Journal of Multidisciplinary Modeling and Optimization 1 (2) (2018),91-99.

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

Khalil K. Abbo[1], Nehal H. Hameed[2]

[1]Department of Mathematics,
Mosul University, Mosul, Iraq
prof_khalil@uomosul.edu.iq

[2]Department of Mathematics,
Mosul University, Mosul, Iraq
nhhshe@yahoo.com

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

$$\min\{ f(x) : x \in R^n \} \tag{1}$$

where $f : R^n \to R$ 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 $x_1 \in R^n$ and generates a sequence $\{x_k\} \subset R^n$ as follows:

Cite as: *K. K Abbo, N. H. Hameed, New Hybrid Conjugate Gradient Method as a Convex Combination of Liu-Storey and Dixon Method, Journal of Multidisciplinary Modeling and Optimization 1 (2) (2018), 91-99.*

$$x_{k+1} = x_k + \alpha_k d_k \tag{2}$$

where $\alpha_k > 0$ is a step size, recipient from the line search, and directions d_k are given

[16], [1] by, $d_1 = -g_1$ and

$$d_{k+1} = -g_{k+1} + \beta_k s_k \tag{3}$$

In the equation (3) $g_k = \nabla f(x_k)$, $s_k = x_{k+1} - x_k$ and β_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]

$$f(x_k + \alpha_k d_k) \le f(x_k) + \rho \alpha_k g_k^T d_k \tag{4}$$

$$g_{k+1}^T d_k \ge \sigma\, g_k^T d_k \tag{5}$$

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)

$$\left| g_{k+1}^T d_k \right| \le -\sigma\, g_k^T d_k \tag{6}$$

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

$$\beta^{\mathrm{FR}} = \frac{\mathrm{g}_{k+1}^{\mathrm{T}}\, g_{k+1}}{g_k^T g_k} \qquad \beta^{\mathrm{DY}} = \frac{\mathrm{g}_{k+1}^{\mathrm{T}}\, g_{k+1}}{d_k^T y_k} \qquad \beta^{\mathrm{DX}} = \frac{-\mathrm{g}_{k+1}^{\mathrm{T}}\, g_{k+1}}{g_k^T d_k}$$

$$\beta^{\mathrm{HS}} = \frac{\mathrm{g}_{k+1}^{\mathrm{T}}\, \mathrm{y}_k}{d_k^T y_k} \qquad \beta^{\mathrm{PR}} = \frac{\mathrm{g}_{k+1}^{\mathrm{T}}\, \mathrm{y}_k}{g_k^T g_k} \qquad \beta^{LS} = \frac{g_{k+1}^T y_k}{-\, g_k^T d_k}$$

Where $y_k = g_{k+1} - g_k$. Balance that these algorithms can be classified as algorithms with $g_{k+1}^T\ g_{k+1}$ in the numerator of β_k and algorithms with $g_{k+1}^T\ y_k$ in the numerator of parameter β_k. The first CG algorithm with β^{FR} (FR) for nonlinear function introduced by [8]. β^{DY} (DY) method proposed by [4] and β^{DX} (DX) conjugate gradient method suggested by [5]. With $g_{k+1}^T\ g_{k+1}$ in the numerator of β_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 β^{HS} (HSCG) suggested by [10], β^{PR} (PR) developed by [15] and β^{LS} (LS) derived by [13] methods with $g_{k+1}^T\ y_k$ in the numerator of parameter β_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 β_k tends to zero. Therefore β_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 β_k to avoid strays and their performances are better than the performance of method with $g_{k+1}^T\ g_{k+1}$ in the numerator of β_k [1].

2 New hybrid conjugate gradient algorithm (NK1- say)

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

Let $\beta_k^{NK1} = \gamma\, \beta_k^{LS} + (1-\gamma)\, \beta_k^{DX}$ (7)

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

$$d_{k+1} = -g_{k+1} + \beta_k^{NK1} s_k \tag{8}$$

To fined the value of, γ we use the pure conjugacy condition i. e

$$y_k^T d_{k+1} = 0 \tag{9}$$

Therefore

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

$$-y_k^T g_{k+1} + (\gamma \beta_k^{Ls} + (1-\gamma)\beta_k^{DX}) y_k^T s_k = 0$$

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

with simple algebra we get

$$\gamma = \frac{d_k^T g_k y_k^T g_{k+1} + g_{k+1}^T g_{k+1} s_k^T y_k}{s_k^T y_k g_{k+1}^T g_k} \tag{10}$$

We restrict γ by $0 \le \gamma \le 1$, 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 $x_1 \in R^n$, compute $f(x_1)$ and g_1. Set

$d_1 = g_1$ and set the initial guess $\alpha_1 = 1/\|g_1\|$, 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 β_{k+1}^{NK1} from (7) where γ defined in (10) and calculate d_{k+1} according to

$$d_{k+1} = -g_{k+1} + \beta_k^{NK1} s_k \tag{11}$$

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 $\Omega = \{x \in R^n : f(x) \le f(x_1)\}$ *is bounded, where* x_1 *is the starting point.*

(ii): In some neighborhood N *of* Ω, f *is continuously differentiable, and it is gradient is Lipschitz continuous:*

$$\|g(x) - g(y)\| \le L\|x - y\| \qquad \forall\, x, y \in N \tag{12}$$

under assumption (i) and (ii) it is clear that

$$\|x\| \le \chi \text{ and } \|\nabla f(x)\| \le \omega \quad \forall\ x \in R^n$$

where χ *and* ω *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),* β_{k+1}^{NK1} *and* γ *are computed from equations (7) and (10), if* α_k *satisfies Wolfe conditions or strong Wolfe conditions, then* $g_k^T d_k \le -c\ \|g_k\|^2$ *for all* $k \ge 1$, *where* $c > 0$.

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

$$g_1^T d_1 = -g_1^T g_1 \le -c\ \|g_1\|^2$$

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

$-\sigma \le \dfrac{g_{k+1}^T d_k}{g_k^T d_k} \le \sigma$. From equations (3), (7) and (10) we have

$$d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2 + \{\gamma\beta_{k+1}^{LS} + (1-\gamma)\beta_{k+1}^{DX}\} s_k^T g_{k+1}$$

$$= -\|g_{k+1}\|^2 + \{\gamma \frac{y_k^T g_{k+1}}{-d_k^T g_k} + (1-\gamma)\frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}\} s_k^T g_{k+1}$$

$$= -\|g_{k+1}\|^2 + \{-\gamma\, y_k^T g_{k+1} - (1-\gamma)\, g_{k+1}^T g_{k+1}\} \frac{s_{k+1}^T g_{k+1}}{d_k^T g_k}$$

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

$$= -\|g_{k+1}\|^2 + \{-\gamma\, g_{k+1}^T g_{k+1} + \gamma\, g_{k+1}^T g_k - g_{k+1}^T g_{k+1} + \gamma\, g_{k+1}^T g_{k+1}\} \alpha_k \sigma$$

$$= -\|g_{k+1}\|^2 + \{\gamma\, g_{k+1}^T g_k - g_{k+1}^T g_{k+1}\} \alpha_k \sigma$$

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 α_k satisfies strong Wolfe conditions, if the assumption 1 holds, then either*

$$\liminf_{k\to\infty} \|g_k\| = 0 \quad or \quad \sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty$$

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

$$\left\|d_{k+1}^{NK1}\right\| \leq \left\|d_{k+1}^{LS}\right\| + \left\|d_{k+1}^{DX}\right\| \tag{13}$$

Furthermore

$$\left\|d_{k+1}^{LS}\right\| \leq \left\|g_{k+1}\right\| + \left|\beta_k^{LS}\right| \left\|s_k\right\|$$

From assumption_1, $\left\|g_{k+1}\right\| \leq \omega$ and by descent we have

$$-g_k^T d_k \geq c \left\|g_k\right\| \text{ then } \quad \frac{1}{-g_k^T d_k} \leq \frac{1}{\mathrm{c} \left\|g_k\right\|} .$$

Therefore

$$\left|\beta^{LS}\right| = \frac{y_k^T g_{k+1}}{-d_k^T g_k} \leq \frac{y_k^T g_{k+1}}{c \left\|g_k\right\|} \leq \frac{\left\|y_k\right\| \left\|g_{k+1}\right\|}{c \left\|g_k\right\|} \leq \frac{L \left\|s_k\right\|}{c} = \frac{LD}{c} \tag{14}$$

where D is a diameter of the level set Ω. Similarly

$$\left|\beta^{DX}\right| = \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k} \leq \frac{g_{k+1}^T g_{k+1}}{c \left\|g_k\right\|} \leq \frac{\left\|g_{k+1}\right\|^2}{c \left\|g_k\right\|} \leq \frac{\omega}{c} . \tag{15}$$

From (13), (14) and (15) we have

$$\begin{aligned} \left\|d_{k+1}^{NK1}\right\| &\leq \left\|d_{k+1}^{LS}\right\| + \left\|d_{k+1}^{DX}\right\| \\ &= 2\omega + \frac{LD}{c} + \frac{\omega D}{c} \\ &= 2\omega + \frac{(L+\omega)D}{c} \end{aligned}$$

But, now we can get

$$\frac{\left\|g_{k+1}\right\|^4}{\left\|d_{k+1}^{NK1}\right\|^2} \geq \frac{1}{2\omega} + \frac{c^4}{(L+\omega)D} ,$$

where from $\sum \frac{\left\|g_{k+1}\right\|^4}{\left\|d_{k+1}^{NK1}\right\|^2} = \infty$ 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

generalized form, for each function we have believe numerical experiments with the number of variables n=100, …,1000. We have compared the performance of these algorithms versus to the Dixon method(β^{DX})and Liu-Storey(β^{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 $\|g_k\|_2 \leq 10^{-6}$ 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 evoluations

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 τ of the best (iter) or (fg) or CPU time.

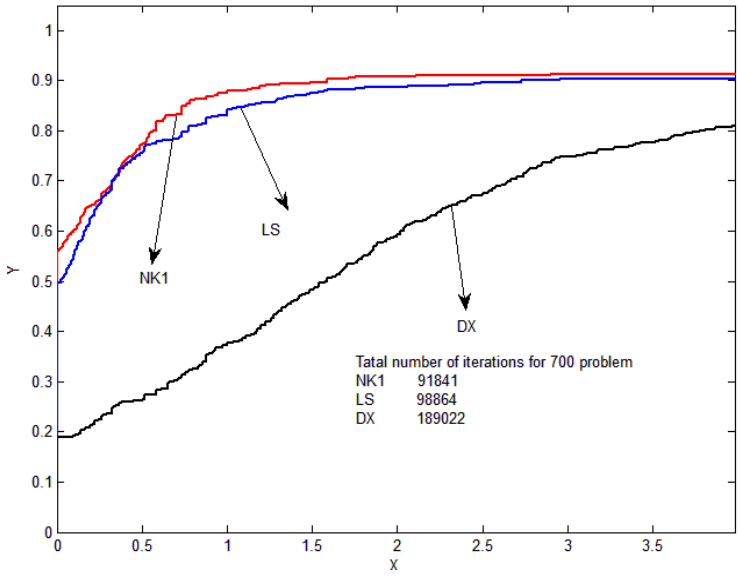

Figure 1. Performance based on iteration.

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 τ of the best (iter, fg, time).

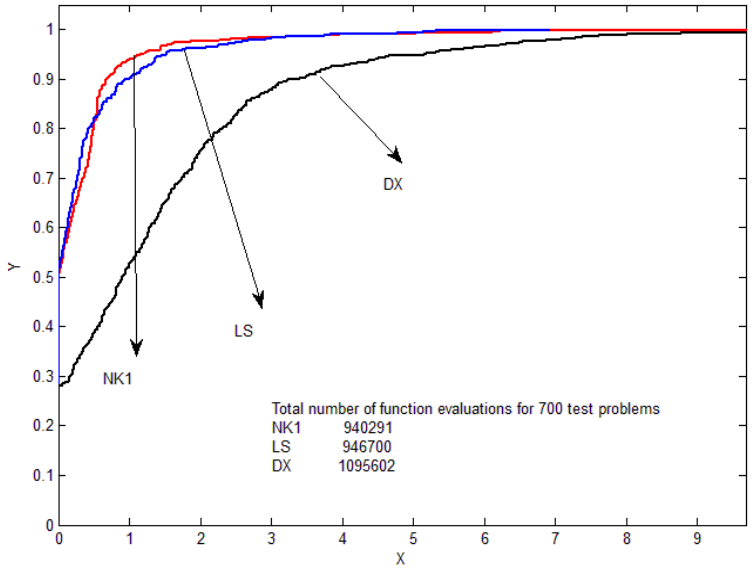

Figure 2. Performance based on function gradient evaluations.

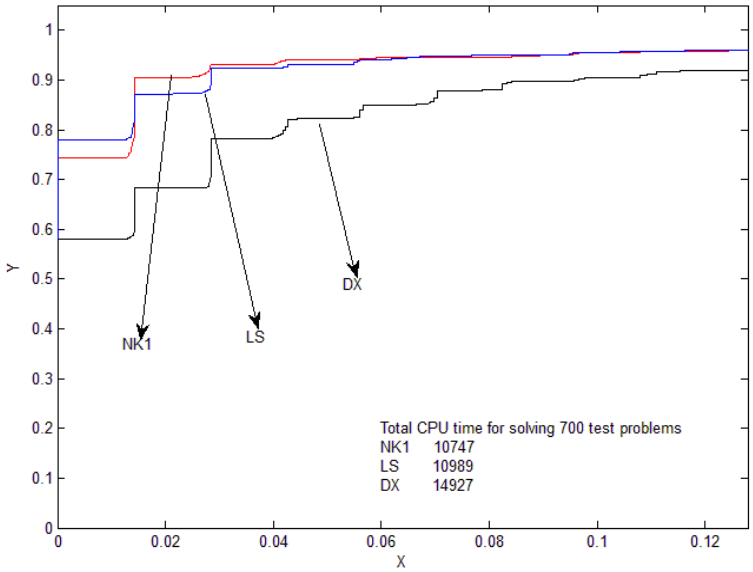

Figure 3. Performance based on time.

References

[1] N. Andrei , New hybrid conjugate gradient algorithms for unconstrained optimization, Enc. Mat. Sci., 2009 2560-2571.

[2] N. Andrei, An unconstrained optimization test function collection, Adv. Model. Optim., 10(1) 2008, 147-161.

[3] I. Bongartz, A. Conn, N. Gould and P. Toint, Constrained and unconstrained testing environment, J. Optim. Theory Appl., 21(1), 1993, 123–160

[4] Y. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. Optimiz., 10(1) 1999, 177-182.

[5 L. C. W. Dixon, Nonlinear optimisation: A survey of the state of the art, Hatfield Polytechnic. Numerical Optimisation Centre (1973).

[6] D. Dolan and J. Moré, Benchmarking optimization software with performance profiles, Math. Program., 91(2) 2002, 201-213.

[7] E. K. Chong and S. H. Zak, An introduction to optimization, John Wiley & Sons 2013.

[8] R. Fletcher and C. M. Reeves, Function minimization by Conjugate gradients, comput. J., 7(2) 1964, 149-154.

[9] W. Hager and H. Zhang , A survey of nonlinear conjugate gradient methods, Pac. J. Optim., 2(1) 2006, 35-58.

[10] M. Hestenes and E. Stiefel, Methods of conjugate Gradients For solving linear systems, J. Res. Nat. Bur. Stand., 49(1) 1952.

[11] K. K. Abbo and L. A. Abdulwahid, Generalized Dai-Yuan non-linear conjugate gradient method for unconstrained optimization, Int. J. Sci. Math. Educ., 8(6) 2017, 17993-17999.

[12] X. Li and X. Zhao, A hybrid conjugate gradient method for optimization problems, Nat. Sci., 3(1) 2011, 85.

[13] Y. Liu and C. Story, Efficient generalized conjugate gradient algorithms, part l : Theory, J. Optimiz. Theory App., 69(1) 1991, 129-137.

[14] J. Nocedal and J. Wright, Numerical Optimization, Springer Series in Operations Research, Springer Verlag, New York, 2006.

[15] E. Polak and G. Ribiere, Note sur la convergence de méthodes de directions conjuguées, Rev. Fr. Inform. Rech. O., 3(16) 1969, 35-43.

[16] S. S. Djordjević, New hybrid conjugate gradient method as a convex combination of FR and PRP Methods. Filomat, 30(11) 2016, 3083-3100.

[17] P. Wolfe, Convergence conditions for ascent methods, SIAM rev., 11(2) 1969, 226-235.

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