These methods satisfy the sufficient descent condition (2.4) and were also shown to converge globally under the standard Wolfe line search conditions.

Assumption 3.1

In [2], Dai suggests that any $\beta_k$ of the form $\beta_k = g_k^T z_k$, where $z_k \in \mathbb{R}^n$ is any vector, can be modified as

$$\beta_k^{\text{GSD}} = g_k^T z_k - \sigma \| z_k \|^2 g_k^T d_{k-1},$$

(2.7)

with $\sigma > \frac{1}{4}$ and will satisfy the sufficient descent condition (2.4) with $\epsilon = 1 - \frac{1}{4\sigma}$. In order to prove the global convergence of (2.7) an assumption that $\beta_k^{\text{GSD}} \geq \eta_k$, where $\eta_k$ is defined as in (2.3), is made in Nakamura et al. [16]. That is, they proposed

$$\beta_k^{\text{GSD}} = \max\{\beta_k^{\text{GSD}}, \xi_k\},$$

(2.8)

where $\xi_k \in [\eta_k, 0]$.

Motivated by the work of [1,8,16,22], we propose a direction

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

(2.9)

where

$$\beta_k^{\text{SP}} = \begin{cases} \beta_k^{\text{DPRP}}, & \text{if } F_k^T w_{k-1} \geq 0, \\
\max\{\beta_k^{\text{DPRP}}, \xi_k\}, & \text{if } F_k^T w_{k-1} < 0,
\end{cases}$$

(2.10)

with $\xi_k \in [\eta_k, 0]$, $\eta_k$ and $\beta_k^{\text{DPRP}}$ as given in (2.3) and (2.6), respectively. Note that if $\xi_k = 0$, then we have (2.5) and if $\xi_k = \eta_k$, then $\beta_k^{\text{SP}} = \max\{\beta_k^{\text{DPRP}}, \eta_k\}$. Also, since $\eta_k$ is negative, it follows that $\beta_k^{\text{SP}} = \max\{\beta_k^{\text{DPRP}}, \xi_k\} \in [\beta_k^{\text{DPRP}}, \beta_k^{\text{MPRP}}]$. The term $\theta_k$ is determined such that Condition (1.8) holds. Below, we present our algorithm.

Algorithm 1 Three-term Conjugate Gradient Projection based algorithm (3TCGPB)

1: Given $x_0 \in \Omega$, the parameters $\sigma, \mu, \epsilon, s > 0$ and $\rho \in (0,1)$. Set $k = 0$.
2: for $k = 0, 1, \ldots$ do
3: if $\| F(x_k) \| \leq \epsilon$, then stop. Otherwise, go to Step 4.
4: Compute $d_k$ by (2.9) and (2.10) guaranteeing the condition (1.8).
5: Find $z_k = x_k + \alpha_k d_k$, where $\alpha_k = \max\{\rho, \rho^2, \rho^3, \ldots\}$ is such that (1.6) is satisfied.
6: Compute $x_{k+1} = \rho \Omega \left[ x_k - \frac{F(z_k) - F(x_k)}{F(z_k) - F(x_k)} F(z_k) \right]$
7: Set $k = k + 1$ and go to Step 3.
8: end for

3 Global convergence of the proposed method

In order to establish the global convergence of the proposed approach, the following assumption is necessary.

Assumption 3.1

A1. The solution set $\Omega$ is nonempty.
A2. The mapping function $F(x)$ is monotone on $\mathbb{R}^n$, i.e.,

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

A3. The function $F(x)$ is Lipschitz continuous on $\mathbb{R}^n$, i.e., there exists a positive constant $L$ such that

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

(3.1)
Lemma 3.2 (3TCGPB1) Let $d_k$ be generated by Algorithm 1 with
\[
\theta_k = \sigma \left( \frac{(F_k^T y_{k-1}) \parallel w_{k-1} \parallel^2 - (F_k^T y_{k-1})(d_{k-1}^T w_{k-1})}{\parallel F_{k-1} \parallel^4} \right),
\]
and $\beta^SP_k \in [\beta^DP_k, \beta^MP_k]$. Then
\[
F_k^T d_k \leq -\tau \parallel F_k \parallel^2, \quad \forall k \geq 0,
\]
where $\tau = (1 - \frac{1}{4\sigma})$ and $\sigma > \frac{1}{4}$.

Proof For $k = 0$, we have $F_0^T d_0 = -\parallel F_0 \parallel^2$ which satisfies (3.3). For $k \geq 1$, we divide the rest of the proof into the following cases.

Case I Suppose $F_k^T w_{k-1} \geq 0$. Using the inequality
\[
\|u^T v\| \leq \frac{1}{2} (\|u\|^2 + \|v\|^2),
\]
where $u = \frac{1}{\sqrt{2\sigma}} \parallel F_{k-1} \parallel^2 F_k$ and $v = \sqrt{2\sigma} (F_k^T y_{k-1})w_{k-1}$, and Eq. (2.9), we obtain that
\[
F_k^T d_k \leq \frac{-(1 - \frac{1}{4\sigma}) \parallel F_k \parallel^2 \parallel F_{k-1} \parallel^4 + \psi - \theta_k \parallel F_{k-1} \parallel^4 (F_k^T y_{k-1})}{\parallel F_{k-1} \parallel^4},
\]
where
\[
\psi = \sigma ((F_k^T y_{k-1})^2 \parallel w_{k-1} \parallel^2 - (F_k^T d_{k-1}) \parallel y_{k-1} \parallel^2 (F_k^T w_{k-1})).
\]
Substituting (3.2) into (3.5) immediately gives
\[
F_k^T d_k \leq -\left(1 - \frac{1}{4\sigma}\right) \parallel F_k \parallel^2.
\]

Case II Suppose $F_k^T w_{k-1} < 0$. If $\beta^SP_k = \beta^DP_k$ the proof follows from Case I. On the other hand, if $\beta^SP_k \neq \beta^DP_k$, then $\beta^DP_k \leq \beta^SP_k \leq 0$. It follows from (2.9) that
\[
F_k^T d_k = -\parallel F_k \parallel^2 + \beta^SP_k (F_k^T w_{k-1}) F_k (F_k^T y_{k-1}) - \theta_k (F_k^T y_{k-1}) \leq -\parallel F_k \parallel^2 + \beta^DP_k (F_k^T w_{k-1}) - \theta_k (F_k^T y_{k-1}).
\]
Substituting (3.2) into (3.6) and using the inequality (3.4) we get that
\[
F_k^T d_k \leq -\left(1 - \frac{1}{4\sigma}\right) \parallel F_k \parallel^2.
\]
Hence (3.3) is proved.

Lemma 3.3 (3TCGPB2) Consider the search direction $d_k$ generated by Algorithm 1 and
\[
\theta_k = \frac{(F_k^T w_{k-1}) \parallel F_{k-1} \parallel^2 - \sigma (F_k^T y_{k-1})(d_{k-1}^T w_{k-1})}{\parallel F_{k-1} \parallel^4},
\]
and $\beta^SP_k \in [\beta^DP_k, \beta^MP_k]$. Then
\[
F_k^T d_k \leq -\parallel F_k \parallel^2, \quad \forall k \geq 0.
\]
Proof For $k = 0$, we have $F_0^T d_0 = -\| F_0 \|^2$ which satisfies (3.8). For $k \geq 1$, we divide the rest of the proof into the following cases.

Case I Suppose $F_k^T w_{k-1} \geq 0$. From (2.9) we have
\begin{equation}
F_k^T d_k = -\| F_k \|^2 \| F_{k-1} \|^4 + \delta - \theta_k \| F_{k-1} \|^4 (F_k^T y_{k-1}), \tag{3.9}
\end{equation}
where
\[ \delta = (F_k^T y_{k-1}) \| F_{k-1} \|^2 - (F_k^T w_{k-1}) - \sigma (F_k^T d_{k-1}) \| y_{k-1} \|^2 (F_k^T w_{k-1}). \]
Substituting (3.7) into (3.9) gives
\[ F_k^T d_k = -\| F_k \|^2. \]

Case II Suppose $F_k^T w_{k-1} < 0$. If $\beta_k^{SP} = \beta_k^{DPRP}$ the proof follows from Case I. For $\beta_k^{SP} \neq \beta_k^{DPRP}$, we have from (3.6) that
\[ F_k^T d_k \leq -\| F_k \|^2 + \beta_k^{DPRP} F_k^T w_{k-1} - \theta_k (F_k^T y_{k-1}). \tag{3.10} \]
Substituting (3.7) into (3.10) immediately we obtain that
\[ F_k^T d_k \leq -\| F_k \|^2. \]
Hence, the direction given by (2.9) and (2.10) is a descent direction.

Lemma 3.4 The line search procedure (1.6) of Step 5 in Algorithm 1 is well-defined.

Proof We proceed by contradiction. Suppose that for some iterate indexes such as $\hat{k}$ the condition (1.6) does not hold. As a result, by setting $\alpha_k = \rho^m s$, it can be concluded that
\[ -F(x_\hat{k} + \rho^m s d_\hat{k})^T d_\hat{k} < \mu \rho^m s \| F(x_\hat{k} + \rho^m s d_\hat{k}) \| \| d_\hat{k} \|^2, \quad \forall m \geq 0. \]
Letting $m \to \infty$ and using the continuity of $F$ yields
\[ -F(x_\hat{k})^T d_\hat{k} \leq 0. \tag{3.11} \]
Combining (3.11) with the sufficient descent property (1.8), we have $F(x_\hat{k}) = 0$. Obviously from Steps 3 and 5 of Algorithm 1, we have $F(x_\hat{k}) \neq 0$ if the line search (1.6) is executed, which contradicts with $F(x_\hat{k}) = 0$. \qed

Lemma 3.5 Suppose Assumption 3.1 holds and let $\{x_k\}$ and $\{z_k\}$ be sequences generated by Algorithm 1, then $\{x_k\}$ and $\{z_k\}$ are bounded. Furthermore, it holds that
\[ \lim_{k \to \infty} \alpha_k \| d_k \| = 0. \tag{3.12} \]

Proof Since $x^*$ is such that $F(x^*) = 0$ and the mapping $F$ is monotone, then $F(z_k)^T (z_k - x^*) \geq 0$. By using (1.6), we have
\[ F(z_k)^T (x_k - z_k) \geq \mu \| F(z_k) \| \| x_k - z_k \|^2 > 0. \tag{3.13} \]
For $x^* \in \Omega$ we have from (1.5) and (1.7) that
\begin{align}
\| x_{k+1} - x^* \|^2 &= \| P_\Omega(x_k - \nu_k F(z_k)) - x^* \|^2 \\
&\leq \| x_k - \nu_k F(z_k) - x^* \|^2 \\
&= \| x_k - x^* \|^2 - 2\nu_k F(z_k)^T (x_k - x^*) + \nu_k^2 \| F(z_k) \|^2, \tag{3.14}
\end{align}
where
\[ \nu_k = \frac{F(z_k)^T (x_k - z_k)}{\| F(z_k) \|^2}. \]
By the monotonicity of $F$, we have that
\[
F(z_k)^T(x_k - x^*) = F(z_k)^T(x_k - z_k) + F(z_k)^T(z_k - x^*) \\
\geq F(z_k)^T(x_k - z_k) + F(x^*)^T(z_k - x^*) \\
= F(z_k)^T(x_k - z_k).
\] (3.15)

Using (3.13) and (3.15), we have from (3.14) that
\[
\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\nu_k F(z_k)^T(x_k - z_k) + \nu_k^2 \|F(z_k)\|^2 \\
= \|x_k - x^*\|^2 - \frac{(F(z_k)^T(x_k - z_k))^2}{\|F(z_k)\|^2} \\
\leq \|x_k - x^*\|^2 - \mu^2 \|x_k - z_k\|^4.
\] (3.16)

which means that
\[
\|x_{k+1} - x^*\| \leq \|x_k - x^*\|, \quad \forall k \geq 0.
\] (3.17)

This shows that $\{\|x_k - x^*\|\}$ is a decreasing sequence and hence $\{x_k\}$ is bounded. Also, from (3.13), it follows that
\[
\mu \|F(z_k)\| \|x_k - z_k\|^2 \leq F(z_k)^T(x_k - z_k) \\
\leq \|F(z_k)\| \|x_k - z_k\|,
\] (3.18)

which implies that
\[
\mu \|x_k - z_k\| \leq 1,
\]
indicating that $\{z_k\}$ is bounded. Furthermore, it follows from (3.16) that
\[
\mu^2 \sum_{k=0}^{\infty} \|x_k - z_k\|^4 \leq \sum_{k=0}^{\infty} (\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2) < \infty,
\]
and thus
\[
\lim_{k \to \infty} \|x_k - z_k\| = \lim_{k \to \infty} \alpha_k \|d_k\| = 0.
\]

\[\square\]

**Theorem 3.6** Suppose that Assumption 3.1 holds, and the sequence $\{x_k\}$ is generated by Algorithm 1. Then, we have
\[
\lim_{k \to \infty} \inf \|F_k\| = 0.
\] (3.19)

**Proof** We consider the following two possible cases.

**Case I** If $\lim_{k \to \infty} \inf \|F_k\| = 0$, then this together with the continuity of $F$ implies that the sequence $\{x_k\}$ has some accumulation point $x^*$ such that $F(x^*) = 0$. From (3.17), it holds that $\{\|x_k - x^*\|\}$ converges and since $x^*$ is an accumulation point of $\{x_k\}$, it holds that $\{x_k\}$ converges to $x^*$.

**Case II** If $\lim_{k \to \infty} \inf \|F_k\| > 0$, then there exists $\epsilon_0$, such that
\[
\|F_k\| \geq \epsilon_0, \quad \forall k \geq 0.
\]

Then, by means of (3.3), we also have
\[
\tau \|F_k\|^2 \leq -F_k^T d_k \leq \|F_k\| \|d_k\| \quad \forall k \geq 0,
\]
where $\tau = 1 - \frac{1}{4\epsilon^2}$. Hence $\|d_k\| \geq \tau \epsilon_0 > 0, \forall k \geq 0$. According to this condition and (3.12), it follows that
\[
\lim_{k \to \infty} \alpha_k = 0.
\]

Therefore, from the line search (1.6), for sufficiently large $k$, we have
\[
-F(x_k + \rho^m s d_k)^T d_k < \mu \rho^m \|F(x_k + \rho^m s d_k)\| \|d_k\|^2.
\] (3.20)
Since \( \{x_k\} \) and \( \{d_k\} \) are both bounded, we can choose a sequence \( \{x_k\} \) and letting \( m \to \infty \) in (3.20), we obtain
\[
-F(x^*)^T d^* \leq 0,
\] (3.21)
where \( x^* \), \( d^* \) are limit points of corresponding subsequences. On the other hand, by (3.3), we obtain
\[
-F(x_k)^T d_k \geq \tau \| F(x_k) \|^2, \quad \forall k \geq 0,
\]
where \( \tau = 1 - \frac{1}{4\eta} \). Letting \( k \to \infty \) in the above inequality, we obtain
\[
-F(x^*)^T d^* \geq \tau \| F(x^*) \|^2.
\] (3.22)
Thus by (3.21) and (3.22), we get \( \| F(x^*) \| = 0 \), and this contradicts the fact that
\[
\lim_{k \to \infty} \inf \| F_k \| > 0.
\] (3.23)
Therefore (3.23) does not hold. \( \square \)

4 Numerical experiments

In this section, we present numerical results obtained from our two proposed methods, 3TCGPB1 and 3TCGPB2, and compare them with the methods proposed by Ahookhosh et al. [1], DFPB1 and DFPB2. All algorithms are coded in MATLAB R2016a and run on a computer with Intel(R) Core(TM) i7-4770 CPU at 3.40GHz and installed memory (RAM) of 8.00 GB. The parameters used in all the four methods are set as \( \rho = 0.7 \) and \( \mu = 0.3 \). Similar to [1], the initial adaptive step length is taken as
\[
s_k = \frac{F_k^T d_k}{(F(x_k + t d_k) - F_k)^T d_k / t},
\]
where \( t = 10^{-6} \). For our two methods 3TCGPB1 and 3TCGPB2, we use additional parameters \( \sigma = 0.7, \eta = 0.01 \), and set \( \xi_k = \eta_k \). We adopt the same termination condition for all the four methods, i.e., we stop the algorithms when the maximum number of iterations exceeds 500 or the inequality \( \| F(x_k) \| \leq \epsilon = 10^{-5} \) is satisfied. Test problems used here are taken from Hu and Wei [9], Sun and Liu [18,19] and Zhang and Zhou [24]. These problems are outlined below.

Problem 4.1 The mapping \( F(\cdot) \) is taken as \( F(x) = (F_1(x), F_2(x), F_3(x), \ldots, F_n(x))^T \), where
\[
F_i(x) = e^{x_i} - 1, \quad \text{for} \quad i = 1, 2, 3, \ldots, n, \quad \text{and} \quad \Omega = R^n_+.
\]
Initial guess \( x_0 = (1, 1, 1, \ldots, 1)^T \).

Problem 4.2 The mapping \( F(\cdot) \) is taken as \( F(x) = (F_1(x), F_2(x), F_3(x), \ldots, F_n(x))^T \), where
\[
F_1(x) = (3 - x_1)x_1 - 2x_2 + 1,
F_i(x) = (3 - x_i)x_i - x_{i-1} - 2x_{i+1} + 1, \quad \text{for} \quad i = 2, 3, \ldots, n - 1,
F_n(x) = (3 - x_n)x_n - x_{n-1} + 1,
\]
and \( \Omega = R^n \). Here we set \( x_0 = (-1, -1, -1, \ldots, -1)^T \).

Problem 4.3 The mapping \( F(\cdot) \) is taken as \( F(x) = (F_1(x), F_2(x), F_3(x), \ldots, F_n(x))^T \), where
\[
F_i(x) = x_i - \sin | x_i |, \quad \text{for} \quad i = 1, 2, 3, \ldots, n, \quad \text{and} \quad \Omega = R^n.
\]
Initial guess \( x_0 = (1, 1, 1, \ldots, 1)^T \).
Table 1 Numerical results of Problem 4.1

| Method     | $N$ | $NI$ | $FE$ | $\| F(x_k) \|$ | $CPU$ |
|------------|-----|------|------|-----------------|-------|
| 3TCGPB1    | 100 | 6    | 18   | $1.20e^{-07}$   | 0.0061|
|            | 1000| 13   | 71   | $9.35e^{-06}$   | 0.0028|
|            | 10,000| 38  | 324  | $3.28e^{-06}$   | 0.0520|
|            | 20,000| 52  | 500  | $3.11e^{-06}$   | 0.1027|
|            | 50,000| 82  | 894  | $7.20e^{-07}$   | 0.3850|
| 3TCGPB2    | 100 | 6    | 18   | $1.20e^{-07}$   | 0.0060|
|            | 1000| 13   | 71   | $9.35e^{-06}$   | 0.0028|
|            | 10,000| 38  | 323  | $9.00e^{-06}$   | 0.0554|
|            | 20,000| 52  | 500  | $3.14e^{-06}$   | 0.1056|
|            | 50,000| 82  | 894  | $7.60e^{-07}$   | 0.3898|
| DFPB1      | 100 | 6    | 18   | $1.20e^{-07}$   | 0.0066|
|            | 1000| 13   | 71   | $9.35e^{-06}$   | 0.0028|
|            | 10,000| 38  | 323  | $6.00e^{-06}$   | 0.0554|
|            | 20,000| 52  | 500  | $3.14e^{-06}$   | 0.1056|
|            | 50,000| 82  | 894  | $7.60e^{-07}$   | 0.3898|
| DFPB2      | 100 | 6    | 18   | $1.20e^{-07}$   | 0.0057|
|            | 1000| 13   | 71   | $9.35e^{-06}$   | 0.0028|
|            | 10,000| 38  | 323  | $6.00e^{-06}$   | 0.0554|
|            | 20,000| 52  | 500  | $3.14e^{-06}$   | 0.1056|
|            | 50,000| 82  | 894  | $7.60e^{-07}$   | 0.3898|

Table 2 Numerical results of Problem 4.2

| Method     | $N$ | $NI$ | $FE$ | $\| F(x_k) \|$ | $CPU$ |
|------------|-----|------|------|-----------------|-------|
| 3TCGPB1    | 100 | 33   | 94   | $9.32e^{-06}$   | 0.0058|
|            | 1000| 33   | 94   | $9.32e^{-06}$   | 0.0055|
|            | 10,000| 35  | 98   | $9.32e^{-06}$   | 0.0489|
|            | 20,000| 35  | 98   | $9.32e^{-06}$   | 0.0600|
|            | 50,000| 35  | 98   | $9.32e^{-06}$   | 0.1526|
| 3TCGPB2    | 100 | 32   | 92   | $7.12e^{-06}$   | 0.0060|
|            | 1000| 31   | 90   | $7.12e^{-06}$   | 0.0052|
|            | 10,000| 33  | 94   | $7.12e^{-06}$   | 0.0388|
|            | 20,000| 33  | 94   | $7.12e^{-06}$   | 0.0571|
|            | 50,000| 33  | 94   | $7.12e^{-06}$   | 0.1447|
| DFPB1      | 100 | 34   | 96   | $9.11e^{-06}$   | 0.0062|
|            | 1000| 39   | 106  | $9.11e^{-06}$   | 0.0061|
|            | 10,000| 36  | 100  | $9.11e^{-06}$   | 0.0422|
|            | 20,000| 36  | 100  | $9.11e^{-06}$   | 0.0701|
|            | 50,000| 36  | 100  | $9.11e^{-06}$   | 0.1434|
| DFPB2      | 100 | 31   | 89   | $9.51e^{-06}$   | 0.0055|
|            | 1000| 33   | 93   | $9.52e^{-06}$   | 0.0054|
|            | 10,000| 37  | 101  | $9.54e^{-06}$   | 0.0369|
|            | 20,000| 37  | 101  | $9.54e^{-06}$   | 0.0747|
|            | 50,000| 37  | 101  | $9.54e^{-06}$   | 0.1713|

Problem 4.4 The mapping $F(\cdot)$ is taken as $F(x) = (F_1(x), F_2(x), F_3(x), \ldots, F_n(x))^T$, where

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

and $\Omega = R_n^+$. Initial guess $x_0 = (1, 1, 1, \ldots, 1)^T$. 
Table 3 Numerical results of Problem 4.3

| Method         | N   | NI | FE | || F(x_k) || | CPU    |
|----------------|-----|----|----|-------------------------------|--------|
| 3TCGPB1        | 100 | 11 | 24 | 3.60e−06                      | 0.0010 |
|                | 1000| 16 | 55 | 9.22e−06                      | 0.0021 |
|                | 10,000| 39 | 243| 6.14e−06                      | 0.0271 |
|                | 20,000| 54 | 387| 3.12e−06                      | 0.0659 |
|                | 50,000| 82 | 695| 4.37e−06                      | 0.2304 |
| 3TCGPB2        | 100 | 11 | 24 | 3.60e−06                      | 0.0009 |
|                | 1000| 16 | 55 | 9.08e−06                      | 0.0021 |
|                | 10,000| 39 | 243| 6.42e−06                      | 0.0277 |
|                | 20,000| 54 | 387| 1.62e−06                      | 0.0658 |
|                | 50,000| 82 | 695| 4.09e−06                      | 0.2184 |
| DFPB1          | 100 | 11 | 24 | 3.60e−06                      | 0.0012 |
|                | 1000| 16 | 55 | 9.22e−06                      | 0.0020 |
|                | 10,000| 39 | 243| 6.14e−06                      | 0.0267 |
|                | 20,000| 54 | 387| 3.11e−06                      | 0.0638 |
|                | 50,000| 82 | 695| 4.37e−06                      | 0.2262 |
| DFPB2          | 100 | 11 | 24 | 3.60e−06                      | 0.0009 |
|                | 1000| 16 | 55 | 9.27e−06                      | 0.0020 |
|                | 10,000| 39 | 243| 5.67e−06                      | 0.0281 |
|                | 20,000| 54 | 387| 2.87e−06                      | 0.0647 |
|                | 50,000| 82 | 695| 2.56e−06                      | 0.2301 |

Table 4 Numerical results of Problem 4.4

| Method         | N   | NI | FE | || F(x_k) || | CPU    |
|----------------|-----|----|----|-------------------------------|--------|
| 3TCGPB1        | 100 | 17 | 66 | 4.75e−06                      | 0.0054 |
|                | 1000| 37 | 198| 6.46e−06                      | 0.0202 |
|                | 10,000| 75 | 711| 7.36e−06                      | 0.5739 |
|                | 20,000| 99 | 1058| 7.60e−07                      | 1.6355 |
|                | 50,000| 145| 1842| 6.33e−06                      | 6.4960 |
| 3TCGPB2        | 100 | 19 | 69 | 5.35e−06                      | 0.0054 |
|                | 1000| 36 | 198| 4.19e−06                      | 0.0201 |
|                | 10,000| 75 | 712| 7.85e−06                      | 0.5980 |
|                | 20,000| 102| 1068| 4.53e−06                      | 1.6288 |
|                | 50,000| 151| 1853| 5.15e−06                      | 6.5939 |
| DFPB1          | 100 | 16 | 63 | 5.35e−06                      | 0.0058 |
|                | 1000| 37 | 198| 4.31e−06                      | 0.0200 |
|                | 10,000| 71 | 699| 9.00e−08                      | 0.5884 |
|                | 20,000| 97 | 1051| 1.20e−07                      | 1.7114 |
|                | 50,000| 144| 1836| 2.90e−07                      | 6.4733 |
| DFPB2          | 100 | 20 | 73 | 3.69e−06                      | 0.0054 |
|                | 1000| 32 | 188| 7.69e−06                      | 0.0189 |
|                | 10,000| 82 | 732| 3.58e−06                      | 0.6000 |
|                | 20,000| 109| 1087| 7.30e−06                      | 1.7348 |
|                | 50,000| 151| 1858| 5.96e−06                      | 6.5776 |

Problem 4.5 The mapping \( F(\cdot) \) is taken as \( F(x) = (F_1(x), F_2(x), F_3(x), \ldots, F_n(x))^T \), where

\[
F_1(x) = 2.5x_1 + x_2 - 1, \\
F_i(x) = x_{i-1} + 2.5x_i + x_{i+1} - 1, \text{ for } i = 2, 3, \ldots, n - 1, \\
F_n(x) = x_{n-1} + 2.5x_n - 1, \\
\]

and \( \Omega = \mathbb{R}^n \). Initial guess \( x_0 = (-1, -1, -1, \ldots, -1)^T \).

We present the results in Tables 1, 2, 3, 4, 5, where the dimension \( (N) \) of each problem is varied from 100 to 50,000. In each table, we present the results in terms of iterations (NI), function evaluations (FE), the optimal function value (\( || F(x_k) || \)) at termination as well as the CPU time. In all the test runs, the methods
Table 5 Numerical results of Problem 4.5

| Method      | N    | NI | FE  | $\|F(x_k)\|$ | CPU  |
|-------------|------|----|-----|--------------|------|
| 3TCGPB1     | 100  | 52 | 156 | 8.75e-06     | 0.0045 |
|             | 1000 | 60 | 174 | 8.69e-06     | 0.0088 |
|             | 10,000 | 80 | 238 | 9.67e-06     | 0.0773 |
|             | 20,000 | 94 | 288 | 9.23e-06     | 0.1483 |
|             | 50,000 | 88 | 326 | 8.65e-06     | 0.3762 |
| 3TCGPB2     | 100  | 60 | 172 | 8.78e-06     | 0.0045 |
|             | 1000 | 59 | 172 | 8.70e-06     | 0.0088 |
|             | 10,000 | 67 | 212 | 9.70e-06     | 0.0773 |
|             | 20,000 | 73 | 246 | 9.25e-06     | 0.1483 |
|             | 50,000 | 77 | 304 | 8.65e-06     | 0.3762 |
| DFPB1       | 100  | 58 | 168 | 8.76e-06     | 0.0051 |
|             | 1000 | 63 | 180 | 8.69e-06     | 0.0088 |
|             | 10,000 | 72 | 222 | 9.68e-06     | 0.0608 |
|             | 20,000 | 84 | 268 | 9.23e-06     | 0.2122 |
|             | 50,000 | 94 | 338 | 8.63e-06     | 0.4105 |
| DFPB1       | 100  | 59 | 170 | 8.71e-06     | 0.0048 |
|             | 1000 | 72 | 198 | 8.59e-06     | 0.0101 |
|             | 10,000 | 69 | 216 | 9.57e-06     | 0.0886 |
|             | 20,000 | 70 | 240 | 9.12e-06     | 0.1279 |
|             | 50,000 | 87 | 323 | 9.96e-06     | 0.3694 |

Fig. 1 Iterations performance profile

were successful in solving all the problems. A comparison of the methods from Tables 1, 2, 3, 4, 5, shows that the proposed methods are very competitive with the DFPB1 and the DFPB2 methods.

We further compare the methods using the performance profile tools suggested by Dolan and Moré [5]. We do this by plotting the performance profiles on NI, FE and CPU time. Figure 1 presents the performance profile on NI, Fig. 2 shows the performance profile on FE and finally Fig. 3 shows the performance profile on CPU time. It is clear from the figures that 3TCGPB2 performs much better than the other methods. However, overall the proposed methods are very much competitive and therefore promising.

5 Conclusion

In this work, two new derivative-free conjugate gradient projection methods for systems of large-scale nonlinear monotone equations were proposed. The proposed methods were motivated by the work of Ahookhosh et al. [1], Zhang et al. [8], Nakamura et al. [16] and Yuan [22]. The proposed methods were shown to satisfy the sufficient
descent condition and also their global convergence was established. The proposed methods were tested on a number of problems and compared with other competing methods and their numerical results indicate the methods to be efficient and very competitive.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Ahookhosh, M.; Amini, K.; Bahrami, S.: Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations. Numer. Algor. 64, 21–42 (2013)
2. Dai, Y.H.: Nonlinear conjugate gradient methods, Wiley Encyclopedia of Operations Research and Management Science, John Wiley and Sons (2011)
3. Dennis, J.E.; More, J.J.: Quasi-Newton method, motivation and theory. SIAM Rev. 19, 46–89 (1977)
4. Dirkse, S.P.; Ferris, M.C.: A collection of nonlinear mixed complementarity problems. Optim. Meth. Softw. 5, 319–345 (1995)
5. Dolan, E.D.; Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. 91, 201–213 (2002)
6. Goldstein, A.A.: Convex programming in Hilbert space. Amer. Math. Soc. 70, 709–710 (1964)
7. Grippo, L.; Sciandrone, M.: Nonmonotone derivative-free methods for nonlinear equations. Comput. Optim. Appl. 37, 297–328 (2007)
8. Hager, W.W.; Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. 16, 170–192 (2005)
9. Hu, Y.; Wei, Z.: A modified Liu-Storey conjugate gradient projection algorithm for nonlinear monotone equations. Int. Math. Forum. 9, 1767–1777 (2014)
10. Iusem, A.N.; Solodov, M.V.: Newton-type methods with generalized distance for constrained optimization. Optim. 41, 257–298 (1997)
11. Kanzow, C.; Yamashita, N.; Fukushima, M.: Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties. J. Comput. Appl. Math. 172, 375–397 (2004)
12. Li, Q.N.; Li, D.H.: A class of derivative-free methods for large-scale nonlinear monotone equations. IMA J. Numer. Anal. 31, 1625–1635 (2011)
13. Liu, J.K.; Li, S.J.: A projection method for convex constrained monotone nonlinear equations with applications. Comput. Math. Appl. 70, 2442–2453 (2015)
14. Liu, J.K.; Li, S.J.: Multivariate spectral DY-type projection method for nonlinear monotone equations. J. Ind. Manag. Optim. 13(1), 283–295 (2017)
15. Liu, J.K.; Li, S.J.: A three-term derivative-free projection method for systems of nonlinear monotone equations. Calcolo 53, 427–450 (2016)
16. Nakamura, W.; Narushima, Y.; Yabe, H.: Nonlinear conjugate gradient methods with sufficient descent properties for unconstrained optimization. J. Ind. Manag. Optim. 9(3), 595–619 (2013)
17. Solodov, M.V.; Svaiter, B.F.: A globally convergent inexact newton method for systems of monotone equations. Reformulation: Nonsmooth, pp. 355–369. Semismoothing methods, Springer, US, Piecewise Smooth (1998)
18. 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)
19. Sun, M.; Liu, J.: New hybrid conjugate gradient projection method for the convex constrained equations. Calcolo 53(3), 999–1016 (2016)
20. Tong, X.J.; Qi, L.: On the convergence of a trust-region method for solving constrained nonlinear equations with degenerate solution. J. Optim. Theory Appl. 123, 187–211 (2004)
21. 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, 33–46 (2007)
22. Yuan, G.: Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems. Optim. Lett. 3, 11–21 (2009)
23. Yuan, Y.: Subspace methods for large-scale nonlinear equations and nonlinear least squares. Optim. Eng. 10, 207–218 (2009)
24. Zhang, L.; Zhou, W.: Spectral gradient projection method for solving nonlinear monotone equations. J. Comput. Appl. Math. 196, 478–484 (2006)
25. Zheng, L.: A modified PRP projection method for nonlinear equations with convex constraints. Int. J. Pure. Appl. Math. 79, 87–96 (2012)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.