New Hybrid Conjugate Gradient Method as a Convex Combination of Liu-Storey and Dixon Methods

Khalil K. Abbo\textsuperscript{1}, Nehal H. Hameed\textsuperscript{2}

\textsuperscript{1}Department of Mathematics, Mosul University, Mosul, Iraq
\textsuperscript{2}Department of Mathematics, Mosul University, Mosul, Iraq

Received: 31.10.2018, Accepted: 21.01.2019, Published: 21.01.2019

Abstract — In this research we estimation a new hybrid conjugate gradient algorithm, whom is convex combination of the Liu-Story algorithm and Dixon algorithm, the descent property and global convergence are establish for the new suggested method. Numerical comparisons show that the present method often behaves better than Liu-Storey and Dixon methods.

Keywords: Conjugate gradient, Descent property, Global convergence.
Mathematics Subject Classification: 80C50, 30A40, 90C26.

1 Introduction
Estimation the non-linear unconstrained optimization problem
\[ \min \{ f(x) : x \in \mathbb{R}^n \} \tag{1} \]

where \( f : \mathbb{R}^n \to \mathbb{R} \) is continuously differentiable function and bounded from below. There are many different methods for solving the problem (1) see [7], [9], [11] and [14]. We are attentive in conjugate gradient (CG) methods, which have low memory requirements and strong local and global convergence properties [1], [12]. For solving the problem (1), we estimation the CG method, which starts from an initial point \( x_1 \in \mathbb{R}^n \) and generates a sequence \( \{x_k\} \subseteq \mathbb{R}^n \) as follows:
where \( \alpha_k > 0 \) is a step size, recipient from the line search, and directions \( d_k \) are given [16], [1] by,

\[
d_{k+1} = -g_{k+1} + \beta_k s_k
\]

(3)

In the equation (3) \( g_k = \nabla f(x_k) \), \( s_k = x_{k+1} - x_k \) and \( \beta_k \) is the conjugate gradient parameter. The standard Wolfe line search conditions are repeatedly used in the conjugate gradient methods, these conditions are given by [17]

\[
f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k
\]

(4)

\[
g_{k+1}^T d_k \geq \sigma g_k^T d_k
\]

(5)

where \( d_k \) is descent direction i.e. \( g_k^T d_k < 0 \) and \( 0 < \rho < \sigma < 1 \). Strong Wolfe conditions contain of (4) and the next stronger version of (5)

\[
\left| g_{k+1}^T d_k \right| \leq -\sigma g_k^T d_k
\]

(6)

Various choices of the scalar \( \beta_k \) exist which give different performance on non-quadratic functions, yet they are equivalent for quadratic functions. In order to select the parameter \( \beta_k \) for the method in current paper, we mention the following choices:

\[
\beta_{FR} = \frac{g_{k+1}^T g_k}{g_k^T g_k}, \quad \beta_{DY} = \frac{g_{k+1}^T g_k}{d_k^T y_k}, \quad \beta_{DX} = \frac{-g_{k+1}^T g_k}{g_k^T d_k}
\]

\[
\beta_{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad \beta_{PR} = \frac{g_{k+1}^T y_k}{g_k^T g_k}, \quad \beta_{LS} = \frac{g_{k+1}^T y_k}{-g_k^T d_k}
\]

Where \( y_k = g_{k+1} - g_k \). Balance that these algorithms can be classified as algorithms with \( g_{k+1}^T g_k \) in the numerator of \( \beta_k \) and algorithms with \( g_{k+1}^T y_k \) in the numerator of parameter \( \beta_k \). The first CG algorithm with \( \beta_{FR} \) (FR) for nonlinear function introduced by [8]. \( \beta_{DY} \) (DY) method proposed by [4] and \( \beta_{DX} \) (DX) conjugate gradient method suggested by [5]. With \( g_{k+1}^T g_k \) in the numerator of \( \beta_k \) having strong convergence theory but all these methods are susceptible to strays. They begin to pick small steps without making any marked progress to the minimum [9]. On the other hand the \( \beta_{HS} \) (HSCG) suggested by [10], \( \beta_{PR} \) (PR) developed by [15] and \( \beta_{LS} \) (LS) derived by [13] methods with \( g_{k+1}^T y_k \) in the numerator of parameter \( \beta_k \), have a built in restart feature that addresses the strays phenomenon. When the step \( s_k = x_{k+1} - x_k \) is small, the factor \( y_k \) in the numerator of \( \beta_k \) tends to zero. Therefore \( \beta_k \) becomes small and the new direction \( d_{k+1} \) in (3) is essentially steepest descent (SD) direction. With other words HS,
PR and LS methods automatically adjust $\beta_k$ to avoid strays and their performances are better than the performance of method with $g_{k+1}^Tg_{k+1}$ in the numerator of $\beta_k$ [1].

2 New hybrid conjugate gradient algorithm (NK1- say)

The conjugate gradient algorithms with $g_{k+1}^Tg_{k+1}$ in the numerator of $\beta_k$ having strong convergence theory, but all these methods are susceptible to strays On the other hand the methods with $\alpha_k$ in numerator of parameter $\beta_k$ performance better than performance of methods with $g_{k+1}^Tg_{k+1}$ in the numerator of $\beta_k$ [9]. In this paper, we suggest a new hybrid $\beta^{NK1}_k$ conjugate gradient algorithm for unconstrained optimization, which is convex combination of $\beta^{DX}_k$ (Dixon) and $\beta^{LS}_k$ (Liu-Storey) as follows:

Let $\beta^{NK1}_k = \gamma \beta^{LS}_k + (1 - \gamma) \beta^{DX}_k$ \hspace{1cm} (7)

Where $\gamma \in R$ is scalar, then

$$d_{k+1} = -g_{k+1} + \beta^{NK1}_k s_k$$ \hspace{1cm} (8)

To fined the value of, $\gamma$ we use the pure conjugacy condition i.e

$$y_k^T d_{k+1} = 0$$ \hspace{1cm} (9)

Therefore

$$y_k^T d_{k+1} = -y_k^T g_{k+1} + \beta^{NK1}_k y_k^T s_k = 0$$

$$-y_k^T g_{k+1} + (\gamma \beta^{LS}_k + (1 - \gamma) \beta^{DX}_k) y_k^T s_k = 0$$

$$-y_k^T g_{k+1} + (\gamma \frac{y_k^T g_{k+1}}{-d_k^T g_k} + (1 - \gamma) \frac{g_{k+1} g_{k+1}^T}{d_k^T g_k} y_k^T s_k = 0$$

with simple algebra we get

$$\gamma = \frac{d_k^T g_k y_k^T g_{k+1} + g_{k+1} g_{k+1}^T s_k y_k}{s_k^T y_k^T g_{k+1} g_k}$$ \hspace{1cm} (10)

We restrict $\gamma$ by $0 \leq \gamma \leq 1$, for $\gamma < 0$ or $\gamma > 1$, we set $\gamma = 1$. In the following we summarize our new suggested (NK1) algorithm as follows:

Algorithm NK1-CG

Step 1. Select an arbitrary $x_0 \in R^n$, compute $f(x_0)$ and $g_0$. Set $d_0 = g_0$ and set the initial guess $\alpha_0 = 1/\|g_0\|$, and k=1.

Step 2. Fined $\alpha_k > 0$ which satisfies the Wolfe line search condition (4) and (5) or strong condition (4) and (5) and
update the variables \( x_{k+1} = x_k + \alpha_k d_k \).

**Step 3.** If \( \|g_{k+1}\| < 10^{-6} \), then stop, balance evaluate \( \beta_{k+1}^{NK_1} \) from (7) where \( \gamma \) defined in (10) and calculate \( d_{k+1} \) according to

\[
d_{k+1} = -g_{k+1} + \beta_{k}^{NK_1} s_k
\]  

**Step 4.** Increase \( k \) by one and go to Step 2.

### 3 Global convergence analysis

In order to derive the global convergence to the NK1 conjugate gradient algorithm we use the following assumptions.

**Assumption 1**

(i): The level set \( \Omega = \{ x \in R^n : f(x) \leq f(x_1) \} \) is bounded, where \( x_1 \) is the starting point.

(ii): In some neighborhood \( \mathcal{N} \) of \( \Omega \), \( f \) is continuously differentiable, and it is gradient is Lipschitz continuous:

\[
\|g(x) - g(y)\| \leq L\|x - y\| \quad \forall \ x, y \in \mathcal{N}
\]  

under assumption (i) and (ii) it is clear that \( \chi \leq \|x\| \) and \( \omega \leq \|\nabla f(x)\| \quad \forall \ x \in R^n \) where \( \chi \) and \( \omega \) are scalars.

First we prove that our suggested algorithm (NK1) generates descent search directions in the following theorem.

**Theorem 3.1** Suppose that assumption 1 holds. Let the sequence \( \{x_k\} \) be generated by \( x_{k+1} = x_k + \alpha_k d_k \) where \( d_k \) defined in equation (3), \( \beta_{k+1}^{NK_1} \) and \( \gamma \) are computed from equations (7) and (10), if \( \alpha_k \) satisfies Wolfe conditions or strong Wolfe conditions, then

\[ g_k^T d_k \leq -c \|g_k\|^2 \]  

for all \( k \geq 1 \), where \( c > 0 \).

**Proof:** The result can be establish by induction. When \( k=1 \) we have

\[
g_1^T d_1 = -g_1^T g_1 \leq -c \|g_1\|^2
\]  

Let \( g_k^T d_k \leq -c\|g_k\| \). To prove the conclusion for \( k+1 \), note that from strong Wolfe condition we have

\[
-\sigma \leq \frac{g_{k+1}^T d_k}{g_k^T d_k} \leq \sigma .
\]  

From equations (3), (7) and (10) we have

\[
d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2 + \{\gamma \beta_{k+1}^{LS} + (1-\gamma) \beta_{k+1}^{DX}\} s_k^T g_{k+1}
\]
New Hybrid Conjugate Gradient Method

\[ -\|g_{k+1}\|^2 + \{\gamma y_k^T g_{k+1} + (1-\gamma) g_{k+1}^T g_{k+1}\} s_k^T g_{k+1} \]

\[ = -\|g_{k+1}\|^2 + \{-\gamma y_k^T g_{k+1} - (1-\gamma) g_{k+1}^T g_{k+1}\} s_k^T g_{k+1} \]

\[ \leq -\|g_{k+1}\|^2 + \{-\gamma y_k^T g_{k+1} - (1-\gamma) g_{k+1}^T g_{k+1}\} \alpha_k \sigma \]

\[ = -\|g_{k+1}\|^2 + \{-\gamma g_{k+1}^T g_{k+1} + \gamma g_{k+1}^T g_{k} - g_{k+1}^T g_{k+1} + \gamma g_{k+1}^T g_{k+1}\} \alpha_k \sigma \]

\[ = -\|g_{k+1}\|^2 + \{\gamma g_{k+1}^T g_{k} - g_{k+1}^T g_{k+1}\} \alpha_k \sigma \]

Since \( g_{k+1}^T g_k \leq g_{k+1}^T g_{k+1} \) therefore, we have

\[ d_{k+1}^T g_{k+1} \leq -\|g_{k+1}\|^2 + \{\gamma \|g_{k+1}\|^2 - \|g_{k+1}\|^2\} \sigma \alpha_k \]

\[ = -\|g_{k+1}\|^2 + \gamma \sigma \alpha_k \|g_{k+1}\|^2 - \sigma \alpha_k \|g_{k+1}\|^2 \]

\[ = -(1-\gamma \sigma \alpha_k + \sigma \alpha_k) \|g_{k+1}\|^2 \]

\[ = -c \|g_{k+1}\|^2 \]

Where \( 0 < c = (1-\gamma \sigma \alpha_k + \sigma \alpha_k) < 1. \)

**Theorem 3.2** [9] Consider any iterative method of the form (3) and (7), where \( d_k \) satisfies a descent condition or sufficient descent condition, and \( \alpha_k \) satisfies strong Wolfe conditions, if the assumption 1 holds, then either

\[ \liminf_{k \to \infty} \|g_k\| = 0 \text{ or } \sum_{k=1}^{\infty} \|g_k\|^4 < \infty \]

In the following theorem we prove that our algorithm (NK1) is globally convergent

**Theorem 3.3** Consider the iterative method (11). Let all conditions of Theorem 3.2 hold. Then

\[ \liminf_{k \to \infty} \|g_k\| = 0 \]

**Proof:** Assume, on the contrary, that the method is not convergent i.e there exists a real number \( c_1 > 0 \) such that \( \|g_k\| \geq c_1 \) for all \( k \). From (11) we get

\[ d_{k+1}^{NK1} = -g_{k+1} + \{\gamma \beta_k^{LS} + (1-\gamma) \beta_k^{DX}\} s_k \]

\[ = -g_{k+1} + \gamma g_{k+1} - \gamma g_{k+1} + \gamma \beta_k^{LS} s_k + (1-\gamma) \beta_k^{DX} s_k \]

\[ = -\gamma g_{k+1} + \gamma \beta_k^{LS} s_k + (1-\gamma) g_{k+1} + (1-\gamma) \beta_k^{DX} s_k \]

\[ = \gamma d_{k+1}^{LS} + (1-\gamma)d_{k+1}^{DX} \]
Therefore
\[ \|d_{k+1}^{NK1}\| \leq \|d_{k+1}^{LS}\| + \|d_{k+1}^{DX}\| \] (13)

Furthermore
\[ \|d_{k+1}^{LS}\| \leq \|g_{k+1}\| + \|\beta_{k}^{LS}\|s_{k}\| \]

From assumption_1, \( \|g_{k+1}\| \leq \omega \) and by descent we have
\[ -g_{k}^{T}d_{k} \geq c\|g_{k}\| \quad \text{then} \quad \frac{1}{-g_{k}^{T}d_{k}} \leq \frac{1}{c\|g_{k}\|}. \]

Therefore
\[ \|\beta_{k}^{LS}\| = \frac{y_{k}^{T}g_{k+1}}{-d_{k}^{T}g_{k}} \leq \frac{y_{k}^{T}g_{k+1}}{c\|g_{k}\|} \leq \frac{\|y_{k}\||g_{k+1}\|}{c\|g_{k}\|} \leq \frac{L\|s_{k}\|}{c} = \frac{LD}{c} \] (14)

where D is a diameter of the level set \( \Omega \). Similarly
\[ \|\beta_{k}^{DX}\| = \frac{g_{k+1}^{T}g_{k+1}}{-d_{k}^{T}g_{k}} \leq \frac{g_{k+1}^{T}g_{k+1}}{c\|g_{k}\|} \leq \frac{\|g_{k+1}\|^{2}}{c\|g_{k}\|} \leq \frac{\omega}{c}. \] (15)

From (13), (14) and (15) we have
\[ \|d_{k+1}^{NK1}\| \leq \|d_{k+1}^{LS}\| + \|d_{k+1}^{DX}\| \]
\[ = 2\omega + \frac{LD}{c} + \frac{\omega D}{c} \]
\[ = 2\omega + \frac{(L + \omega)D}{c} \]

But, now we can get
\[ \frac{\|g_{k+1}\|^{4}}{\|d_{k+1}^{NK1}\|^{2}} \geq \frac{1}{2\omega} + \frac{c^{4}}{(L + \omega)D}, \]

where from \( \sum \frac{\|g_{k+1}\|^{4}}{\|d_{k+1}^{NK1}\|^{2}} = \infty \) which is contradiction with Theorem 3.2.

4 Numerical results and comparisons

In this section we shall report numerical performance of FORTRAN implementation of our new NK1 conjugate gradient algorithm on a set of unconstrained optimization test problems pick from [2] and [3]. We choose (70) large scale test problems in extended or
New Hybrid Conjugate Gradient Method

generalized form, for each function we have believe numerical experiments with the number of variables \( n=100, \ldots, 1000 \). We have compared the performance of these algorithms versus to the Dixon method(\( \beta^{DX} \)) and Liu-Storey(\( \beta^{LS} \)) method. All these algorithms are perform with the standard Wolfe line search conditions with \( \rho = 0.0001 \) and \( \sigma = 0.9 \) where the initial step size \( \alpha_1 = 1/\|g_1\| \) and initial guess for other iterations i.e. \( (k>1); \alpha_k = \alpha_{k-1}(\|d_{k-1}\|/\|d_k\|) \).

In the all cases the stopping criteria is the \( \|g_k\| \leq 10^{-6} \) and the maximum number of iteration is 1000. The codes are written in double precision FORTRAN (2000) and compiled with F77 default compiler settings. This code originally written by Andrei and modified by us. Our comparisons includes the following:

1- iter: the number of iteration
2- fg : number of function and gradient evolutions
3- CPU time

Figs. 1, 2 and 3 shows performance of these methods for solving 70 unconstrained optimization test problems, relative to the iterations (iter), function–gradient evaluations (fg) and CPU time, which are evaluated using the profile of Dolan and More [6]. That is, for each method, we plot the fraction \( p \) of problems for which the method is within a factor \( \tau \) of the best (iter) or (fg) or CPU time.

![Figure 1. Performance based on iteration.](image-url)
The left side of the figure gives the percentage of the test problems for which a method is the fastest, the right side gives the percentage of the test problems that are successfully solved by each of the methods. The top curve is the method that solved the most problems in a (iter, fg, time) that was within a factor $\tau$ of the best (iter, fg, time).

Figure 2. Performance based on function gradient evaluations.

Figure 3. Performance based on time.
References


Khalil K. Abbo, ORCID: https://orcid.org/0000-0001-5858-625X
Nehal H. Hameed, ORCID: https://orcid.org/0000-0002-4548-3091