A Test of Non-linear Conjugate Gradient Methods Via Exact Line Search

Adeleke O. J.
Department of Computer and Information Science/Mathematics
Covenat University, Ota. Nigeria.
e-mail:wale.adeleke@covenantuniversity.edu.ng

Ajimoti Adam
Department of Mathematics, University of Ilorin, Ilorin.
e-mail: kadam4best@gmail.com

Aderemi O. A.
School of Mathematics, Statistics and Computer Science,
University of KwaZulu-Natal, Westville Campus
e-mail: Adewumia@ukzn.ac.za

Abstract

The conjugate gradient method provides a very powerful tool for solving unconstrained optimization problems. In this paper the non-linear conjugate gradient methods are tested using some benchmark non-polynomial unconstrained optimization functions. The task was accomplished by finding the exact values of the descent also known as the minimizing argument or rather the minimizer in each method. Findings also show that the basic requirement for exact convergence was satisfied by all the methods.

Keywords: Line Search, non-polynomial functions, unconstrained optimization, non-linear conjugate gradient methods, step length, search direction, descent direction.

1. Introduction

The major focus of this paper is the unconstrained optimization problem of the general form

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

where $f: \mathbb{R}^n \to \mathbb{R}$ is a continuous and differentiable function. The gradient of f at a point x_m is denoted by $G(x_m)$. In equation (1), the number of objective variable or decision variable, n, is assumed to be very large since we are dealing with large scale problems.

In order to solve problem (1), a general method of descent has been employed as discussed in [6]. This method better known as the conjugate gradient method (CGM) was first proposed by Hestenes and Stiefel in a seminar in 1952 [8]. The method so proposed was an approach for solving linear system of equations that is characterised by it symmetric positive definite nature. Later the non-linear conjugate gradient methods evolve, first from the work of Fletcher and Reeves [4] in 1964. The recurrence formula for a non-linear CGM is given by

$$x_{m+1} = x_m + \alpha_m d_m$$

(2)The step length α_m is positive and can be obtained by a line search. The search direction d_m is evaluated by

$$d_m = \begin{cases} -G_0 & if\ \ m=0 \\ -G_m+\beta_m d_{m-1} & if\ \ m\geq 1 \end{cases} \quad (3)$$

whereβ_m is the conjugate gradient updating parameter. Since the emergence of the non-linear CGMs, several variants of β_m have been proposed corresponding to different CGMs. Few of this parameters has proposed are given in table 1 below.

Table 1: Variants of Conjugate Gradient Updating Parameter

S/N	Author(s)	Year	CG Parameter
1	Hestenes and Stiefel [8]	1952	$\beta_m^{HS} = \frac{G_{m+1}^T y_m}{d_m^T y_m}$
2	Fletcher and Reeves [4]	1964	$\beta_m^{FR} = \frac{\|\|G_{m+1}\|\|^2}{\|\|G_m\|\|^2}$
3	Polak, Ribiere and Polyak [10,11]	1969	$\beta_m^{PRP} = \frac{G_{m+1}^T y_m}{\|\|G_m\|\|^2}$
4	Fletcher [5]	1987	$\beta_m^{CD} = \frac{\|\|G_{m+1}\|\|^2}{-d_m^T G_m}$
5	Liu and Storey [9]	1991	$\beta_m^{LS} = \frac{G_{m+1}^T y_m}{-d_m^T G_m}$
6	Dai and Yuan [3]	2000	$\beta_m^{DY} = \frac{\|\|G_{m+1}\|\|^2}{d_m^T y_m}$
7	Bamigbola, Ali and Nwaeze [2]	2010	$\beta_m^{BAN} = -\frac{G_{m+1}^T y_m}{G_m^T y_m}$

Here, $y_m = G_{m+1} - G_m$.

It is noteworthy that if f is convex and quadratic, in the presence of an exact line search, these methods are equivalent. This distinct behaviour is lost in the case of non-convex functions. The nature of a function, whether convex or concave, has a lot to contribute to the convergence of the methods. More on convergence in sections 5 and 6.

The remainder of this work is structured as follows: in section 2 we discussed the approach of line search in nonlinear CGMs. Section 3 discusses the exact line search which is the main focus of the experiment carried out in the work, while in section 4, the basic algorithmused in this research was presented. Sections 5 and 6, as noted above, present the numerical results and the appropriate inference respectively.

2. Line Search in Conjugate Gradient Method

In finding the local minimum, x^*, of an optimization function $f: \mathbb{R}^n \to \mathbb{R}$, two basic iterative techniques are required. One is called the line search, while the other approach is the trust region. For an iterative sequence given by$x_{m+1} = x_m + \alpha_m d_m$, the former initially computes a search direction along which the function f undergoes a decrease, and afterward determines the length of each step along that direction.

The success of a line search is largely dependent on (i) the choice of the search direction, d_m and (ii) the step length α_m. In most cases, a line search algorithm requires d_m to be a descent direction such that $d_m^T \nabla f_m < 0$.With this property, the objective function f is guaranteed to be reducible along such direction of descent. Most often, a simple line search always has the form

$$d_m = -A_m^{-1} \nabla f_m \tag{4}$$

whereA_m is a non-singular symmetric matrix.

In performing a line search, various methods are applicable. One of such methods is the steepest descent which simply recognises A_m as the identity matrix I. The Newton method on the other hand computes A_m as the exact Hessian of $\nabla^2 f(x_m)$. In the quasi-Newton method, A_m is approximated to an updated Hessian at every iteration by means of the formula

$$d_m^T \nabla f_m = -\nabla f_m^T A_m^{-1} \nabla f_m < 0 \quad (5)$$

This shows clearly that d_m is a descent direction.

To determine the value of step length α_m, one can either do this exactly or approximately. As will be evident in subsequent sections, the application of exact line search to non-linear optimization functions may fail in most cases, this is coupled with the fact that accurate line searches are very expensive to carry out and the possibility that an exact descent may not exist. As a matter of fact, it is often desirable to forfeit accuracy for a global convergence [7]. The inexact line search affords us this opportunity.

The inexact line search is a more practical approach to identifying a step length that offers adequate reduction in f. Until a certain predefined condition is satisfied, a typical inexact line search algorithm continues to search a sequence of value for the step length α. Any approximate line search of this nature works in two phases: (i) an interval-searching phase. The interval so searched contains desired step lengths. This is also known as the bracketing stage. (ii) the interpolation stage which computes a better step length with the interval in (i).

In what follows, we present a brief discussion on exact line search and the accompanying algorithm to implement it for the few chosen benchmark optimization problems.

3. Exact Line Search

Finding the step length α_m for a particular objective function $f(x)$, which is to be minimized, can be narrowed down to finding the value of $\alpha_m = \alpha$ which consequently minimizes the function

$$f(x_{m+1}) = f(x_m + \alpha d_m) = f(\alpha) \quad (6)$$

where α_m and d_m are fixed. By (6), $f(x_{m+1})$ has become a function of a single variable, that is, α. The implication is that, to find the value of α_m, a one-dimensional minimization technique will suffice. The aim of every line search is to determine the step length α_m such that $\alpha_m > 0$ along the directiond_m with the objective of ensuring a non-deteriorating rate of global convergence. To do this, we first set $\alpha_m = \alpha^*$ in such a way that

$$\alpha^* = argmin\, f(x_m + \alpha d_m) = 0 \quad (7)$$

In other words, α_m is the value of $\alpha > 0$ that minimizes the function $f$along d_m. Thus α^* in (7) can be obtained by solving the differential equation

$$\frac{d}{d\alpha} f(x_m + \alpha d_m) = 0 \quad (8)$$

Any approach which yields an exact value such as in (8) is referred to an exact line search. To a polynomial objective function, the method can be directly amendable. For a non-polynomial function, an indirect application of (8) by expanding the function using Taylor' series will do. We obtained the step length α_m from (8) by finding the real root which satisfies (7).

4. Algorithm for Exact Line Search

Step 1: Given a non-polynomial objective function $f(x)$, expand in Taylor's series and truncate the series after a number of terms. In this paper we considered only the first four terms in each series.

Step 2: For the truncated $f(x)$, substitute x with $x + \alpha d$to get $f(\alpha)$, that is, $f(\alpha) = f(x + \alpha d)$ and write as a polynomial of α.

Step 3: Compute the first-order derivative of $f(x + \alpha d)$ with respect to $\alpha$and equate to zero.

Step 4: Solve for the real root α such that $\alpha > 0$.

5. Results

The following non-polynomial objective functions obtained from Andrei [1] were used as benchmark problems.

i. Raydan 1 Function

$$f(x) = \sum_{i=1}^{n} \frac{1}{10} [\exp(x_i) - x_i], \qquad x_o = [1, 1, \dots, 1]^T$$

ii. Raydan 2 Function

$$f(x) = \sum_{i=1}^{n} [\exp(x_i) - x_i], \qquad x_o = [1, 1, \dots, 1]^T$$

iii. Diagonal 3 Function

$$f(x) = \sum_{i=1}^{n} [\exp(x_i) - \sin(x_i)], \qquad x_o = [1, 1, \dots, 1]^T$$

iv. Diagonal 6 Function

$$f(x) = \sum_{i=1}^{n} [\exp(x_i) - (1 - x_i)], \qquad x_o = [1, 1, \dots, 1]^T$$

v. Cosine Function

$$f(x) = \sum_{i=1}^{n} [\cos(x_i) + x_i^2], \qquad x_o = [1, 1, \dots, 1]^T$$

The following results were generated by a code based on the seven CG parameters in Table 1. The notations used are: n – dimension, ITR – number of iteration, f^* - optimal value of the objective function, $||g^*||$ - norm of the optimal gradient g^*, Ext – program execution time, B_mF – benchmark function, AT – average execution time per computation.

Table 2. Numerical Results with BAN (AT=0.165)

B_mF	**n**	**BAN**			
		ITR	$\boldsymbol{f^*}$	$\|\|\boldsymbol{g^*}\|\|$	**Ext**
i	5000	1	5.00e003	2.7e-011	0.11
	10000	1	1.00e004	3.0e-011	0.09
ii	5000	1	5.00e003	4.2e-012	0.02
	10000	1	1.00e004	1.2e-012	0.02
iii	5000	1	5.00e003	9.1e-012	0.04
	10000	1	1.00e004	1.7e-011	0.06
iv	5000	1	1.56e010	4.3e-007	0.07
	10000	3	5.00e010	9.6e-007	0.77
v	5000	1	-8.33e003	1.3e-013	0.43
	10000	1	-1.67e004	1.8e-013	0.04

Table 3. Numerical Results with FR (AT=0.042+)

B_mF	**n**	**FR**			
		ITR	$\boldsymbol{f^*}$	$\|\boldsymbol{g^*}\|$	**Ext**
i	5000	1	5.00e003	2.7e-011	0.05
	10000	1	1.00e004	3.0e-011	0.09
ii	5000	1	5.00e003	4.2e-012	0.02
	10000	1	1.00e004	1.2e-012	0.02
iii	5000	1	5.00e003	9.1e-012	0.04
	10000	1	1.00e004	1.7e-011	0.06
iv	5000	1	1.56e010	4.3e-007	0.04
	10000	Test Failed			
v	5000	1	-8.33e003	1.3e-013	0.03
	10000	1	-1.67e004	1.8e-013	0.03

Table 4. Numerical Results with PRP (AT=0.082)

B_mF	**n**	**PRP**			
		ITR	$\boldsymbol{f^*}$	$\|\boldsymbol{g^*}\|$	**Ext**
i	5000	1	5.00e003	2.7e-011	0.07
	10000	1	1.00e004	3.0e-011	0.09
ii	5000	1	5.00e003	4.2e-012	0.02
	10000	1	1.00e004	1.2e-012	0.02
iii	5000	1	5.00e003	9.1e-012	0.04
	10000	1	1.00e004	1.7e-011	0.06
iv	5000	1	1.56e010	4.3e-007	0.10
	10000	3	5.00e010	6.0e-007	0.23
v	5000	1	-8.33e003	1.3e-013	0.16
	10000	1	-1.67e004	1.8e-013	0.03

Table 5. Numerical Results with HS (AT = 0.057)

B_mF	n	HS			
		ITR	f^*	$\|\|g^*\|\|$	Ext
i	5000	1	5.00e003	2.7e-011	0.07
	10000	1	1.00e004	3.0e-011	0.09
ii	5000	1	5.00e003	4.2e-012	0.02
	10000	1	1.00e004	1.2e-012	0.02
iii	5000	1	5.00e003	9.1e-012	0.04
	10000	1	1.00e004	1.7e-011	0.06
iv	5000	1	1.56e010	4.3e-007	0.04
	10000	3	5.00e010	6.0e-007	0.15
v	5000	1	-8.33e003	1.3e-013	0.05
	10000	1	-1.67e004	1.8e-013	0.03

Table 6. Numerical Results with CD (AT = 0.046^+)

B_mF	n	CD			
		ITR	f^*	$\|\|g^*\|\|$	Ext
i	5000	1	5.00e003	2.7e-011	0.08
	10000	1	1.00e004	3.0e-011	0.09
ii	5000	1	5.00e003	4.2e-012	0.02
	10000	1	1.00e004	1.2e-012	0.02
iii	5000	1	5.00e003	9.1e-012	0.04
	10000	1	1.00e004	1.7e-011	0.06
iv	5000	1	1.56e010	4.3e-007	0.04
	10000	Test Failed			
v	5000	1	-8.33e003	1.3e-013	0.03
	10000	1	-1.67e004	1.8e-013	0.03

Table 7. Numerical Results with DY (AT=0.045^{+})

B_mF	**n**	**DY**			
		ITR	f^*	$\|\|g^*\|\|$	**Ext**
i	5000	1	5.00e003	2.7e-011	0.07
	10000	1	1.00e004	3.0e-011	0.10
ii	5000	1	5.00e003	4.2e-012	0.02
	10000	1	1.00e004	1.2e-012	0.02
iii	5000	1	5.00e003	9.1e-012	0.05
	10000	1	1.00e004	1.7e-011	0.05
iv	5000	1	1.56e010	4.3e-007	0.04
	10000	Test Failed			
v	5000	1	-8.33e003	1.3e-013	0.03
	10000	1	-1.67e004	1.8e-013	0.03

Table 8. Numerical Results with LS (AT=0.055)

B_mF	**n**	**LS**			
		ITR	f^*	$\|\|g^*\|\|$	**Ext**
i	5000	1	5.00e003	2.7e-011	0.07
	10000	1	1.00e004	3.0e-011	0.10
ii	5000	1	5.00e003	4.2e-012	0.02
	10000	1	1.00e004	1.2e-012	0.02
iii	5000	1	5.00e003	9.1e-012	0.04
	10000	1	1.00e004	1.7e-011	0.05
iv	5000	1	1.56e010	4.3e-007	0.04
	10000	3	5.00e010	7.9e-007	0.15
v	5000	1	-8.33e003	1.3e-013	0.03
	10000	1	-1.67e004	1.8e-013	0.03

6. Remark

The norm of g^* is defined as

$$\|g^*\| = \left(\sum_{i=1}^{n} g_i^2\right)^{1/2} < \varepsilon \Longrightarrow \sum_{i=1}^{n} g_i^2 < \varepsilon^2 \quad (9)$$

If all the components of g^* have the same value, then

$$n g_i^2 < \varepsilon^2 \Longrightarrow g_i^* < \sqrt{\frac{\varepsilon^2}{n}} \quad \forall\, i \quad (10)$$

The highest value of n used in all the methods is 10,000 and the predefined tolerance used is $\varepsilon = 10^{-6}$. Substituting these for n and ε in (10), we have

$$g_i^* < \sqrt{\frac{10^{-12}}{10000}} = \frac{10^{-6}}{100} = 10^{-8} \quad \forall\ i$$

Now, if $g_k^* \neq 0$ and $g_i^* = 0 \ \forall\, i \neq k$

i.e., $$g_k^{*2} < \varepsilon^2 \Longrightarrow g_k^* = \sqrt{\varepsilon^2} = 10^{-6}$$

Thus, $10^{-8} \leq g_i^* \leq 10^{-6}$, from which we deduce that $g^* \approx 0$. This is the requirement for the exact convergence.

The minimization of the above problems using all the methods are all satisfied for $\varepsilon = 10^{-6}$ except for the methods FR, CD, and DY which failed for problem (iv) at $n = 10000$. All the methods gave the same value of f^*. Considering the execution time, we conclude that all used methods have fast rates of convergence. Thus, the methods are stable and consistent.

7. References

[1] Andrei, N. (2004). Unconstrained optimization test functions. Unpublished manuscript; Research Institute for Informatics. Bucharest 1, Romania.

[2] Bamiigbola, O.M., Ali, M. And Nwaeze, E. (2010). An efficient and convergent method for unconstrained nonlinear optimization. Proceedings of International Congress of Matthematicians. Hyderabad, India.

[3] Dai, Y and Yuan, Y. (2000). A nonlinear conjugate gradient with a strong global convergence properties: SIAM Journal on Optimization. Vol. 10, pp. 177-182.

[4] Fletcher, R. and Reeves, C.M. (1964). Function minimization by conjugate gradients. Computer Journal. Vol 7, No. 2.

[5]Fletcher, R. (1997). Practical method of optimization. Second ed. John Wiley, New York.

[6]Hasdorff L. (1976). Gradient optimization and nonlinear control. John Wiley, New York.

[7] Hatam Z. A. T. (1991). The effect of inexact line search on conjugate gradient method. Unpublished M.Sc. Dissertation. King Fahd University of Petroleum and Minerals, Saudi Arabia, pp. 2-3.

[8] Hestenes, M.R. and Stiefel, E. (1952). Method of conjugate gradient for solving linear equations. J. Res. Nat. Bur. Stand. 49.

[9] Liu, Y. and Storey, C. (1992). Efficient generalized conjugate gradient algorithms. Journal of Optimization Theory and Application. Vol. 69, pp. 129-137.

[10] Polak, E. and Ribiere, G. (1969). Note sur la convergence de directions conjugees. Rev. Francaise Informat Recherche Operationelle. 3e Annee 16, pp. 35-43.

[11] Polyak, B.T. (1969). The conjugate gradient in extreme problems. USSR Comp. Math. Math. Phys. 94-112.