Al-Mustansiriyah Journal of Science
ISSN: 1814-635X (print), ISSN:2521-3520 (online)
Volume 28, Issue 3, 2017
DOI: http://doi.org/10.23851/mjs.v28i3.114

Research Article

A New Hybrid Conjugate Gradient Method with Guaranteed Descent for Unconstraint Optimization

Basim A. Hassan*, Haneen A. Alashoor
Department of Mathematics, College of Computers Sciences and Mathematics,
University of Mosul, IRAQ
*Correspondent Author Email: basimabas39@gmail.com

Article Info

Received
01 Mar. 2017

Accepted
17 Oct. 2017

Abstract
The unconstrained optimization problem can be solving by using the conjugate gradient method. In this paper, we suggest new hybrid nonlinear conjugate gradient methods, which have the descent at every iteration and globally convergence properties under certain conditions. It can be seen clearly that new hybrid method are efficient for the given test problems depending on their numerical results.

Keywords: Conjugate gradient, Hybrid conjugate gradient, Descent condition, Numerical results.

الخلاصة
المسائل الامثلية غير المقيدة يمكن حلها باستعمال طريقة التدرج المترافق. في هذا البحث تم اقتراح طريقة تد رج مترافق مهجنة جديدة والتي تمتلك خاصيتي الانحدار عند كل تكرار وخاصية التقارب الشامل تحت شروط معينة. رأينا بشكل واضح بأن الطريقة المهجنة الجديدة كفوءة باعتماد مسائل الاختبار المُعطى على نتائجها العددية.

Introduction

Conjugate gradient methods (CG) methods are used to solve a class of numerical methods of the following unconstrained optimization problem:

$$\min\{f(x) \mid x \in R^n\} \qquad (1)$$

where f is a smooth function of n variables. We recall that these types of methods are iterative. Starting with an initial point $x_1 \in R^n$, they generate a sequence $x_k \in R^n$, by the process:

$$x_{k+1} = x_k + \alpha_k d_k \ , \qquad (2)$$

where d_k is a direction vector and the step size α_k is chosen in such a way that $\alpha_k > 0$ and satisfies the Wolfe (W) conditions :

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta_1 \alpha_k d_k^T g_k \qquad (3)$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \delta_2 d_k^T g_k \qquad (4)$$

with $\delta_1 < 1/2$ and $\delta_1 < \delta_2 < 1$, where $f_k = f(x_k)$, $g_k = g(x_k)$, g_k is the gradient of f evaluated at the current iterate x_k. The search direction is calculated by :

$$d_{k+1} = \begin{cases} -g_{k+1} & \text{if } k = 0 \\ -g_{k+1} + \beta_k d_k & \text{if } k > 0 \end{cases} \qquad (5)$$

Conjugate gradient methods differ in their way of defining the conjugancy coefficient β_k. In the literature, there have been proposed several choices for β_k which give rise to distinct conjugate gradient methods. Thus we obtain six basic conjugate gradient methods:

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{y_k^T d_k}, \quad \beta_{k+1}^{PRP} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad \beta_k^{LS} = -\frac{g_{k+1}^T y_k}{g_k^T d_k}, \qquad (6)$$

(HS-Hestenes and Stiefel [8], PR-Polak and Ribire [12], LS-Liu and Storey [11]),

$$\beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{y_k^T d_k}, \ \beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}, \beta_k^{CD} = -\frac{g_{k+1}^T g_{k+1}}{g_k^T d_k} \quad (7)$$

(DY-Dai and Yuan [5], FR-Fletcher and Reeves [6], CD-conjugate descent [7]).

These methods can be divided into two groups by the numerator used. Methods HS,
PR, LS are more efficient than DY, FR, CD (since they keep the conjugacy of direction vectors more successfully), but their global convergence cannot be proved without additional modifications. Methods DY, FR, CD are globally convergent (with some limitations concerning the step size selection), but they are less efficient than HS, PR, LS methods. More details can be found in [10].

The **idea** to attach these methods in sequence to obtain efficient algorithms leads to hybrid conjugate gradient algorithms. More details can be found in [2] [3].

Recently, the authors in [4] planned new conjugate gradient methods based on the strictly convex quadratic function approximation involves computation of the $d_k^T G s_k$ in practice it is often preferred to replace the exact computation with the use of an approximate the Hessian matrix (or sometimes its inverse) with a symmetric positive definite matrix through some effective procedure. Conjugate gradient methods are defined by the formula:

$$\beta_k^{BSQ} = \frac{g_{k+1}^T g_{k+1}}{\xi_{k+1}} \quad (8)$$

Choice β_k taken in (8), giving the conjugate gradient methods strong convergence properties and, in the same time, they may have modest practical performance.

On the other hand, methods may not be convergent, but usually they have better computer performances. The choices of β_k in these methods are:

$$\beta_k^{INQ} = \frac{g_{k+1}^T y_k}{\xi_{k+1}} \quad (9)$$

where

$$\xi_{k+1} = \alpha_k (g_k^T d_k)^2 / 2(f_k - f_{k+1}) \quad (10)$$

Using good convergence properties of the first group of methods and, in the same time, good computational performances of the second one, here we want to exploit choices of β_k in (8) and (9).

The remaining parts of the paper are in the order. In Section 2, we propose a new hybrid nonlinear conjugate gradient method. In Section 3, we present the algorithm and show that our corresponding formula can always guarantee descent condition. In Section 4, convergence analysis for the proposed method is presented. Section 5 entails the proposed method numerical results and also the representation of proposed method against some CG methods.

Materials and Methodology

A Convex Combination

In this paper we use another combination of BSQ and INQ methods. The parameter β_k of the hybrid conjugate gradient method of BSQ and INQ is formulized as :

$$\beta_k^{HBSQ} = (1-\vartheta_k)\beta_k^{INQ} + \vartheta_k \beta_k^{BSQ} \quad (11)$$

Hence, the direction d_k is given by :

$$d_0^{HBSQ} = -g_0 \ , \ d_{k+1}^{HBSQ} = -g_{k+1} + \beta_k^{HBSQ} s_k \quad (12)$$

The parameter ϑ_k is the scalar parameter to be determined later. We see that, if $\vartheta_k = 0$, then $\beta_k^{HBSQ} = \beta_k^{INQ}$ and $\vartheta_k = 1$, then $\beta_k^{HBSQ} = \beta_k^{BSQ}$. On the other hand, if $0 < \vartheta_k < 1$, then β_k^{HBSQ} is a proper convex combination of the parameters is β_k^{INQ} and β_k^{BSQ}.

Theorem 1.

If the relations (11) and (12) hold, then :

$$d_{k+1}^{HBSQ} = (1-\vartheta_k) d_{k+1}^{INQ} + \vartheta_k d_{k+1}^{BSQ} \quad (13)$$

Proof :

Having in view the relations β_k^{BSQ} and β_k^{INQ}, the relation (11) becomes:

$$\beta_k^{HBSQ} = (1-\vartheta_k)\frac{g_{k+1}^T y_k}{\xi_{k+1}} + \vartheta_k \frac{g_{k+1}^T g_{k+1}}{\xi_{k+1}} \quad (14)$$

So, the relation (12) becomes:

$$d_{k+1}^{HBSQ} = -g_{k+1} + (1-\vartheta_k)\frac{g_{k+1}^T y_k}{\xi_{k+1}} s_k + \vartheta_k \frac{g_{k+1}^T g_{k+1}}{\xi_{k+1}} s_k \quad (15)$$

In further consideration of the relation (15), we can get :

$$d_{k+1}^{HBSQ} = -(\vartheta_k g_{k+1} + (1-\vartheta_k) g_{k+1}) + \beta_k^{HBSQ} s_k, \quad (16)$$

$$d_{k+1}^{HBSQ} = -(\vartheta_k g_{k+1} + (1-\vartheta_k) g_{k+1}) + ((1-\vartheta_k)\beta_k^{INQ} + \vartheta_k \beta_k^{BSQ}) s_k \quad (17)$$

The last relation yields:

$$d_{k+1}^{HBSQ} = \vartheta_k(-g_{k+1} + \beta_k^{BSQ} s_k) + (1-\vartheta_k)(-g_{k+1} + \beta_k^{INQ} s_k) \quad (18)$$

From (18) we finally conclude:

$$d_{k+1}^{HBSQ} = (1-\vartheta_k) d_{k+1}^{INQ} + \vartheta_k d_{k+1}^{BSQ}. \quad (19)$$

Our way to find ϑ_k is to make that the conjugacy condition:

$$y_k^T d_k^{HBSQ} = 0 \quad (20)$$

Holds:

Multiplying (15) by y_k^T from the left and using (20), we get:

$$y_k^T\left[-g_{k+1} + (1-\vartheta_k)\frac{g_{k+1}^T y_k}{\xi_{k+1}} s_k + \vartheta_k \frac{g_{k+1}^T g_{k+1}}{\xi_{k+1}} s_k\right] = 0 \quad (21)$$

$$-y_k^T g_{k+1} + (1-\vartheta_k)\frac{g_{k+1}^T y_k}{\xi_{k+1}}(y_k^T s_k) + \vartheta_k \frac{g_{k+1}^T g_{k+1}}{\xi_{k+1}}(y_k^T s_k) = 0, \quad (22)$$

So,

$$y_k^T g_{k+1} - \frac{g_{k+1}^T y_k}{\xi_{k+1}}(y_k^T s_k) = \vartheta_k\left[\frac{g_{k+1}^T g_{k+1}}{\xi_{k+1}}(y_k^T s_k) - \frac{g_{k+1}^T y_k}{\xi_{k+1}}(y_k^T s_k)\right], \quad (23)$$

i.e.

$$\frac{(\xi_{k+1} - y_k^T s_k)}{\xi_{k+1}}(y_k^T g_{k+1}) = \vartheta_k \frac{g_{k+1}^T g_k}{\xi_{k+1}}(y_k^T s_k) \quad (24)$$

Finally,

$$\vartheta_k = \frac{(\xi_{k+1} - y_k^T s_k)(y_k^T g_{k+1})}{g_{k+1}^T g_k (y_k^T s_k)} \quad (25)$$

It is possible that ϑ_k, calculated as in (25), has the values outside the interval [0, 1]. However. In order to have a real convex combination in (14) the following rule is used : if $\vartheta_k \le 0$, then set $\vartheta_k = 0$ in (14) i.e. $\beta_k^{HBSQ} = \beta_k^{INQ}$, if $\vartheta_k \ge 1$, then set $\vartheta_k = 1$ in (14) i.e. $\beta_k^{HBSQ} = \beta_k^{BSQ}$. Therefore, under this rule for ϑ_k selection, the direction d_{k+1} combines the properties of the INQ and the BSQ algorithms in a convex way.

Algorithm and Lemmas

Setting up the global convergence of the proposed methods, will need the assumption on objective function, which have been used often in the literature to analyze the global convergence of nonlinear conjugate gradient methods.

Assumption (1)

i. The level set $S = \{x \in R^n \,|\, f(x) \le f(x_0)\}$ is bounded, i.e., there exists a positive constant $\zeta > 0$ such that:

$$\|x\| \le \zeta \ , \quad \forall\ x \in S\ . \quad (26)$$

ii. In some neighborhood U and S, $f(x)$ is continuously differentiable and its gradient is Lipschitz continuous namely, there exists a constant $L > 0$ such that:

$$\|g(x_{k+1}) - g(x_k)\| \le L\|x_{k+1} - x_k\|,\ \forall x_{k+1}, x_k \in U\ . \quad (27)$$

Under these assumptions of $f(x)$, there exists a constant $\Gamma \ge 0$ such that :

$$\|g_{k+1}\| \le \Gamma\ . \quad (28)$$

Now we can obtain the new conjugate gradient algorithms, as follows:

New Algorithm

Step 1. Initialization. Select $x_1 \in R^n$ and the parameters $0 < \delta_1 < \delta_2 < 1$. Compute $f(x_1)$ and

g_1. Consider $d_1 = -g_1$ and set the initial guess $\alpha_1 = 1/\|g_1\|$.

Step 2. Test for continuation of iterations. If

$\|g_{k+1}\| \le 10^{-6}$, then stop.

Step 3. Line search. Compute $\alpha_{k+1} > 0$ satisfying the Wolfe line search condition (4) and (5) and update the variables $x_{k+1} = x_k + \alpha_k d_k$.

Step 4. Compute ϑ_k parameter computation. If $g_{k+1}^T g_k (y_k^T s_k) = 0$, then set $\vartheta_k = 0$, else set ϑ_k as in (26) respectively.

Step 5. Compute β_k as in (11).

Step 6. Compute the search direction d_{k+1} as in (12). If the restart criterion of Powell $\left|g_{k+1}^T g_k\right| \ge 0.2\,\|g_{k+1}\|^2$, is satisfied, then set $d_{k+1} = -\,g_{k+1}$ otherwise set $k = k+1$ and go to **step 2.**

Here we have to present descent property.

Theorem 2.
Assume that (27) and (28) hold and let Wolfe conditions (3)–(4) hold. Also, let $\{\|s_k\|\}$ tend to zero, and let there exist some nonnegative constants η_1, η_2 such that:

$$\xi_{k+1} \ \ge \ \eta_1 \|s_k\|^2, \tag{29}$$

$$\|g_{k+1}\|^2 \ \le \ \eta_2 \|s_k\| \ . \tag{30}$$

then d_k^{HBSQ} satisfies the descent condition.

Proof :

It holds $d_0 = -g_0$. So, for $k = 0$, it holds $g_0^T d_0 = -\|g_0\|^2 < 0$. Multiplying (19) by g_{k+1}^T from the left, we get :

$$g_{k+1}^T d_{k+1}^{HBSQ} = (1-\vartheta_k) g_{k+1}^T d_{k+1}^{INQ} + \vartheta_k g_{k+1}^T d_{k+1}^{BSQ}. \tag{31}$$

If $\vartheta_k = 0$, the relation (31) becomes:

$$g_{k+1}^T d_{k+1}^{HBSQ} = g_{k+1}^T d_{k+1}^{INQ}. \tag{32}$$

So, if $\vartheta_k = 0$, the sufficient descent holds for the hybrid method, if it holds for INQ method. It is able to prove the descent for INQ method under the conditions of **Theorem 2**. It holds:

$$d_{k+1}^{INQ} = -g_{k+1} + \beta_k^{INQ} s_k. \tag{33}$$

Multiplying (33) by g_{k+1}^T from the left, we get :

$$g_{k+1}^T d_{k+1}^{INQ} = -g_{k+1}^T g_{k+1} + \beta_k^{INQ} g_{k+1}^T s_k. \tag{34}$$

Using β_k^{INQ}, we get :

$$g_{k+1}^T d_{k+1}^{INQ} = -g_{k+1}^T g_{k+1} + \beta_k^{INQ} g_{k+1}^T s_k. \tag{35}$$

From (35), we get:

$$g_{k+1}^T d_{k+1}^{INQ} = -g_{k+1}^T g_{k+1} + \beta_k^{INQ} g_{k+1}^T s_k. \tag{36}$$

From Lipschitz condition we have $\|y_k\| \le L\,\|s_k\|$, so :

$$g_{k+1}^T d_{k+1}^{INQ} = -g_{k+1}^T g_{k+1} + \beta_k^{INQ} g_{k+1}^T s_k. \tag{37}$$

But, using (29)–(30), we get :

$$g_{k+1}^T d_{k+1}^{INQ} = -g_{k+1}^T g_{k+1} + \beta_k^{INQ} g_{k+1}^T s_k. \tag{38}$$

But, because of the assumption $\|s_k\| \Rightarrow 0$, the second summand in (38) tends to zero, so there exists a number $0 < \delta \le 1$, such that:

$$\frac{1}{\eta_1} \eta_2\, L\,\|s_k\| \le \delta \|g_{k+1}\|^2. \tag{39}$$

Now, (38) becomes:

$$g_{k+1}^T d_{k+1}^{INQ} \le -\|g_{k+1}\|^2 + \delta \|g_{k+1}\|^2, \tag{40}$$

i.e.

$$g_{k+1}^T d_{k+1}^{INQ} \le -(1-\delta)\|g_{k+1}\|^2 < 0. \tag{41}$$

On the other hand, for $\vartheta_k = 1$, the relation (31) becomes:

$$g_{k+1}^T d_{k+1}^{HBSQ} = g_{k+1}^T d_{k+1}^{BSQ}. \tag{42}$$

But, the BSQ method satisfies the descent condition [4] under the Wolfe line search.

Now, let $0 < \vartheta_k < 1$ and from (31), we get :

$$g_{k+1}^T d_{k+1}^{HBSQ} \le (1-\vartheta_k) g_{k+1}^T d_{k+1}^{INQ} + \vartheta_k g_{k+1}^T d_{k+1}^{BSQ} . \quad (43)$$

We obviously can conclude now :

$$g_{k+1}^T d_{k+1}^{HBSQ} \le 0 . \quad (44)$$

Convergence Analysis

For the target of this section we remind to the next theorem.

Theorem 3.

Consider any iterative method of the form (2) and (5), where d_k satisfies a descent condition $g_k^T d_k < 0$ and α_k satisfies strong Wolfe conditions. If the Lipschitz condition holds, then either

$$\liminf_{k\to\infty} \|g_k\| = 0 \quad (45)$$

or

$$\sum_{k\ge 1} \frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2} < \infty . \quad (46)$$

It was originally given by Hager and Zhang [9]. Now we give the next theorem.

Theorem 4.

Consider the iterative method of the form $(2), (31), (12), (26)$. Let all conditions of Theorem 2 hold. Then either $g_k = 0$ for some k, or

$$\liminf_{k\to\infty} \|g_k\| = 0 . \quad (47)$$

Proof :

Let $g_k \ne 0$ for all k. Then, that lead to prove (47). Suppose, on the contrary, that there exists a number $c > 0$, such that :

$$g_k \ge c, \ \forall\ k . \quad (48)$$

From (13), we get:

$$\|d_{k+1}^{HBSQ}\| \le \|d_{k+1}^{INQ}\| + \|d_{k+1}^{BSQ}\| . \quad (49)$$

Next, it holds:

$$\|d_{k+1}^{BSQ}\| \le \|g_{k+1}\| + |\beta_k^{BSQ}| \|s_k\| . \quad (50)$$

From $(8),(28),(29), (30)$ and (50) we get :

$$\|d_{k+1}^{BSQ}\| \le \Gamma + \frac{\eta_2}{\eta_1} . \quad (51)$$

Also,

$$\|d_{k+1}^{BSQ}\| \le \Gamma + \frac{\eta_2}{\eta_1} . \quad (52)$$

Using $(28),(29), (30)$ and (52) we get :

$$\|d_{k+1}^{BSQ}\| \le \Gamma + \frac{\eta_2}{\eta_1} . \quad (53)$$

So, using $(49), (51)$ and (53) we get :

$$\|d_{k+1}^{HBSQ}\| \le 2\Gamma + \frac{\Gamma L}{\eta_1} + \frac{\eta_2}{\eta_1} . \quad (54)$$

But, now we can get :

$$\frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2} \ge \frac{c^4}{\left[2\Gamma + \frac{\Gamma L}{\eta_1} + \frac{\eta_2}{\eta_1} \right]^2} . \quad (55)$$

wherefrom

$$\sum_{k\ge 1}^{\infty} \frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^2} = \infty . \quad (56)$$

Using **Theorem 3** we conclude that this is a contradiction. So, we finish the proof.

Numerical Results

In this section, we statement several numerical experiments. We test the HBSQ method on problems in the [1] and compare their performance to that of the FR method [6]. We stop the iteration if the inequality $\|g_{k+1}\| \le 10^{-6}$ is satisfied and all these algorithms are implemented with the standard Wolfe line search conditions with $\delta_1 = 0.001$ and $\delta_2 = 0.9$. In this paper, all codes were written in FORTRAN. Tables 1 list numerical results. The meaning of each column is as follows: NI : number of iterations. NF : number of function evaluations.

Table 1: Comparison of different CG-algorithms with different test functions and different dimensions

		FR algorithm		HBSQ algorithm		HBSQ with $u = 0.5$	
P. No	**n**	**NI**	**NF**	**NI**	**NF**	**NI**	**NF**
1	**100**	**15**	**25**	**14**	**20**	**18**	**26**
	1000	**F**	**F**	**F**	**F**	**17**	**203**
2	**100**	**124**	**231**	**53**	**91**	**50**	**89**
	1000	**445**	**711**	**153**	**264**	**175**	**300**
3	**100**	**180**	**313**	**74**	**136**	**86**	**158**
	1000	**F**	**F**	**82**	**158**	**65**	**121**
4	**100**	**40**	**65**	**36**	**55**	**39**	**61**
	1000	**43**	**68**	**47**	**390**	**38**	**57**
5	**100**	**102**	**161**	**76**	**123**	**76**	**119**
	1000	**F**	**F**	**F**	**F**	**F**	**F**
6	**100**	**74**	**123**	**84**	**128**	**83**	**132**
	1000	**370**	**616**	**254**	**426**	**244**	**403**
7	**100**	**121**	**218**	**80**	**127**	**82**	**124**
	1000	**345**	**634**	**234**	**361**	**243**	**384**
8	**100**	**69**	**1202**	**33**	**286**	**32**	**223**
	1000	**98**	**1967**	**81**	**1649**	**65**	**1321**
9	**100**	**671**	**1066**	**500**	**775**	**433**	**675**
	1000	**F**	**F**	**F**	**F**	**F**	**F**
10	**100**	**95**	**150**	**97**	**149**	**97**	**148**
	1000	**349**	**568**	**309**	**481**	**330**	**511**
11	**100**	**F**	**F**	**F**	**F**	**11**	**27**
	1000	**60**	**131**	**12**	**27**	**58**	**1401**
12	**100**	**89**	**174**	**86**	**254**	**F**	**F**
	1000	**107**	**211**	**F**	**F**	**71**	**174**
13	**100**	**13**	**25**	**13**	**26**	**11**	**22**
	1000	**15**	**29**	**16**	**33**	**12**	**25**
14	**100**	**122**	**156**	**12**	**22**	**14**	**25**
	1000	**130**	**166**	**12**	**23**	**11**	**22**
15	**100**	**23**	**45**	**20**	**38**	**20**	**39**
	1000	**27**	**55**	**22**	**48**	**22**	**49**
Total		**3531**	**8715**	**2232**	**5678**	**2239**	**6314**

Fail : The algorithm fail to converge.

Problems numbers indicant for : 1. is the Trigonometric, 2. is the Perturbed Quadratic, 3. is the Raydan 1, 4. is the Extended Three Expo Terms, 5. is the Generalized Tridiagonal 2, 6. is the Quadratic QF2, 7. is the TRIDIA (CUTE), 8. is the Extended Tridiagonal 1, 9. is the ARWHEAD (CUTE), 10. is the NONDIA (CUTE), 11. is the EDENSCH (CUTE), 12. is the LIARWHD (CUTE), 13. is the Extended Block-Diagonal BD2, 14. is the DENSCHNA (CUTE), 15. is the LIARWHD (CUTE) .

Table 2: Relative efficiency of the new Algorithm

	FR algorithm	HBSQ algorithm	HBSQ with $u = 0.5$
NI	**100** %	**63.21** %	**63.40** %
NF	**100** %	**65.15** %	**72.44** %

Tables 1, show that HBSQ outperforms FR about (65 %) test problems. Moreover, FR can solve all given test problems successfully. The method HBSQ performs faster than the method FR, but it failed to solve many problems however, the method HBSQ can almost solve all given test problems successfully.
Table 1, give a comparison between the new hybrid descent methods and the Fletcher-Reeves method taking nonlinear test function with n=100, 1000. This table indicates that the new Hybrid methods saves (34 - 36) % NI and (27 - 36) % NF, especially for our selected test problems. The Percentage Performance of the improvements of the Table 1 is given by the following Table 2. Relative Efficiency of the Different Methods Discussed in the Paper.

Conclusions

We have proposed new descent hybrid conjugate gradient methods, that is, the BSQ method and the INQ method. Under suitable conditions, we proved that these method converge globally.
Extensive numerical results are also reported. The performance profiles showed that the new descent hybrid methods are efficient for the given test problems.

References

[1] N. Andrei, "An unconstrained optimization test functions collection," *Adv. Model. Optim,* vol. 10, pp. 147-161, 2008.
[2] N. Andrei, "New hybrid conjugate gradient algorithms for unconstrained optimization New Hybrid Conjugate Gradient Algorithms for Unconstrained Optimization," in *Encyclopedia of Optimization*, ed: Springer, 2008, pp. 2560-2571.
[3] N. Andrei, "A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Hestenes-Stiefel and Dai-Yuan," *Studies in Informatics and Control*, vol. 17, p. 57, 2008.
[4] B. A. Hassan and H. A. Alashoor, "A New Nonlinear Conjugate Gradient Method Based on the Scaled Matrix," *Al-Mustansiriyah Journal of Science,* vol. 27, pp. 68-73, 2017.
[5] Y.-H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on optimization,* vol. 10, pp. 177-182, 1999.
[6] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The computer journal,* vol. 7, pp. 149-154, 1964.
[7] R. Fletcher, *Practical methods of optimization*: John Wiley & Sons, 2013.
[8] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems* vol. 49: NBS Washington, DC, 1952.
[9] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific journal of Optimization,* vol. 2, pp. 35-58, 2006.
[10] L. Lukšan and J. Vlček, "Nonlinear conjugate gradient methods," *Programs and Algorithms of Numerical Mathematics,* pp. 130-135, 2015.
[11] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: theory," *Journal of optimization theory and applications,* vol. 69, pp. 129-137, 1991.
[12] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *Revue française d'informatique et de recherche opérationnelle. Série rouge,* vol. 3, pp. 35-43, 1969.