New multi-step three-term conjugate gradient algorithms with inexact line searches

Abbas Y. Al-Bayati$^1$, Muna M. M. Ali$^2$

$^1$Department of Mathematics, College of Basic Education, Telafer University, Iraq
$^2$Department of Mathematics, College of Computers Sciences and Mathematics, Mosul University, Iraq

ABSTRACT

This work suggests several multi-step three-term Conjugate Gradient (CG) algorithms that satisfy their sufficient descent property and conjugacy conditions. First, we considered (39) well-known three-term CG method, and we have, therefore, suggested two new classes of this type of algorithms based on Hestenes and Stiefel (HS) and Polak-Ribière (PR) formulas with four different versions. Both descent and conjugacy conditions for all the proposed algorithms are satisfied, at each iteration by using the strong Wolfe line search condition and its accelerated version. These new suggested algorithms are some sort of modifications to the original HS and PR methods. These CG-algorithms are considered as a sort of the memoryless BFGS update. All of our new suggested methods are proved to be globally convergent and numerically, more efficient than similar methods in the same area based on our selected set of used numerical problems.

Keywords: Global convergence property, Inexact line searches, Multi-step conjugate gradient, Sufficient descent property, Three-term conjugate gradient, Scaled conjugate gradient

1. INTRODUCTION

This paper considers the calculation of a local minimizer $x^*$, say, for the problem:

$$ \text{Min } f(x) \; \text{where } f : \mathbb{R}^n \rightarrow \mathbb{R} \quad (1) $$

Is a nonlinear function and its gradient vector is available. The Hessian matrix is not available. At the current iterative point $x_k$ the CG-method has the following form; based on the quadratic form:

$$ x_{k+1} = x_k + \alpha_k d_k \quad (2a) $$

$$ d_{k+1} = \begin{cases} -g_{k+1}, & k = 0 \\ -g_{k+1} + \beta_k d_k, & k \geq 1 \end{cases} \quad (2b) $$

Here, $\alpha_k$; step-length, $d_k$; search direction, $\beta_k$; parameter. Standard algorithms for solving this problem include CG-algorithms, with very low memory requirements, which are iterative algorithms and generate a sequence of approximations of the $f(x)$. For more details, see Hassan [1].

Journal homepage: http://ijeecs.iaescore.com
The first Three-Term CG-method was proposed by Beale [2] as:

$$d_{k+1}^{Beale} = -g_{k+1} + \beta_k d_k + \gamma_k d_T, \quad \beta_k = \beta_k^{HS}, \beta_k^{FR}, \beta_k^{PR} \text{ etc...}$$

(3a)

$$\gamma_k = \begin{cases} 0, & k = t + 1 \\ \frac{g_k^T y_t}{g_{k+1}^T y_t}, & k > t + 1 \end{cases}$$

(3b)

And $d_o$ is a restart direction and $y_k = -g_{k+1} - g_k$. Nazareth [3] proposed another three-term recurrence formula:

$$d_{k+1} = -y_k + (y_k^T y_k / y_k^T d_k) d_k + (y_k^T y_k / y_k^T d_{k-1}) d_{k-1} ; d_{-1} = 0 \text{ and } d_0 = 0$$

(4)

Also, two different three-term CG-algorithms was considered by Zhang [4, 5], that is,

$$d_{k+1}^{ZPR} = -g_{k+1} + \beta_k^{PR} d_k - \theta_k^{(1)} y_k; \quad d_{k+1}^{ZHIS} = -g_{k+1} + \beta_k^{HS} s_k - \theta_k^{(2)} y_k$$

(5)

$$\theta_k^{(1)} = \frac{g_k^T d_k}{g_k^T g_k} ; \quad \theta_k^{(2)} = \frac{g_k^T s_k}{s_k^T y_k}$$


$$d_{k+1}^{ZDL} = -g_{k+1} + [g_k^T (y_k - ts_k) / d_k^T y_k] d_k - g_k^T d_{k-1} / d_k^T y_k (y_k - ts_k)$$

(7a)

where $d_0 = g_0$; $t \geq 0$. The sufficient descent condition also holds independent of the used line search procedure, i.e. for this method:

$$g_k^T d_k = \|g_k\|^2 \text{ for all } k.$$  

(7b)

A specialization of (7) was developed by Nazareth [8] where the search direction is computed as:

$$d_{k+1}^{BS} = -g_{k+1} + \beta_k^{DL^*} s_k - g_k^T s_k / g_k^T g_k (y_k - ts_k)$$

(8a)

$$\beta_k^{DL^*} = \max \left[ \frac{y_k^T g_k}{y_k^T s_k}, 0 \right] - ts_k g_k^T s_k / y_k^T s_k : t = \frac{\|y_k\|^2}{y_k^T s_k}$$

(8b)

Furthermore, it is easy to see that (8) satisfies the sufficient descent condition independent of the line search used. Moreover, a different three-term CG-method was improved by Al-Bayati and Hassan [9], and their search direction, with inexact line search (ILS), is as follows:

$$d_{k+1}^{BH} = -g_{k+1} + \beta_k^{ILS} d_k + \gamma_k ; \gamma_k = -(d_k^T g_{k+1}) / g_k^T g_k$$

(9)

Recently, a three-term CG-method was introduced by Al-Bayati and Al-Khayat [10] and their search direction is as follows:

$$d_{k+1}^{BK} = -g_{k+1} + \beta_k^{ILS} d_k + \phi_k (y_k - ts_k) ; \phi_k = s_k^T g_k / y_k^T s_k , \quad t_k = 1 + y_k^T y_k / y_k^T s_k$$

(10)

Conjugate Gradient algorithms could be regarded as a sort of the Memoryless QN-updates, especially, for the BFGS update. This type of method was suggested for the first time by Perry [11] noted that the scalar $\beta_k$ has been chosen so that the search directions $d_k$ and $d_{k+1}$ are conjugate using ELS. Perry
relaxed this requirement where $\beta_k$ is defined by HS formula in an equivalent form, but assuming ILS, thus he obtained,

$$d_{k+1}^{\text{PI}} = -(I - d_k y_k^T / d_k^T y_k) g_{k+1}$$  \quad (11a)

but this matrix is not of full rank; Perry modified it further as:

$$d_{k+1}^{\text{P2}} = -(I - s_k y_k^T / s_k^T y_k + s_k s_k^T / s_k^T y_k) g_{k+1}$$  \quad (11b)

Then Shanno [12] addressed that (11) does not satisfy the actual QN-condition, so he modified it to obtain:

$$Q_{k+1}^{\text{SH}} = I - (s_k s_k^T + y_k s_k^T / y_k^T s_k) + (1 + y_k^T y_k / y_k^T s_k) (s_k s_k^T / y_k^T s_k)$$  \quad (12)

This new form of the projection matrix $Q_{k+1}$ has a special relationship with the BFGS update formula; defined by Dennis and More [13].

$$H_{k+1}^{\text{BFGS}} = H_k - \left[ s_k y_k^T H_k + H_k y_k s_k^T \right] \left[ y_k^T s_k \right] s_k^T$$  \quad (13)

It is easily seen that (12) is equivalent to (13) when $H_k$ replaced by $I$, i.e. if $H_k = I$, the identity matrix. For more details, see Hassan [14]. The Memoryless BFGS method is defined by:

$$d_{k+1} = -Q_{k+1} g_{k+1}$$  \quad (14)

$$d_{k+1} = -g_{k+1} + [((g_k^T y_k / s_k^T y_k) - (y_k y_k^T / s_k^T y_k)(s_k y_k g_{k+1} / s_k^T y_k)) s_k + (s_k^T g_{k+1} / s_k^T y_k) (y_k - s_k)$$  \quad (15a)

Finally, Al-Bayati, et al. [15] introduced a new CG-algorithm with different parameters, namely for:

$$\lambda_k = (1 + t) s_k^T g_{k+1} / y_k^T g_{k+1}; \quad \rho_k = (\lambda_k y_k y_k^T / s_k^T g_{k+1}) s_k$$  \quad (15b)

\section{ON MULTI-STEP THREE-TERM CG-ALGORITHMS}

Nazareth and Nocedal [16] developed a multi-step CG-method which does not need ELS; by defining the following matrices:

$$D = d_1, d_2, \ldots, d_n; \quad G = g_1, g_2, \ldots, g_n, \quad -G = DB$$  \quad \text{where B is an (NxN) upper triangular matrix with, } \beta_n = 1, \ \text{for } \ i = 1, 2, 3, k. \ \text{Assuming the Hussain matrix } G=I \ \text{(identity matrix), to get a set of mutually orthogonal vectors } g_1, g_2, \ldots, g_n. \ \text{Let us define:}

$$g_1^* = g_1$$  \quad (16a)

iterate for $k = 2, 3, 4, \ldots$ with

$$g_k^* = g_k - \left( g_k^T g_{k-1} \right) \left( g_k^* / g_{k-1}^T \right) g_{k-1}^*$$  \quad (16b)

$$g_k^* = g_k^* + c_{k-1}$$  \quad (16c)
\( c_{k-1} = \begin{cases} 
  \left( c_{k-2} + \left[ (g_k^T g_{k-2}) / (g_{k-2}^T g_{k-2}) \right] g_{k-2}^* \right) & \text{if } k = 3, 4, \ldots \\
  0 & \text{if } k = 1, 2 
\end{cases} \) \hspace{1cm} (16d)

Note that, (16), for the first iterate, is equivalent to the normal set of gradients, while it gives better approximations for the next iterates. In this paper; we have the following (four) new CG-algorithms:

\[ d_{k+1}^{N_1} = -g_{k+1}^* + \beta^H_{k} s_{k+1} + (s_k^T g_{k+1} / y_k^T s_k) (y_k^+ - t_k s_k) , \quad i = 1, 2 \] \hspace{1cm} (17a)

\[ d_{k+1}^{N_3} = -g_{k+1}^* + \beta^H_{k} s_{k+1} + (s_k^T g_{k+1} / y_k^T s_k) (y_k^+ - t_k s_k) + t_{y_k} s_k \] \hspace{1cm} (17b)

\[ d_{k+1}^{N_4} = -\theta_k g_{k+1}^* + \beta^{SPR}_{k} s_{k} - (s_k^T g_{k+1} / y_k^T s_k) P_{k} ; \quad y^* = g_{k+1}^* - g_k^* \] \hspace{1cm} (17c)

Wolfe line search procedure is fully described by many researchers, see for example Nocedal [17] and Liu and Nocedal [18]. This line search scheme has been modified by Andrei [19]. The standard Wolfe line search conditions can be defined as:

\[ f(x_k + \alpha_k d_k) \leq f(x_k) + \mu \alpha_k g_k^T d_k \] \hspace{1cm} (18a)

\[ f(x_k + \alpha_k d_k) \geq f(x_k) + (1 - \mu) \alpha_k g_k^T d_k \] \hspace{1cm} (18b)

The strong Wolfe line search conditions can be defined as:

\[ f(x_k + \alpha_k d_k) \leq f(x_k) + \mu \alpha_k g_k^T d_k \] \hspace{1cm} (18c)

\[ \left| g_{k+1}^T d_k \right| \leq -\sigma \ g_k^T d_k \] \hspace{1cm} (18d)

The accelerating Scheme for Wolfe line search technique is as follows:

\[ x_{k+1} = x_k + \lambda_k \alpha_k d_k \ ; \ \lambda_k = -\alpha_k / b_k \] \hspace{1cm} (18e)

\[ a_k = \alpha_k g_k^T d_k , \quad b_k = -\alpha_k (g_k - g_z)^T d_k , \quad g_z = \nabla f(z) \ and \ z = x_k + \alpha_k d_k . \] \hspace{1cm} (18f)

Hence, if \( b_k \neq 0 \). The new estimation of the solution is computed as, \( x_{k+1} = x_k + \lambda_k \alpha_k d_k \), else \( x_{k+1} = x_k + \alpha_k d_k \). For this reason, using the definitions of \( g_k \) , \( s_k \) , \( y_k \) and the above acceleration scheme can present the accelerated Wolfe line search procedure.

**NOTE:** (For the rest of this paper, and for simplicity, set \( g_k^* = g_k \) , \( y_k^* = y_k \)).

**First Multi-Step Three-Term CG-Method (N1)**

To compute the new search direction, \( d_{k+1}^{N_1} \) let us consider the QN-(BFGS update with H=I).

\[ d_{k+1}^{N_1} = -\theta_{k+1}^* g_{k+1} \] \hspace{1cm} (19)

\[ \theta_{k+1}^* = I - (s_k^T y_k + y_k^T s_k / y_k^T s_k) + (y_k^T s_k / y_k^T s_k) (s_k^T s_k / y_k^T s_k) \] \hspace{1cm} (20)

\[ d_{k+1} = -g_{k+1} + \beta^H_{k} d_k + (s_k^T g_{k+1} / y_k^T s_k) y_k - (1 + \left| y_k^T s_k \right| / y_k^T s_k) (s_k^T g_{k+1} / y_k^T s_k) s_k \] \hspace{1cm} (21)
Or, equivalently:

\[ \phi_k = (s_k^T g_{k+1} / y_k^T s_k) \; ; \; t_1 = (1 + \|y_k\|^2 / y_k^T s_k) \]  \hspace{1cm} (22)  

\[ d_{k+1}^{N1} = -g_{k+1} + \beta_k^{HS} d_k + \phi_k (y_k - t_1 s_k) \]  \hspace{1cm} (23a)  

\[ d_{k+1}^{N1} = -g_{k+1} + \max \{ (g_{k+1}^T y_k / d_k^T y_k), 0 \} \} d_k + \phi_k (y_k - t_1 s_k) \]  \hspace{1cm} (23b)  

Outline of N1-Algorithm.

St1. Given \( x_0 \in \mathbb{R}^n \), let \( 0 < \delta < \sigma < 1 \), \( t \geq 0 \) and \( d_0 = -g_0 \). Set \( k = 0 \).

St2. If stopping criteria \( \|g_k\|_\infty \leq 10^{-6} \) satisfied, then stop.

St3. Compute \( \phi_k \) by accelerated Wolfe line search condition (18).

St4. The parameters \( \phi_k \), \( t_1 \) are computed from (22).

St5. The new search direction \( d_{k+1}^{N1} \) is computed from (23).

St6. Check whether \( (g_{k+1}^T g_k > 0) \|g_{k+1}\|^2 \) is satisfied then set \( d_{k+1}^{N1} = -g_{k+1} \)

St7. Set \( k = k + 1 \), go to St2.

Second Multi-Step Three-Term CG-Method (N2)
To compute the new search direction \( d_{k+1}^{N2} \) let us consider Al-Bayati [20] QN-update with H=I

\[ H_{k+1}^{Bayati} = H_k + (2y_k^T H_k y_k / (s_k^T y_k)^2) s_k^T s_k - (H_k y_k y_k^T + s_k y_k^T H_k) / y_k^T s_k \]  \hspace{1cm} (24)  

Let \( d_{k+1} = -Q^{(2)}_{k+1} g_{k+1} \); and H=I  \hspace{1cm} (25)  

\[ Q^{(2)}_{k+1} = I + (2y_k^T y_k / (s_k^T y_k)^2) s_k^T s_k - (y_k y_k^T + s_k y_k^T) / y_k^T s_k \]  \hspace{1cm} (26)  

\[ \phi_k = s_k^T g_{k+1} / y_k^T s_k \; ; \; t_2 = 2(\|y_k\|^2 / y_k^T s_k) \]  \hspace{1cm} (27)  

\[ d_{k+1}^{N2} = -g_{k+1} + \beta_k^{HS} d_k + \phi_k (y_k - t_2 s_k) \]  \hspace{1cm} (28a)  

\[ d_{k+1}^{N2} = -g_{k+1} + \max \{ g_{k+1}^T y_k / d_k^T y_k, 0 \} d_k + \phi_k (y_k - t_2 s_k) \]  \hspace{1cm} (28b)  

Outline of N2-Algorithm.
All the steps as in Algorithm (N1) except:

St4. The parameters \( \phi_k \), \( t_2 \) are computed from (27).

St5. The new search direction \( d_{k+1}^{N2} \) is computed from (28).

Third Multi-Step Three-Term CG-Method (N3)
To compute the new search direction, \( d_{k+1}^{N3} \), let us consider Oren [21] QN-update with H=I.

\[ H_{k+1}^{Oren} = H_k - (s_k^T y_k H_k + H_k y_k s_k^T) / y_k^T s_k + (\eta_k + y_k^T H_k y_k / y_k s_k s_k^T / y_k^T s_k) s_k s_k^T / y_k^T s_k \]  \hspace{1cm} (29)  

\[ \eta_k = s_k^T y_k / y_k^T y_k ; \; \mathrm{H=I} \]  \hspace{1cm} (30)
\[
Q_{k+1}^{(3)} = I + \left( s_k^T y_k / y_k^T y_k + y_k^T y_k / v \right) s_k^T s_k^T / s_k^T y_k - (y_k^T s_k + s_k y_k^T) / y_k^T s_k
\]  
(31)

\[
d_{k+1} = -g_{k+1} - \frac{s_k^T g_{k+1}}{y_k^T y_k} y_k + \frac{y_k^T g_{k+1}}{y_k^T y_k} s_k - \left[ \frac{s_k^T y_k}{y_k^T y_k} + \frac{y_k^T y_k}{y_k^T s_k} \right] \frac{s_k^T g_{k+1}}{y_k^T s_k} s_k
\]  
(32)

\[
\phi_k = s_k^T g_{k+1} / v; \quad t_3 = (s_k^T y_k / y_k^T y_k + y_k^T y_k / y_k^T s_k)
\]  
(33)

\[
d_{N3}^{k+1} = -g_{k+1} + \beta_k^{HS} d_k + \phi_k(y_k - t_3 s_k)
\]  
(34a)

\[
d_{N3}^{k+1} = -g_{k+1} + \max[g_k^T y_k / d_k^T y_k, 0] d_k + \phi_k(y_k - t_3 s_k)
\]  
(34b)

**Outline of N3-Algorithm.**

All the steps as in Algorithm (N1) except:

**St4.** The parameters \( \phi_k \), \( t_3 \) are computed from (33).

**St5.** The new search direction \( d_{k+1}^{N3} \) is computed from (34).

**Fourth Scaled Multi-Step Three-Term CG-Method (N4)**

Here, we describe our new CG-method (N4) as a scaled CG-algorithm; (for details of scaled CG-algorithms; see Al-Bayati, at el. [22] and Hassan et al. [23]). This algorithm is independent of the line search, at every step. The search direction \( d_{k+1}^{N4} \) is computed as:

\[
d_{k+1}^{N4} = -\theta_{k+1} g_{k+1} + \beta_k^{SPR} s_k - \eta_k p_k
\]  
(35)

\[
\theta_k = (s_k^T s_k / y_k^T s_k). \quad p_k = \theta_k y_k - s_k, \quad \eta_k = s_k^T g_{k+1} / \| g_k \|^2
\]  
(36)

\[
\beta_k^{SPR} = \alpha_k g_k^T (g_{k+1} - g_k) / \alpha_k g_k^T g_k
\]  
(37)

\[
d_{k+1}^{N4} = -\theta_{k+1} g_{k+1} + (\theta_{k+1} s_k^T (g_{k+1} - g_k) / \alpha_k g_k^T g_k) s_k - s_k^T g_{k+1} / \| g_k \|^2 (\theta_k y_k - s_k)
\]  
(38)

**Outline of N4-Algorithm.**

All the steps as in Algorithm (N1) except:

**St4.** The parameters \( \theta_k \), \( p_k \), \( \eta_k \) are computed from (36).

**St5.** \( \beta_k^{SPR} \) and the new search direction \( d_{k+1}^{N4} \) are computed from (37) and (38) respectively.

### 3.1 CONVERGENCE ANALYSIS

Descent and global convergence conditions properties of HS and PR methods can be found directly in Hestenes and Stiefel [24] and Polak and Ribière [25]. To show that our new multi-step TTCG-algorithms (N1, N2, N3, and N4) have a descent and global convergence properties using Wolfe conditions (18). Consider

**Theorem-I.**

Suppose that Wolfe conditions (18) are satisfied, then new search directions, N1, N2, N3, and N4 defined by (23), (28), (34), and (38) satisfy the descent property, i.e.

\[
g_{k+1}^T d_{k+1} \leq 0
\]  
(39)

**Proof.**

I: For \( d_{k+1}^{N1} \) and from Wolfe conditions (18), we get \( y_k^T s_k \geq 0 \) then:

*New multi-step three-term conjugate gradient algorithms with inexact line searches (Abbas Y. Al-Bayati)*
\[ g^T_{k+1}d_{k+1} = -\left\| g_{k+1} \right\|^2 - \left( 1 + \left\| y_k \right\|^2 / y^T_k s_k \right) \left( s^T_k g_{k+1} \right)^2 / y^T_k s_k \leq 0 \] (40)

II: For \( d^N_{k+1} \) and from Wolfe condition (18), we get \( y^T_k s_k > 0 \) then:

\[ g^T_{k+1}d_{k+1} = -\left\| g_{k+1} \right\|^2 - \left( 1 + \left\| y_k \right\|^2 / y^T_k s_k \right) \left( s^T_k g_{k+1} \right)^2 / y^T_k s_k \leq 0 \] (41)

III: For \( d^N_{k+1} \) and from Wolfe condition (18), we get \( y^T_k s_k > 0 \) then:

\[ y^T_k d_{k+1} = -\left\| g_{k+1} \right\|^2 - \left( y^T_k s_k / \left\| y_k \right\|^2 + \left\| y_k \right\|^2 / y^T_k s_k \right) (s^T_k g_{k+1}) \leq 0 \] (42)

IV: For \( d^N_{k+1} \) and (18) is held, then \( y^T_k s_k \neq 0 \). From (36):

\[ g^T_{k+1}d_{k+1} = -s^T_k s_k / y^T_k s_k \left\| g_{k+1} \right\|^2 \] (43)

\[ s^T_k s_k > 0, \text{ then } y^T_k s_k > 0, \text{ for all } k \geq 0 \] (44)

Hence, \( d^N_{k+1} \), defined in (38), satisfies and

\[ g^T_{k+1}d_{k+1} = -c \left\| g_{k+1} \right\|^2, \ c > 0 \] (45)

Therefore, our new algorithms N1, N2, N3, and N4 are descent.

**Assumption (H):**

(a) “The level set \( S = \{ x : x \in \mathbb{R}^n, f(x) \leq f(x_i) \} \) is bounded, and \( x_i \) is the starting point”.

(b) “In a neighborhood \( \Omega \) of \( S \), \( f \) is continuously differentiable and its gradient is Lipchitz continuouly, namely, there exists a constant \( L \geq 0 \) such that \( \left\| g(x) - g(x_i) \right\| \leq L \left\| x - x_i \right\| , \forall x, x_i \in \Omega \)”.

under these assumptions on \( f \) there exists a constant \( \Gamma \geq 0 \) such that \( \left\| g(x) \right\| \leq \Gamma, \forall x \in S \).

We know that the new search directions generated by (23), (28), (34) and (38) are always descent directions.

**To ensure the global convergence property of these algorithms let us consider:**

**Theorem-II**

Assume that (H. a) and (H. b) hold, and consider the algorithms (2), (23), (28), (34), and (38) where N1, N2, N3 and N4 are descent directions and \( \mathcal{A}_k \) computed by (18). Suppose that \( f \) is a uniformly convex function on \( S \), i.e. there exists a constant \( \delta \) such that:

\[ (\nabla f(x) - \nabla f(y))^T (x - y) \geq \delta \left\| x - y \right\|^2 \] (46)

For all \( x, y \in S \) then \( \lim_{k \to \infty} \left\| g_k \right\| = 0 \) (47)

**Proof.**

Consider CG-algorithms (2), (23), (28), (34) and (38). From "Lipschitz continuity", we get \( \left\| y_k \right\| \leq L \left\| s_k \right\| \).

Furthermore, from uniform convexity \( y^T_k s_k \geq \mu \left\| s_k \right\|^2 \). Using the Cauchy inequality, assumption (H. a) and (H. b) and the above inequalities, we get:
\[ |\beta_k^{HS}| \leq \frac{g_k^T y_k}{s_k^T y_k / \alpha_k} \leq \frac{aL}{\Gamma \|s_k\|}: |\varphi_k| \leq \frac{|s_k^T g_{k+1}|}{\|y_k s_k\|} \leq \frac{\Gamma}{\mu \|s_k\|}, \quad |f_k| \leq 1 + \frac{\|y_k\|^2}{\|y_k s_k\|} \leq 1 + \frac{L^2}{\mu} \] \quad (48)

\[ |f_2| \leq 2 \left( \frac{\|y_k\|^2}{\|y_k s_k\|} \right) \leq 2 \frac{L^2}{\mu}, \quad |f_3| \leq \left( \frac{\|y_k s_k\|}{\|y_k\|^2} \right) + \frac{\|y_k\|^2}{\|y_k s_k\|} \leq \frac{\mu + L^2}{\mu} \] \quad (49)

Therefore, putting (48) and (49) in (17a) and (17b) yields:

\[ \|d_{k+1}^{N1}\| = \|g_k\| + \beta_k^{HS} \|\alpha_k\|^2 \|s_k\| + \|\varphi_k\| \|y_k\| + \|\varphi_k\| \|f_k\| \|s_k\| \leq \Gamma + \frac{\Gamma}{M} (L + 1 + \frac{L^2}{M}) \] \quad (50)

\[ \|d_{k+1}^{N2}\| = \|g_k\| + \beta_k^{HS} \|\alpha_k\|^2 \|s_k\| + \|\varphi_k\| \|y_k\| + \|\varphi_k\| \|f_k\| \|s_k\| \leq \Gamma + \frac{\Gamma}{M} (L + \frac{2L^2}{M}) \] \quad (51)

\[ \|d_{k+1}^{N3}\| = \|g_k\| + \beta_k^{HS} \|\alpha_k\|^2 \|s_k\| + \|\varphi_k\| \|y_k\| + \|\varphi_k\| \|f_k\| \|s_k\| \leq \Gamma + \frac{\Gamma}{M} (L + \frac{M}{L^2} + \frac{2L^3}{M}) \] \quad (52)

Similarly, for \( d_{k+1}^{N4} \), using assumption (H) and the above inequalities, we get:

\[ |\theta_k| \leq \frac{\|s_k^T s_k\|}{\|y_k^T s_k\|^2} \leq \frac{1}{\eta} \left( \frac{\|s_k\|^2}{\|y_k s_k\|^2} \right) \leq \frac{L}{\eta}: |\beta_k^{SPR}| \leq \frac{\|\theta_k g_k^T g_{k+1} y_k\|}{\alpha_k \|\theta_k g_k g_k\|} \leq \frac{\Gamma L}{\|s_k\| \|g_k\|^2} \leq \frac{\Gamma}{\|s_k\|} \] \quad (53)

Hence, putting (53) in (17c) yields:

\[ \|d_{k+1}^{N4}\| = \|\theta_k\|^2 \|g_{k+1}\| + |\beta_k^{SPR}| \|s_k\|^2 + \|y_k\|^2 \|\theta_k\|^2 \|y_k\| + \|y_k\|^2 \|s_k\| \leq \frac{\Gamma}{M} + \frac{\Gamma L}{M} + \left( \frac{1}{M} L + 1 \right) \] \quad (54)

Hence, \( d_{k+1}^{N1}, d_{k+1}^{N2}, d_{k+1}^{N3}, \) and \( d_{k+1}^{N4} \) ensure the global convergence property and \( \lim_{k \to \infty} \|g_k\| = 0 \).

3. RESULTS AND DISCUSSION

Here, we report the performance of the new proposed CG-algorithms, namely (N1, N2, N3, N4) on a set of (39) large-scale nonlinear test problems; CUTE library (see Bongartz [26] for details of these test problems) using codes written in Fortran. We have taken (10) numerical experiments with \( N=1000, 4500, 10000 \) for each test problem. To assess the reliability of our new proposed method, we have tested it against other similar CG-algorithms with the stopping criterion \( \|s_k\| \leq 10^{-6} \) or when the iterations exceed 10000 or the number of function gradient evaluations (NOFG) reach 15000 without satisfying the stopping criteria. In all these tables: \( N = "\text{Dimension of the problem}"; \) NOI = "Number of iterations"; \( \text{TIME} = \"\text{Total time required to complete the evaluation process for each test Problem}". All our numerical results are represented in Figures 1, 2, 3, 4, 5, and 6.

Figures 1 and 2 compares N1, N2, N3, and N4 against TT(BS; PR and HS) due to NOI. Figures 3 and 4 compares N1, N2, N3, and N4 against TT(BS; PR and HS) due to NOFG. Figures 5 and 6 compares N1, N2, N3, and N4 against TT(BS; PR and HS) due to TIME.
4. CONCLUSION

In this work, we have investigated new three-term multi-step search directions defined in (17). The goal of these new algorithms is a multi-step property that combines three-term CG techniques with memoryless QN-updates. Our theoretical implementation related to the requirement of sufficient descent property and ensures the property of global convergence. Also, we have presented four new three-term multi-step CG algorithms, which they use Wolfe’s and it’s acceleration as a line search subprogram. Moreover, Our numerical results show that our new algorithms have robust numerical results as compared to other similar algorithms in the same field.

ACKNOWLEDGEMENTS

The research is supported by the College of Computer Sciences and Mathematics, University of Mosul, Republic of Iraq. The authors declare that there are no conflicts of interest regarding this work.

REFERENCES


