Arab. J. Math. (2020) 9:443–457
https://doi.org/10.1007/s40065-019-0264-6

Arabian Journal of Mathematics

M. Y. Waziri · K. Ahmed · J. Sabi'u

A Dai–Liao conjugate gradient method via modified secant equation for system of nonlinear equations

Received: 5 February 2019 / Accepted: 26 June 2019 / Published online: 8 July 2019

Abstract In this paper, we propose a Dai–Liao (DL) conjugate gradient method for solving large-scale system of nonlinear equations. The method incorporates an extended secant equation developed from modified secant equations proposed by Zhang et al. (J Optim Theory Appl 102(1):147–157, 1999) and Wei et al. (Appl Math Comput 175(2):1156–1188, 2006) in the DL approach. It is shown that the proposed scheme satisfies the sufficient descent condition. The global convergence of the method is established under mild conditions, and computational experiments on some benchmark test problems show that the method is efficient and robust.

Mathematics Subject Classification 90C30 · 65K05 · 90C53

1 Introduction

A typical system of nonlinear equations has the general form

$$F(x) = 0, \tag{1}$$

where $F : R^n \to R^n$ is a nonlinear mapping assumed to be continuously differentiable in a neighborhood of R^n. Systems of nonlinear equations play important role in sciences and engineering fields; therefore, solving (1) has become a subject of interest to researchers in the aforementioned areas. Numerous algorithms or schemes have been developed for solving these systems of equations. Notable among them are the Newton and quasi-Newton schemes [14,22,34,52], which converge rapidly from sufficiently good starting point. However, the requirement for computation and storage of the Jacobian matrix or an approximation of it at each iteration makes the two methods unattractive for large-scale nonlinear systems [51].

The ideal method for solving large-scale systems is the conjugate gradient (CG) method, which forms an important class of algorithms used in solving large-scale unconstrained optimization problems. The method is popular with mathematicians and engineers engaged in large-scale problems because of it low memory requirement and strong global convergence properties [19]. Generally, the nonlinear conjugate gradient method is used to solve large-scale problems in the following form;

$$\min f(x), \quad x \in R^n, \tag{2}$$

M. Y. Waziri (✉) · K. Ahmed
Department of Mathematical Sciences, Bayero University, Kano, Nigeria
E-mail: mywaziri.mth@buk.edu.ng

J. Sabi'u
Department of Mathematics, Northwest University, Kano, Nigeria

where $f : R^n \to R$ is a continuously differentiable function that is bounded from below and its gradient is available. The method generates a sequence of iterates x_k from an initial point $x_0 \in R^n$ using the iterative formula

$$x_{k+1} = x_k + s_k, \quad s_k = \alpha_k d_k, \quad k = 0, 1, \ldots, \tag{3}$$

where x_k is the current iterate, $\alpha_k > 0$ is a step length computed using suitable line search technique, and d_k is the CG search direction defined by

$$d_k = \begin{cases} -F_k, & \text{if} \quad k = 0, \\ -F_k + \beta_k d_{k-1}, & \text{if} \quad k \geq 1, \end{cases} \tag{4}$$

where β_k is a scalar known as the CG update parameter, and $F_k = \nabla f(x_k)$. It is worth noting that a crucial element in any CG algorithm is the formula definition of the update parameter β_k [4], which is why different CG algorithms corresponding to different choices of β_k in (4) have been proposed (see [8,10–14,17,33,50,51,53,65]).

Also, some of the CG methods for unconstrained optimization are not globally convergent, so efforts have been made by researchers to develop CG methods that are not only globally convergent but also are numerically efficient. These new methods are based on secant equations. For nonlinear conjugate gradient methods, the conjugacy condition is given by

$$d_k^T y_{k-1} = 0. \tag{5}$$

Perry [44] extended (5) by exploiting the following secant condition of quasi-Newton schemes:

$$B_k s_{k-1} = y_{k-1}, \tag{6}$$

and quasi-Newton search direction d_k given by

$$B_k d_k = -F_k, \tag{7}$$

where B_k is a square matrix, which approximates the Hessian $\nabla^2 f(x)$. By using (6) and (7), Perry gave an extension of (5) as:

$$d_k^T y_{k-1} = -F_k^T s_{k-1}, \tag{8}$$

and using (4), the Perry search direction is given as

$$d_k = \begin{cases} -F_k, & \text{if} \quad k = 0, \\ -P_k F_k = -F_k + \beta_k^P d_{k-1}, & \text{if} \quad k \geq 1, \end{cases} \tag{9}$$

where

$$B_k^P = \frac{(y_{k-1} - s_{k-1})^T}{s_{k-1}^T y_{k-1}} F_k, \tag{10}$$

and

$$P_k = I - \frac{s_{k-1}(y_{k-1} - s_{k-1})^T}{s_{k-1}^T y_{k-1}}. \tag{11}$$

Following Perry's approach, Dai and Liao [18] incorporated a nonnegative parameter t to propose the following extension of (8):

$$d_k^T y_{k-1} = -t F_k^T s_{k-1}. \tag{12}$$

It is noted that for $t = 0$, (12) reduces to (5), and if $t = 1$, we obtain Perry's condition (8). Consequently, by substituting (4) into (12), Dai and Liao [18] proposed the following CG update parameter:

$$B_k^{DL} = \frac{(y_{k-1} - t s_{k-1})^T F_k}{d_{k-1}^T y_{k-1}}, \quad t \geq 0. \tag{13}$$

Numerical results have shown that the DL method is effective; however, it is much dependent on the nonnegative parameter t for which there is no optimal value [4], and it may not necessarily generate descent directions [8]. That is, the method may not satisfy the descent condition

$$F_k^T d_k < 0, \quad \forall k, \tag{14}$$

or the sufficient descent condition, namely there exists a constant $\lambda > 0$ such that

$$F_k^T d_k \leq -\lambda \parallel F_k \parallel^2, \quad \forall k. \tag{15}$$

Based on the DL conjugacy condition (12), conjugate gradient methods have been proposed over the years using modified secant equations. For example, Babaie-Kafaki et al. [13] and Yabe and Takano [55] proposed CG methods by applying a revised form of the modified secant equation proposed by Zhang and Xu [63] and Zhang et al.[64] and the modified secant equation proposed by Li and Fukushima [36]. Li et al.[37] applied the modified secant equation proposed by Wei et al. [54], while Ford et al. [26] employed the multi-step quasi-Newton conditions proposed by Ford and Moghrabi [27,28]. CG methods based on modified secant equations have also been studied by Narushima and Yabe [57] and Reza Arazm et al. [7]. These methods have been found to be numerically efficient and globally convergent under suitable conditions, but like the DL method, they also fail to ensure sufficient descent.

Recently, by employing Perry's idea [44], efficient CG methods with descent directions have been proposed. Liu and Shang [39] proposed a Perry conjugate gradient method, which provides prototypes for developing other special form of the Perry method like the HS method and the DL method [18]. Liu and Xu [40] presented a new Perry CG method with sufficient descent properties, which is independent of any line search. Also, based on the self-scaling memoryless BFGS update, Andrei [6] proposed an accelerated adaptive class of Perry conjugate gradient algorithms, whose search direction is determined by symmetrization of the scaled Perry CG direction [44].

CG methods for systems of nonlinear equations are rare as most of the methods are for unconstrained optimization. However, over the years, the method has been extended to large-scale nonlinear systems of equations by researchers. Using a combination of the Polak–Ribieré–Polyak (PRP) conjugate gradient method for unconstrained optimization [45,47] and the hyperplane projection method of Solodov and Svaiter [48], Cheng [16] proposed a PRP-type method for systems of monotone equations. Yu [58,59] extended the PRP method [45] to solve large-scale nonlinear systems with monotone line search strategies, which are modifications of the Grippo–Lampariello–Lucidi [29] and Li–Fukushima [35] schemes. As a further research of the Perry's conjugate gradient method, Dai et al. [21] combined the modified Perry conjugate gradient method [41] and the hyperplane projection technique of Solodov and Svaiter [48] to propose a derivative-free method for solving large-scale nonlinear monotone equations. By combining the descent Dai–Liao CG method by Babaie-Kafaki and Ghanbari [54] and the projection method in [48], Abubakar and Pumam [2] proposed a descent Dai–Liao CG method for nonlinear equations. Numerical results show the method to be efficient. Based on the projection strategy [48], Liu and Feng [38] proposed a derivative-free iterative method for large-scale nonlinear monotone equations, which can be used to solved large-scale non-smooth problems due to its lower storage and derivative-free information. Abubakar and Kumam [1] proposed an improved three-term derivative-free method for solving large-scale nonlinear equations. The method is based on a modified HS method with the projection technique of Solodov and Svaiter [48]. Abubakar et al. [3] proposed a descent Dai–Liao CG method for solving nonlinear convex constraint monotone equations. The method is an extension of the method in [2]. By using a convex combination of two different positive spectral coefficients, Mohammed and Abubakar [42] proposed a combination of positive spectral gradient-like method and projection method for solving nonlinear monotone equations. Awwal et al. [43] proposed a hybrid spectral gradient algorithm for system of nonlinear monotone equations with convex constraints. The scheme is combination of a convex combination of two different positive spectral parameters and the projection technique.

Here, based on the work of Babaie-Kafaki and Ghanbari [9], and the Dai–Liao (DL) [18] approach, we propose a Dai–Liao conjugate gradient method for system of nonlinear equations by incorporating an extended secant equation in the classical DL update.

Throughout this work, we use $\parallel . \parallel$ to denote the Euclidean norm of vectors, $y_{k-1} = F_k - F_{k-1}$, $s_{k-1} = x_k - x_{k-1}$ and $F_k = F(x_k)$. We also assume that problem (1) is Lipschitz continuous and f in (2) is specified by

$$f(x) := \frac{1}{2} \parallel F(x) \parallel^2 . \tag{16}$$

The paper is organized as follows: in Sect. 2, we present details of the method. Convergence analysis is presented in Sect. 3. Numerical results of the method are presented in Sect. 4. Finally, conclusions are made in Sect. 5.

2 Proposed method and its algorithm

Following the Dai–Liao approach, Babaie-Kafaki and Ghanbari [9] proposed the following extension of the PRP update parameter

$$\beta_k^{\text{EPRP}} = \beta_k^{\text{PRP}} - t\frac{F_k^T d_{k-1}}{\| F_{k-1} \|^2}, \tag{17}$$

where β_k^{PRP} is the classical PRP parameter and t is a nonnegative parameter, whose values were determined by carrying out eigenvalue analysis. Motivated by this, and employing similar approach, we propose a modification of the classical DL update parameter. In what follows, we suggest an extension of some previously modified secant equations.

By expanding (6), Zhang et al. [64] proposed the following modified secant equation

$$B_k s_{k-1} = \hat{y}_{k-1}, \quad \hat{y}_{k-1} = y_{k-1} + \left(\frac{\theta_{k-1}}{s_{k-1}^T \mu_{k-1}}\right)\mu_{k-1}, \tag{18}$$

where

$$\theta_{k-1} = 6\left(f_{k-1} - f_k\right) + 3s_{k-1}^T\left(F_{k-1} + F_k\right), \tag{19}$$

where $\mu_{k-1} \in R^n$ is a vector parameter such that $s_{k-1}^T \mu_{k-1} \neq 0$ (see [64]).

Similarly, Wei et al. [54] gave the following modified secant equation

$$B_k s_{k-1} = \bar{y}_{k-1}, \quad \bar{y}_{k-1} = y_{k-1} + \left(\frac{\vartheta_{k-1}}{s_{k-1}^T \mu_{k-1}}\right)\mu_{k-1}, \tag{20}$$

with

$$\vartheta_{k-1} = 2\left(f_{k-1} - f_k\right) + s_{k-1}^T\left(F_{k-1} + F_k\right), \tag{21}$$

where $\mu_{k-1} \in R^n$ is a vector parameter such that $s_{k-1}^T \mu_{k-1} \neq 0$ (see [60]). Also, in (18) and (20), the vector parameter $\mu_{k-1} = s_{k-1}$ [55].

Here, we propose the following secant equation as an extension of (6), (18), and (20):

$$B_k s_{k-1} = u_{k-1} = y_{k-1} + 2\phi\frac{\vartheta_{k-1}}{s_{k-1}^T \mu_{k-1}}\mu_{k-1}, \tag{22}$$

where ϕ is a nonnegative parameter, ϑ_{k-1} is defined by (21) and $s_{k-1}^T \mu_{k-1} \neq 0$. We observe that for $\phi = 0$, (22) becomes the standard secant equation defined by (6), and if $\phi = \frac{3}{2}$, (22) reduces to (19). Also, for $\phi = \frac{1}{2}$, we see that (22) reduces to the modified secant equation proposed by Zhang et al. [64]. Substituting u_{k-1} in (22) for y_{k-1} in (13), we obtain the following version of the DL update parameter:

$$\bar{\beta}_k^{\text{ADL}} = \frac{\left(u_{k-1} - ts_{k-1}\right)^T F_k}{d_{k-1}^T u_{k-1}}, \quad t \geq 0. \tag{23}$$

Observe that, in general, the denominator, $d_{k-1}^T u_{k-1}$ may not be nonzero since ϑ_{k-1} as defined in (22) may be non-positive. Therefore, we redefine u_{k-1} and obtain its revised form as

$$z_{k-1} = y_{k-1} + 2\phi\frac{\max\left\{\vartheta_{k-1}, 0\right\}}{s_{k-1}^T \mu_{k-1}}\mu_{k-1}. \tag{24}$$

Consequently, we get the revised form of (23) as

$$\hat{\beta}_k^{\text{ADL}} = \frac{z_{k-1}^T F_k}{d_{k-1}^T z_{k-1}} - t\frac{s_{k-1}^T F_k}{d_{k-1}^T z_{k-1}}. \tag{25}$$

Andrei [4] noted that the parameter t has no optimal choice and so, to obtain descent directions for our proposed method, we proceed to obtain appropriate values for t. From (4), and after some algebra, our search direction becomes:

$$d_k = -F_k + \left(\frac{s_{k-1} z_{k-1}^T - ts_{k-1}s_{k-1}^T}{s_{k-1}^T z_{k-1}}\right)F_k. \tag{26}$$

Following Perry's approach [44], search direction of our proposed method can be written as

$$d_k = -H_k F_k, \quad k \geq 1, \tag{27}$$

where H_k, called the search direction matrix is given by

$$H_k = I - \frac{s_{k-1} z_{k-1}^T}{s_{k-1}^T z_{k-1}} + t \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T z_{k-1}}, \tag{28}$$

and z_{k-1} is as defined by (24). And from (27) we can write

$$d_k^T F_k = -F_k^T H_k^T F_k = d_k^T F_k = -F_k^T \frac{H_k^T + H_k}{2} F_k, \tag{29}$$

where

$$\begin{aligned} \bar{H}_k &= \frac{H_k^T + H_k}{2} \\ &= I - \frac{1}{2} \frac{s_{k-1} z_{k-1}^T + z_{k-1} s_{k-1}^T}{s_{k-1}^T z_{k-1}} + t \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T z_{k-1}} \end{aligned} \tag{30}$$

Proposition 2.1 *The matrix $\bar{H}_k$ defined by (30) is a symmetric matrix.*

Proof Using direct computation, we see that $\bar{H}_k = \bar{H}_k^T$. Hence, $\bar{H}_k$ is symmetric.

And so, to analyze the descent property of our method, we need to find eigenvalues of $\bar{H}_k$ and their structure. □

Theorem 2.2 *Let the matrix $\bar{H}_k$ be defined by (30). Then, the eigenvalues of $\bar{H}_k$ consist of 1 with ($n-2$ multiplicity), λ_k^+ and λ_k^-, where*

$$\lambda_k^+ = \frac{1}{2}\left[(1 + a_k) + \sqrt{(a_k - 1)^2 + b_k - 1}\right] \tag{31}$$

$$\lambda_k^- = \frac{1}{2}\left[(1 + a_k) - \sqrt{(a_k - 1)^2 + b_k - 1}\right] \tag{32}$$

and $a_k = t \frac{\|s_{k-1}\|^2}{s_{k-1}^T z_{k-1}}, \quad b_k = \frac{\|s_{k-1}\|^2 \|z_{k-1}\|^2}{(s_{k-1}^T z_{k-1})^2}$.

Furthermore, all eigenvalues of $\bar{H}_k$ are positive real numbers.

Proof Since $d_{k-1}^T z_{k-1} \neq 0$, then $s_{k-1}^T z_{k-1} \neq 0$. And so, $s_{k-1} \neq 0$ and $z_{k-1} \neq 0$, which implies that the vectors s_{k-1} and z_{k-1} are nonzero vectors. Suppose V is the vector space spanned by $\{s_{k-1}, z_{k-1}\}$. Then $dim\,(V) \leq 2$ and $dim\left(V^{\perp}\right) \geq n-2$, where $V^{\perp}$ is the orthogonal complement of V. Therefore, there exists a set of mutually orthogonal vectors $\{\tau_{k-1}^i\}_{i=1}^{n-2} \subset V^{\perp}$ satisfying

$$s_{k-1}^T \tau_{k-1}^i = z_{k-1}^T \tau_{k-1}^i = 0. \tag{33}$$

By multiplying both sides of (30) by τ_{k-1}^i, we obtain

$$\bar{H}_k \tau_{k-1}^i = \tau_{k-1}^i, \quad i = 1, \ldots, n-2, \tag{34}$$

which can be viewed as an eigenvector equation. So, τ_{k-1}^i, for $i = 1, \ldots, n-2$ are the eigenvectors of $\bar{H}_k$ with eigenvalue 1 each. Let λ_k^+ and λ_k^- be the remaining two eigenvalues, respectively. Observe that (30) can be written as

$$\bar{H}_k = I - \frac{s_{k-1}(z_{k-1} - 2t s_{k-1})^T}{2 s_{k-1}^T z_{k-1}} - \frac{z_{k-1} s_{k-1}^T}{2 s_{k-1}^T z_{k-1}}. \tag{35}$$

Clearly, $\bar{H}_k$ represents a rank-two update, so from the fundamental algebra formula (see inequality (1.2.70)) of [49]

$$det(I + u_1 u_2^T + u_3 u_4^T) = (1 + u_1^T u_2)(1 + u_3^T u_4) - (u_1^T u_4)(u_2^T u_3), \tag{36}$$

where

$$u_1 = -\frac{s_{k-1}}{2s_{k-1}^T z_{k-1}}, \quad u_2 = (z_{k-1} - 2ts_{k-1}), \quad u_3 = -\frac{z_{k-1}}{2s_{k-1}^T z_{k-1}}, \quad u_4 = s_{k-1}$$

$$det\left(\bar{H}_k\right) = \frac{1}{4} + t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}} - \frac{1}{4}\frac{\| s_{k-1} \|^2 \| z_{k-1} \|^2}{(s_{k-1}^T z_{k-1})^2}. \tag{37}$$

Since sum of the eigenvalues of a square symmetric matrix equals to its trace, from (30), we have

$$\begin{aligned} \text{trace}\left(\bar{H}_k\right) &= n - 1 + t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}} \\ &= \underbrace{1 + \cdots + 1}_{(n-2)\text{times}} + \lambda_k^+ + \lambda_k^-, \end{aligned} \tag{38}$$

for which we obtain

$$\lambda_k^+ + \lambda_k^- = 1 + t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}}. \tag{39}$$

Using the relationship between trace and determinant of a matrix and its eigenvalues, we can obtain λ_k^+ and λ_k^- as roots of the following quadratic polynomial:

$$\lambda^2 - \left(1 + t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}}\right)\lambda + \frac{1}{4} + t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}} - \frac{1}{4}\frac{\| s_{k-1} \|^2 \| z_{k-1} \|^2}{(s_{k-1}^T z_{k-1})^2} = 0. \tag{40}$$

So, the remaining two eigenvalues are obtained from (40). And applying the quadratic formula with some rearrangements, we obtain

$$\lambda_k^{\pm} = \frac{1}{2}\left[1 + t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}} \pm \sqrt{\left(t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}} - 1\right)^2 + \frac{\| s_{k-1} \|^2 \| z_{k-1} \|^2}{(s_{k-1}^T z_{k-1})^2} - 1}\right] \tag{41}$$

We can write (41) as

$$\lambda_k^{\pm} = \frac{1}{2}\left[(1 + a_k) \pm \sqrt{(a_k - 1)^2 + b_k - 1}\right], \tag{42}$$

which proves (31) and (32).

To obtain λ_k^+ and λ_k^- as real numbers, we must have $\Delta = (a_k - 1)^2 + b_{k-1} \geq 0$.

From Cauchy inequality, $b_k = \frac{\|s_{k-1}\|^2 \|z_{k-1}\|^2}{(s_{k-1}^T z_{k-1})^2} \geq 1$, so, $\Delta > 0$. Consequently, both eigenvalues are real numbers and $\lambda_k^+ > 0$ since $(1 + a_k)$ is nonnegative. And to obtain $\lambda_k^- > 0$, the following must be satisfied:

$$\frac{1}{2}\left[1 + t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}} - \sqrt{\left(t\frac{\| s_{k-1} \|^2}{s_{k-1}^T z_{k-1}} - 1\right)^2 + \frac{\| s_{k-1} \|^2 \| z_{k-1} \|^2}{(s_{k-1}^T z_{k-1})^2} - 1}\right] > 0. \tag{43}$$

After some algebra, we obtain the following estimation for the parameter t, which satisfies (43):

$$t > \frac{1}{4}\left(\frac{\| z_{k-1} \|^2}{s_{k-1}^T z_{k-1}} - \frac{s_{k-1}^T z_{k-1}}{\| s_{k-1} \|^2}\right). \tag{44}$$

So, $\lambda_k^- > 0$ if (44) is satisfied. In addition, for t satisfying (44), $\bar{H}_k$ is nonsingular.

Therefore, all the eigenvalues of the symmetric matrix $\bar{H}_k$ are positive real numbers, which ensures that it is a positive-definite matrix. Moreover, using (42) and (44), we obtain the following estimation for λ_k^+ and λ_k^-:

$$\lambda_k^+ \geq \left(\frac{3(s_{k-1}^T z_{k-1})^2 + \| z_{k-1} \|^2 \| s_{k-1} \|^2}{(s_{k-1}^T z_{k-1})^2}\right), \quad \lambda_k^- > 0. \tag{45}$$

And the proof is complete. Hence, from (29), we have

$$d_k^T F_k = -F_k^T \bar{H}_k F_k \leq -\lambda_k^- \parallel F_k \parallel^2 < 0, \tag{46}$$

which shows that the descent condition is satisfied. We, therefore, propose the following formula for the parameter t in the modified DL method:

$$t^{\text{ADL}} = \xi \frac{\parallel z_{k-1} \parallel^2}{s_{k-1}^T z_{k-1}} - \gamma \frac{s_{k-1}^T z_{k-1}}{\parallel s_{k-1} \parallel^2}, \tag{47}$$

where $\xi > \frac{1}{4}$ and $\gamma < \frac{1}{4}$. □

Remark 2.3 Since the DL parameter t is nonnegative, we restrict the values of the parameter γ in (47) to be negative so as to avoid a numerically unreasonable approximation [32]. So, based on the above remark, we can write the modified DL update parameter as

$$\beta_k^{\text{ADL}} = \frac{F_k^T z_{k-1}}{d_{k-1}^T z_{k-1}} - t^{\text{ADL}} \frac{F_k^T s_{k-1}}{d_{k-1}^T z_{k-1}}, \tag{48}$$

with $\xi \geq \frac{1}{4}$ and $\gamma < 0$ satisfying (47) and guaranteeing the descent condition. We also write the search direction for the proposed method as

$$d_k^{\text{ADL}} = -F_k + \left(\frac{(z_{k-1} - t_k^{\text{ADL}} s_{k-1})^T F_k}{d_{k-1}^T z_{k-1}} \right) d_{k-1}. \tag{49}$$

We use the derivative-free line search proposed by Li and Fukushima [34] to compute our step length α_k.

Let $\sigma_1 > 0$, $\sigma_2 > 0$ and $r \in (0, 1)$ be constants and let $\{\eta_k\}$ be a given positive sequence such that

$$\sum_{k=0}^{\infty} \eta_k < \eta < \infty, \tag{50}$$

and

$$\parallel F_{k+1} \parallel^2 - \parallel F_k \parallel^2 \leq -\sigma_1 \parallel \alpha_k F_k \parallel^2 - \sigma_2 \parallel \alpha_k d_k \parallel^2 + \eta_k \parallel F_k \parallel^2 . \tag{51}$$

Let i_k be the smallest non-negative integer i such that (51) holds for $\alpha = r^i$. Let $\alpha_k = r^{i_k}$.

Now, we describe the algorithm of the proposed method as follows:

Algorithm 2.4 A Dai–Liao CG method (ADLCG)

Step 1 Given $\varepsilon > 0$, choose an initial point $x_0 \in R^n$, a positive sequence $\{\eta_k\}$ satisfying (50), and constants $r \in (0, 1)$, $\sigma_1, \sigma_2 > 0$, $\xi \geq \frac{1}{4}$, $\gamma < 0$. Compute $d_0 = -F_0$ and set $k = 0$.
Step 2 Compute $F(x_k)$. If $\parallel F(x_k) \parallel \leq \varepsilon$, stop. Otherwise, compute the search direction d_k by (49).
Step 3 Compute α_k via the line search in (51).
Step 4 Set $x_{k+1} = x_k + \alpha_k d_k$.
Step 5 Set $k := k + 1$ and go to **Step 2**.

3 Convergence analysis

The following assumptions are required to analyze the convergence of the ADLCG algorithm.

Assumption 3.1 The level set

$$\Omega = \{x | F(x) \leq F(x_0)\} \tag{52}$$

is bounded.

Assumption 3.2 (1) The solution set of problem (1) is not empty.
(2) F is continuously differentiable on an open convex set Φ_1 containing Φ.

(3) F is Lipschitz continuous in some neighborhood N of Φ; namely, there exists a positive constant $L > 0$ such that,

$$\| F(x) - F(y) \| \leq L \| x - y \|, \tag{53}$$

for all $x, y \in N$.

Assumption (3.1) and condition (3) imply that there exists a positive constant ω such that

$$\| F(x_k) \| \leq \omega, \tag{54}$$

for all $x \in \Phi$, (see Proposition 1.3 of [13]).

(4) The Jacobian of F is bounded, symmetric and positive-definite on Φ_1, which implies that there exist constants $m_2 \geq m_1 > 0$ such that

$$\| F'(x) \| \leq m_2, \quad \forall x \in \Phi_1, \tag{55}$$

and

$$m_1 \| d \|^2 \leq d^T F'(x) d, \quad \forall x \in \Phi_1, d \in R^n. \tag{56}$$

Lemma 3.3 *Let $\{x_k\}$ be generated by the Algorithm 2.4. Then d_k is a descent direction for $F(x_k)$ at x_k. i.e.,*

$$F(x)^T d_k < 0. \tag{57}$$

Proof By (46), the Lemma is true and we can deduce that the norm function $f(x_k)$ is a descent along the direction d_k. i.e., $\| F(x_{k+1}) \| \leq \| F(x_k) \|$ is true $\forall k$. □

Lemma 3.4 *Suppose Assumptions 3.1 and 3.2 hold. Let $\{x_k\}$ be generated by the Algorithm 2.4. Then $\{x_k\} \subset \Omega$. Moreover, $\| F_k \|\}$ converges.*

Proof By Lemma 3.3, we have $\{\| F(x_{k+1}) \| \leq \| F(x_k) \|$. So, by Lemma 3.3 in [20], we conclude that $\{\| F_k \|\}$ converges. Moreover, for all k, we have

$$\| F(x_{k+1}) \| \leq \| F(x_k) \| \leq \| F(x_{k-1}) \| \cdots \leq \| F(x_0) \| . \tag{58}$$

This implies that $\{x_k\} \subset \Omega$ □

Lemma 3.5 *Suppose Assumption 3.1 and 3.2 hold. Let $\{x_k\}$ be generated by the Algorithm 2.4. Then*

$$\lim_{k \to \infty} \| \alpha_k d_k \| = \lim_{k \to \infty} \| s_k \| = 0, \tag{59}$$

and

$$\lim_{k \to \infty} \| \alpha_k F(x_k) \| = 0. \tag{60}$$

Proof From the line search (51) and for all $k > 0$, we obtain

$$\begin{aligned} \sigma_2 \| \alpha_k d_k \|^2 &\leq \sigma_1 \| \alpha_k F_k \|^2 + \sigma_2 \| \alpha_k d_k \|^2 \\ &\leq \| F_k \|^2 - \| F_{k+1} \|^2 + \eta_k \| F_k \|^2 . \end{aligned} \tag{61}$$

And by summing up the above k inequality, we obtain

$$\begin{aligned} \sigma_2 \sum_{i=0}^{k} \| \alpha_k d_k \|^2 &\leq \sum_{i=0}^{k} \left(\| F(x_i) \|^2 - \| F(x_{i+1}) \|^2 \right) + \sum_{i=0}^{k} \eta_i \| F(x_i) \|^2 \\ &= \| F(x_0) \|^2 - \| F(x_{k+1}) \|^2 + \sum_{i=0}^{k} \eta_i \| F(x_i) \|^2 \\ &\leq \| F(x_0) \|^2 + \| F(x_0) \|^2 \sum_{i=0}^{k} \eta_i \\ &\leq \| F(x_0) \|^2 + \| F(x_0) \|^2 \sum_{i=0}^{\infty} \eta_i . \end{aligned} \tag{62}$$

Therefore, by (52) and since $\{\eta_i\}$ satisfies (50), then the series $\sum_{i=0}^{k} \| \alpha_k d_k \|^2$ is convergent, which implies that (59) holds. Using the same argument as above, with $\sigma_1 \| \alpha_k F(x_k) \|^2$ on the left-hand sides, we obtain (60). □

Lemma 3.6 [62] *Suppose Assumptions 3.1 and 3.2 hold and* $\{x_k\}$ *be generated by Algorithm 2.4. Then, there exists a constant* $m > 0$ *such that,*

$$y_k^T s_k \geq m \parallel s_k \parallel^2 > 0, \quad \forall k \geq 1. \tag{63}$$

Proof By mean-value theorem, we have

$$y_k^T s_k = s_k^T \left(F\left(x_{k+1}\right) - F\left(x_k\right)\right) = s_k^T F'', \tag{64}$$

where $\varphi = \lambda x_k + (1-\lambda) x_{k+1}$, for some $\lambda \in (0, 1)$. We obtain the last inequality from (56). Letting $m_1 = m$, the proof is established. □

Lemma 3.7 *Suppose Assumptions 3.1 and 3.2 hold. Let the sequence* $\{x_k\}$ *be generated by Algorithm 2.4 with update parameter* β_k^{ADL}*. Then, there exists* $M > 0$ *such that*

$$\parallel d_k^{\text{ADL}} \parallel \leq M, \quad \forall k. \tag{65}$$

Proof Using (24) and (64), we get

$$s_{k-1}^T z_{k-1} = s_{k-1}^T y_{k-1} + 2\phi \frac{\max\{\vartheta_{k-1}, 0\}}{s_{k-1}^T \mu_{k-1}} s_{k-1}^T \mu_{k-1} \geq s_{k-1}^T y_{k-1} \geq m \parallel s_{k-1} \parallel^2. \tag{66}$$

Applying the mean-value theorem, we have

$$\begin{aligned} |\vartheta_{k-1}| &= |2\left(f_k - f_{k+1}\right) + \left(F_{k-1} + F_k\right)^T s_{k-1}| \\ &= |(-2\nabla f(\varphi) + \nabla f(x_k) + \nabla f(x_{k+1}))^T s_{k-1}|, \end{aligned} \tag{67}$$

where $\varphi = \lambda x_k + (1-\lambda) x_{k+1}$, for some $\lambda \in (0, 1)$.

Hence from (53), we have

$$\begin{aligned} |\vartheta_{k-1}| &\leq (\parallel \nabla f(x_k) - \nabla f(\varphi) \parallel + \parallel \nabla f(x_{k+1} - \nabla f(\varphi) \parallel) \parallel s_{k-1} \parallel \\ &\leq (L(1-\lambda) \parallel s_{k-1} \parallel + L\lambda \parallel s_{k-1} \parallel) \parallel s_{k-1} \parallel \\ &= L \parallel s_{k-1} \parallel^2. \end{aligned} \tag{68}$$

Utilizing (24), (53), (68), and setting $\mu_{k-1} = s_{k-1}$, we obtain

$$\begin{aligned} \parallel z_{k-1} \parallel &\leq \parallel y_{k-1} \parallel + 2\phi \frac{|\vartheta_{k-1}|}{|s_{k-1}^T s_{k-1}|} \parallel s_{k-1} \parallel \\ &\leq L \parallel s_{k-1} \parallel + 2\phi L \frac{\parallel s_{k-1} \parallel^2}{\parallel s_{k-1} \parallel^2} \parallel s_{k-1} \parallel \\ &= (L + 2\phi L) \parallel s_{k-1} \parallel. \end{aligned} \tag{69}$$

And using (47), (53) and (69), we get

$$\begin{aligned} \left|t^{\text{ADL}}\right| &= \left|\xi \frac{\parallel z_{k-1} \parallel^2}{s_{k-1}^T z_{k-1}} - \gamma \frac{s_{k-1}^T z_{k-1}}{\parallel s_{k-1} \parallel^2}\right| \\ &\leq \left|\xi \frac{\parallel z_{k-1} \parallel^2}{s_{k-1}^T z_{k-1}}\right| + \left|\gamma \frac{s_{k-1}^T z_{k-1}}{\parallel s_{k-1} \parallel^2}\right| \\ &\leq \xi \frac{((L + 2\phi L) \parallel s_{k-1} \parallel)^2}{m \parallel s_{k-1} \parallel^2} + |\gamma| \frac{m \parallel s_{k-1} \parallel^2}{\parallel s_{k-1} \parallel^2} \\ &= \xi \frac{(L + 2\phi L)^2}{m} + m|\gamma|. \end{aligned} \tag{70}$$

By utilizing (4), (47), (48), (69) and (70) we obtain,

$$
\begin{aligned}
\|d_k^{\mathrm{ADL}}\| &= \| - F(x_k) + \beta_k^{\mathrm{ADL}} dk - 1\| \\
&\le \|F(x_k)\| + |\beta_k^{\mathrm{ADL}}| \|dk - 1\| \\
&= \|F(x_k)\| + \frac{\|F(x_k)\|\|z_{k-1}\|}{s_{k-1}^T z_{k-1}} \|s_{k-1}\| + |t^{\mathrm{ADL}}| \frac{\|F(x_k)\|\|s_{k-1}\|}{s_{k-1}^T z_{k-1}} \|s_{k-1}\| \\
&\le \|F(x_k)\| + \frac{\|F(x_k)\|(L+2\phi L)}{m} + \left(\xi \frac{(L+2\phi L)^2}{m} + m|\gamma|\right) \frac{\|F(x_k)\|}{m} \\
&= \left(1 + \frac{(L+2\phi L)}{m} + \left(\xi \frac{(L+2\phi L)^2}{m^2} + |\gamma|\right)\right) \|F(x_k)\| \\
&= \frac{(m^2 + m(L+2\phi L) + ((L+2\phi L)^2 \xi + |\gamma|))\|F(x_k)\|}{m^2} \\
&= \frac{c_1 \|F(x_k)\|}{m^2}, \qquad (71)
\end{aligned}
$$

where $c_1 = (m^2 + m\,(L + 2\phi L) + \left((L + 2\phi L)^2 \xi + |\gamma|\right)\big)$.

Setting $M := \frac{c_1 \|F(x_k)\|}{m^2}$, we obtain the required result.

In the next, we prove the global convergence of the **ADLCG** method. □

Theorem 3.8 *Suppose Assumption 3.1 and 3.2 hold and that the sequence $\{x_k\}$ is generated by Algorithm 2.4. Also, assume that for all $k > 0$*

$$\alpha_k \ge c \frac{|F(x_k)^T d_k|}{\| d_k \|^2}, \qquad (72)$$

where c is some positive constant. Then, $\{x_k\}$ converges globally to a solution of problem (1); i.e,

$$\lim_{k\to\infty} \| F(x_k) \| = 0. \qquad (73)$$

Proof By (59) and the boundedness of $\{\| d_k \|\}$, we have

$$\lim_{k\to\infty} \alpha_k \| d_k \|^2 = 0. \qquad (74)$$

From (72) and (74), we have

$$\lim_{k\to\infty} |F(x_k)^T d_k| = 0. \qquad (75)$$

On the other hand, from (46), and (45), we have

$$
\begin{aligned}
F(x_k)^T d_k &= -\lambda_k^- \| F(x_k) \|^2 \\
\| F(x_k) \|^2 &= \left\| -\frac{1}{\lambda_k^-} F(x_k)^T d_k \right\| \\
&\le |F(x_k)^T d_k| \left| \frac{1}{\lambda_k^-} \right|. \qquad (76)
\end{aligned}
$$

But from (45), we have

$$\lambda_k^+ > \lambda_k^- > 0, \quad \forall k. \qquad (77)$$

Thus, from (76) and applying the sandwich theorem, we obtain

$$0 \le \| F(x_k) \|^2 \le |F(x_k)^T d_k| \left(\frac{1}{\lambda_k^-} \right) \to 0. \qquad (78)$$

Therefore,

$$\lim_{k\to\infty} \| F(x_k) \| = 0. \qquad (79)$$

And the proof is completed. □

4 Numerical result

In this section, we test the efficiency and robustness of our proposed approach using the following method in the literature:

A new derivative-free conjugate gradient method for solving large-scale nonlinear systems of equations (**NDFCG**) [24]. All the codes used were written in MATLAB R2014a environment and run on a personal computer (2.20GHZ CPU, 8GB RAM). Also, the two algorithms used in the experiment were implemented with the same line search procedure, and the parameters are set to $\sigma_1 = \sigma_2 = 10^{-4}$, $\alpha_0 = 0.1$, $r = 0.2$ and $\eta_k = \frac{1}{(k+1)^2}$. In addition, we set $\xi = 0.5$, $\gamma = -0.5$ and $\mu_{k-1} = s_{k-1}$ for the ADLCG method. Also, the iteration was set to terminate if it exceeds 2000 or the inequality $\| F_k \| \leq 10^{-10}$ is satisfied (Table 1).

The two algorithms were tested using the following test problems with various sizes:

Problem 4.1 [2] The elements of the function $F(x)$ are given by:

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

Problem 4.2 [2] The elements of the function $F(x)$ are given by:

$$F_i(x) = \log(x_i + 1) - \frac{x_i}{n}, \qquad i = 2, \ldots, n.$$

Problem 4.3 [67] The elements of the function $F(x)$ are given by:

$$\begin{aligned} &F_1(x) = 2x_1 + \sin(x_1) - 1, \\ &F_i(x) = -2x_{i-1} + 2x_i + \sin(x_i) - 1, \qquad i = 2, \ldots, n-1, \\ &F_n(x) = 2x_n + \sin(x_n) - 1. \end{aligned}$$

Problem 4.4 [56] The elements of the function $F(x)$ are given by:

$$F_i(x) = x_i - \frac{1}{n}x_i^2 + \frac{1}{n}\sum_{i=1}^{n} x_i + i, \qquad i = 1, 2, \ldots, n..$$

Problem 4.5 [38] The elements of the function $F(x)$ are given by:

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

Problem 4.6 [51] The function $F(x)$ is given by

$$F(x) = Ax + b_1,$$

where $b_1 = (e_1^x - 1, \ldots, e_n^x - 1)^T$, and

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

Problem 4.7 [61] The elements of the function $F(x)$ are given by:

$$\begin{aligned} &F_i(x) = \sqrt{10^{-5}}\,(x_i - 1), \\ &F_n(x) = \frac{1}{4n}\sum_{j=1}^{n} x_j^2 - \frac{1}{4}, \quad i = 2, 3, \ldots, n-1. \end{aligned}$$

Problem 4.8 [2] The elements of the function $F(x)$ are given by:

$$F_i(x) = e^{x_i} - 1, \quad i = 1, 2, \ldots, n.$$

Table 1 Initial starting points used for the test problems

Initial point	Value
x1	$(1, 1, \ldots, 1)^T$
x2	$(-\frac{1}{4}, \frac{1}{4}, \ldots, (-1)^n \frac{1}{4})^T$
x3	$(\frac{1}{n}, \frac{2}{n}, \ldots, 1)^T$
x4	$(1, \frac{1}{4}, \ldots \frac{1}{n^2})^T$
x5	$(\frac{1}{2}, \frac{1}{2^2}, \ldots, \frac{1}{2^n})^T$
x6	$(\frac{1}{3}, \frac{1}{3^2}, \ldots, \frac{1}{3^n})^T$
x7	$(1, \frac{1}{2}, \frac{1}{n})^T$
x8	$(1, 0, , (\frac{2}{n}) - 1)^T$
x9	$(1, \frac{2^2}{2^3}, \frac{n^2}{n^3})^T$

Table 2 Number of problems and percentage for which each method is a winner with respect to iterations and CPU time

Method	Iter	Percentage	CPU time	Percentage
ADLCG	76	95	58	72.5
NDFCG	3	3.75	22	27.5
Undecided	1	1.25	0	0

Problem 4.9 [2] The elements of the function $F(x)$ are given by:

$$\begin{aligned} F_1(x) &= x_1\left(x_1^2 + x_2^2\right) - 1, \\ F_i(x) &= x_i\left(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2\right) - 1, \quad i = 2, 3, \ldots, n-1, \\ F_n(x) &= x_n\left(x_{n-1}^2 + x_n^2\right). \end{aligned}$$

Problem 4.10 [2] The elements of the function $F(x)$ are given by:

$$\begin{aligned} F_1(x) &= x_1 - e\left(\cos\frac{x_1 + x_2}{n+1}\right), \\ F_i(x) &= x_i - e\left(\cos\frac{x_{1-1} + xi + x_{i+1}}{n+1}\right), \quad i = 2, 3, \ldots, n-1, \\ F_n(x) &= x_n - e\left(\cos\frac{x_{n-1} + x_n}{n+1}\right). \end{aligned}$$

Using the performance profile of Dolan and More [23], we generate Figs. 1 and 2 to show the performance and efficiency of each of the two methods. To better illustrate the performance of the two methods, a summary of the results is presented in Table 2. The summarized data show the number of problems for which each method is a winner in terms of number of iterations and CPU time, respectively. The corresponding percentages of number of problems solved are also indicated

In Figs. 1 and 2, we observed that the curve representing the ADLCG method is above the curve representing the NDFCG method. This is a measure of the efficiency of the ADLCG method compared to the NDFCG scheme.

Similarly, the summary reported in Table 2 indicated that the ADLCG method is a winner with respect to number of iterations and CPU time. The table shows that the ADLCG method solves 95% (76 out of 80) of the problems with less number of iterations compared to the NDFCG method, which solves only 3.75% (3 out of 80). The summarized result also shows that both methods solve 1 problem with the same number of iteration, which translates to 1.25% and is reported as undecided. Also, the summary indicated that the ADLCG method outperforms the NDFCG scheme as it solves 72.5% (58 out of 80) of the problems with less CPU time compared to 27.5% (22 out of 80) solved by the NDFCG. Therefore, it is clear from Figs. 1 and 2 and the summarized result in Table 2 that our method is more efficient than the NDFCG method and better for large-scale nonlinear systems.

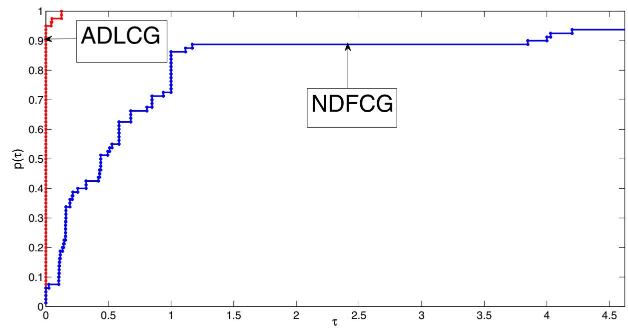

Fig. 1 Performance profile for number of iterations

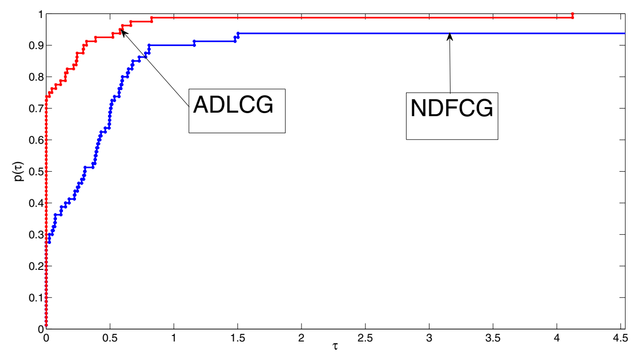

Fig. 2 Performance profile for the CPU time

5 Conclusion

In this work, we proposed a Dai–Liao conjugate gradient method via modified secant equation for systems of nonlinear equations. This was achieved by finding appropriate values for the nonnegative parameter in the DL method using of an extended secant equation developed from the work of Zhang et al. [64] and Wei et al [54]. Numerical comparisons with some existing methods and Global convergence show that the method is efficient.

References

1. Abubakar, A.B.; Kumam, P.: An improved three-term derivative-free method for solving nonlinear equations. Comput. Appl. Math. **37**(5), 6760–6773 (2018)
2. Abubakar, A.B.; Kumam, P.: A descent Dai–Liao conjugate gradient method for nonlinear equations. Numer. Algorithms (2018). https://doi.org/10.1007/s11075-018-0541-z
3. Abubakar, A.B.; Kumam, P.; Auwal, A.M.: A descent Dai–Liao projection method for convex constrained nonlinear monotone equations with applications. Thai J. Math. **17**(1), 128–152 (2018)
4. Andrei, N.: Open problems in conjugate gradient algorithms for unconstrained optimization. Bull. Malays. Math. Sci. Soc. **34**(2), 319–330 (2011)
5. Andrei, N.: An adaptive conjugate gradient algorithm for large-scale unconstrained optimization. J. Comput. Appl. Math. **292**, 8391 (2016)
6. Andrei, N.: Accelerated adaptive Perry conjugate gradient algorithms based on the selfscaling BFGS update. J. Comput. Appl. Math. **325**, 149–164 (2017)
7. Arazm, M.R.; Babaie-Kafaki, S.; Ghanbari, R.: An extended Dai–Liao conjugate gradient method with global convergence for nonconvex functions. Glas. Mat. **52**(72), 361–375 (2017)

8. Babaie-Kafaki, S.; Ghanbari, R.: A descent family of Dai–Liao conjugate gradient methods. Optim. Methods Softw. **29**(3), 583–591 (2013)
9. Babaie-Kafaki, S.; Ghanbari, R.: A descent extension of the Polak–Ribier'*e*–Polyak conjugate gradient method. Comput. Math. Appl. **68**, 2005–2011 (2014)
10. Babaie-Kafaki, S.; Ghanbari, R.: Two modified three-term conjugate gradient methods with sufficient descent property. Optim. Lett. **8**(8), 2285–2297 (2014)
11. Babaie-Kafaki, S.; Ghanbari, R.: The Dai–Liao nonlinear conjugate gradient method with optimal parameter choices. Eur. J. Oper. Res. **234**, 625–630 (2014)
12. Babaie-Kafaki, S.; Ghanbari, R.: Two optimal Dai–Liao conjugate gradient methods. Optimization **64**, 2277–2287 (2015)
13. Babaie-Kafaki, S.; Ghanbari, R.; Mahdavi-Amiri, N.: Two new conjugate gradient methods based on modified secant equations. J. Comput. Appl. Math. **234**(5), 13741386 (2010)
14. Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. Math. Comput. **19**, 577–593 (1965)
15. Cheng, W.: A two-term PRP-based descent method. Numer. Funct. Anal. Optim. **28**, 1217–1230 (2007)
16. Cheng, W.: A PRP type method for systems of monotone equations. Math. Comput. Model. **50**, 15–20 (2009)
17. Cheng, W.; Xiao, Y.; Hu, Q.: A family of derivative-free conjugate gradient methods for large-scale nonlinear systems of equations. J. Comput. Appl. Math. **224**, 11–19 (2009)
18. Dai, Y.H.; Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. **43**(1), 87–101 (2001)
19. Dai, Y.H.; Han, J.Y.; Liu, G.H.; Sun, D.F.; Yin, H.X.; Yuan, Y.X.: Convergence properties of nonlinear conjugate gradient methods. SIAM J. Optim. **10**(2), 348–358 (1999)
20. Dai, Y.H.; Han, J.Y.; Liu, G.H.; Sun, D.F.; Yin, H.X.; Yuan, Y.X.: Convergence properties of nonlinear conjugate gradient methods. SIAM J. Optim. **10**(2), 348–358 (1999)
21. Dai, Z.; Chen, X.; Wen, F.: A modified Perrys conjugate gradient method-based derivativefree method for solving large-scale nonlinear monotone equation. Appl. Math. Comput. **270**, 378–386 (2015)
22. Dauda, M.K.; Mamat, M.; Mohamed, M.A.; Waziri, M.Y.: Improved quasi-Newton method via SR1 update for solving symmetric systems of nonlinear equations. Malay. J. Fundam. Appl. Sci. **15**(1), 117–120 (2019)
23. Dolan, E.D.; Mor, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2, Ser. A), 201–2013 (2002)
24. Fang, X.; Ni, Q.: A new derivative-free conjugate gradient method for nonlinear system of equations. Bull. Aust. Math. Soc. **95**, 1–12 (2017)
25. Feng, D.; Sun, M.; Wang, X.: A family of conjugate gradient methods for large-scale nonlinear equations. J. Inequal. Appl. **2017**, 236 (2017)
26. Ford, J.A.; Moghrabi, I.A.: Multi-step quasi-Newton methods for optimization. J. Comput. Appl. Math. **50**(1–3), 305323 (1994)
27. Ford, J.A.; Moghrabi, I.A.: Using function-values in multi-step quasi-Newton methods. J. Comput. Appl. Math. **66**(12), 201211 (1996)
28. Ford, J.A.; Narushima, Y.; Yabe, H.: Multi-step nonlinear conjugate gradient methods for unconstrained minimization. Comput. Optim. Appl. **40**(2), 191–216 (2008)
29. Grippo, L.; Lampariello, F.; Lucidi, S.: A nonmonotone linesearch technique for Newtons method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
30. Hager, W.; Zhang, H.: A survey of nonlinear conjugate gradient methods. Pac. J. Optim. **2**(1), 35–58 (2006)
31. Hestenes, M.R.; Stiefel, E.L.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)
32. Kincaid, D.; Cheney, W.: Numerical Analysis. Brooks/Cole Publishing Company, California (1991)
33. Li, M.: A derivative-free PRP method for solving large-scale nonlinear systems of equations and its global convergence. Optim. Methods Softw. **29**(3), 503–514 (2014)
34. Li, D.H.; Fukushima, M.: A globally and superlinearly convergent Gauss–Newton-based BFGS method for symmetric nonlinear equations. SIAM J. Numer. Anal. **37**(1), 152–172 (2000)
35. Li, D.H.; Fukushima, M.: A derivative-free linesearch and global convergence of Broydenlike method for nonlinear equations. Optim. Methods Softw. **13**, 583–599 (2000)
36. Li, D.H.; Fukushima, M.: A modified BFGS method and its global convergence in nonconvex minimization. J. Comput. Appl. Math. **129**, 15–35 (2001)
37. Li, G.; Tang, C.; Wei, Z.: New conjugacy condition and related new conjugate gradient methods for unconstrained optimization. J. Comput. Appl. Math. **202**(2), 523539 (2007)
38. Liu, J.; Feng, Y.: A derivative-free iterative method for nonlinear monotone equations with convex constraints. Numer. Algorithms (2018). https://doi.org/10.1007/s11075-018-06032
39. Liu, D.Y.; Shang, Y.F.: A new Perry conjugate gradient method with the generalized conjugacy condition. In: 2010 International Conference on Computational Intelligence and Software Engineering (CiSE), p. 1012 (2010)
40. Liu, D.Y.; Xu, G.Q.: A Perry descent conjugate gradient method with restricted spectrum, optimization online, nonlinear optimization (unconstrained optimization), pp. 1–19 (2011)
41. Livieris, I.E.; Pintelas, P.: Globally convergent modified Perrys conjugate gradient method. Appl. Math. Comput. **218**, 9197–9207 (2012)
42. Mohammad, H.; Abubakar, A.B.: A positive spectral gradient-like method for nonlinear monotone equations. Bull. Comput. Appl. Math. **5**(1), 99–115 (2017)
43. Muhammed, A.A.; Kumam, P.; Abubakar, A.B.; Wakili, A.; Pakkaranang, N.: A new hybrid spectral gradient projection method for monotone system of nonlinear equations with convex constraints. Thai J. Math. **16**(4), 125–147 (2018)
44. Perry, A.: A modified conjugate gradient algorithm. Oper. Res. Tech. Notes **26**(6), 10731078 (1978)
45. Polak, B.T.: The conjugate gradient method in extreme problems. USSR Comput. Math. Math. Phys. **4**, 94–112 (1969)
46. Powell, M.J.D.: Restart procedures of the conjugate gradient method. Math. Prog. **2**, 241–254 (1977)

47. Ribiere, G.; Polak, E.: Note sur la convergence de directions conjugees. Rev. Fr. Inf. Rech. Oper. **16**, 35–43 (1969)
48. Solodov, M.V.; Svaiter, B.F.: A globally convergent inexact Newton method for systems of monotone equations. In: Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, pp. 355–369. Springer (2015)
49. Sun, W.; Yuan, Y.X.: Optimization Theory and Methods: Nonlinear Programming. Springer, NewYork (2006)
50. Waziri, M.Y.; Muhammad, L.: An accelerated three-term conjugate gradient algorithm for solving large-scale systems of nonlinear equations. Sohag J. Math. **4**, 1–8 (2017)
51. Waziri, M.Y.; Sabiu, J.: A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations, Hindawi Publishing Corporation. Int. J. Math. Math. Sci. **2015** (2015)
52. Waziri, M.Y.; Leong, W.J.; Hassan, M.A.: Jacobian free-diagonal Newton's method for nonlinear systems with singular Jacobian. Malay. J. Math. Sci. **5**(2), 241–255 (2011)
53. Waziri, M.Y.; Sabiu, J.; Muhammad, L.: A simple three-term conjugate gradient algorithm for solving symmetric systems of nonlinear equations. Int. J. Adv. Appl. Sci. (IJAAS) (2016)
54. Wei, Z.; Li, G.; Qi, L.: New quasi-Newton methods for unconstrained optimization problems. Appl. Math. Comput. **175**(2), 1156–1188 (2006)
55. Yabe, H.; Takano, M.: Global convergence properties of nonlinear conjugate gradient methods with modified secant condition. Comput. Optim. Appl. **28**(2), 203–225 (2004)
56. Yan, Q.R.; Peng, X.Z.; Li, D.H.: A globally convergent derivative-free method for solving large-scale nonlinear monotone equations. J. Comput. Appl. Math. **234**, 649–657 (2010)
57. Yasushi, N.; Hiroshi, Y.: Conjugate gradient methods based on secant conditions that generate descent directions for unconstrained optimization. J. Comput. Appl. Math. **236**, 4303–4317 (2012)
58. Yu, G.: A derivative-free method for solving large-scale nonlinear systems of equations. J. Ind. Manag. Optim. **6**, 149–160 (2010)
59. Yu, G.: Nonmonotone spectral gradient-type methods for large-scale unconstrained optimization and nonlinear systems of equations. Pac. J. Optim. **7**, 387–404 (2011)
60. Yuan, Y.X.: A modified BFGS algorithm for unconstrained optimization. IMA J. Numer. Anal. **11**, 325–332 (1991)
61. Yuan, N.: A derivative-free projection method for solving convex constrained monotone equations. ScienceAsia **43**, 195–200 (2017)
62. Yuan, G.; Lu, X.: A new backtracking inexact BFGS method for symmetric nonlinear equations. J. Comput. Math. Appl. **55**, 116–129 (2008)
63. Zhang, J.; Xu, C.: Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations. J. Comput. Appl. Math. **137**(2), 269–278 (2001)
64. Zhang, J.Z.; Deng, N.Y.; Chen, L.H.: New quasi-Newton equation and related methods for unconstrained optimization. J. Optim. Theory Appl. **102**(1), 147–157 (1999)
65. Zhang, L.; Zhou, W.; Li, D.: Some descent three-term conjugate gradient methods and their global convergence. Optim. Methods Softw. **22**(4), 697–711 (2007)
66. Zhou, W.; Li, D.: Limited memory BFGS method for nonlinear monotone equations. J. Comput. Math. **25**(1), 89–96 (2007)
67. Zhou, W.; Li, D.: Limited memory BFGS method for nonlinear monotone equations. J. Comput. Math. **25**(1), 89–96 (2007)
68. Zhou, W.; Shen, D.: An inexact PRP conjugate gradient method for symmetric nonlinear equations. Numer. Funct. Anal. Optim. **35**(3), 370–388 (2014)
69. Zhou, W.; Shen, D.: Convergence properties of an iterative method for solving symmetric non-linear equations. J. Optim. Theory Appl. **164**(1), 277–289 (2015)
70. Zhou, W.; Zhang, L.: A nonlinear conjugate gradient method based on the MBFGS secant condition. Optim. Methods Softw. **21**(5), 707–714 (2006)