International Journal of Mathematics and Statistics Studies

Vol.2, No.1, pp. 14-24, March 2013

Published by European Centre for Research Training and Development UK (www.ea-journals.org)

NUMERICAL COMPARISON OF LINE SEARCH CRITERIA IN NONLINEAR CONJUGATE GRADIENT ALGORITHMS

Adeleke O. J.
Department of Computer and Information Science/Mathematics
Covenat University, Ota. Nigeria.

Aderemi O. A.
School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal,
Westville Campus. South Africa.

Omoregbe N. I.
Department of Computer and Information Science/Mathematics
Covenat University, Ota. Nigeria.

Adekunle, R. A.
Department of Mathematics and Statistics
Osun State Polytechnic, Ire. Nigeria.

ABSTRACT: *One of the open problems known to researchers on the application of nonlinear conjugate gradient methods for addressing unconstrained optimization problems is the influence of accuracy of linear search procedure on the performance of the conjugate gradient algorithm. Key to any CG algorithm is the computation of an optimalstep size for which many procedures have been postulated. In this paper, we assess and compare the performance of a modified Armijo and Wolfe line search procedures on three variants of nonlinear CGM by carrying out a numerical test. Experiments reveal that our modified procedure and the strong Wolfe procedures guaranteed fast convergence.*

KEYWORDS: Nonlinear Conjugate Gradient Method, Unconstrained Optimization Problems, Armijo Line Search, Wolfe Line Search, Large Scale Problems.

INTRODUCTION

First introduced by Hestenes and Stiefel [19] in 1952, and later extended to the nonlinear form, extensive works have been carried out on nonlinear conjugate gradient methods by Daniel [10], Dixon et al [12], Hager and Zhang [17], Beale [7], Andrei [2], Yabe and Takano [28], Yuan and Lu [29], just to mention a few.

The nonlinear CGM for addressing optimization problems considers an unconstrained minimization problem of the form

$$\min f(x), \quad x \in \mathbb{R}^n \qquad (1)$$

where$\mathbb{R}^n$ is a$n-$dimensional Euclidean space and $f: \mathbb{R}^n \longrightarrow \mathbb{R}$ is a continuously differentiable function. When n is large (say $n > 1000$) the related problem results in large scale minimization problem. The CGM is a very suitable approach for solving large scale

minimization problems [5, 8, 25]. In order to solve such problems, there is always a need to design an algorithm that avoids high storage of arrays and reduces the cost of computation.
A nonlinear CGM generates a sequence of point x_k, with $k > 0$, by guessing an initial point $x_0 \in \mathbb{R}^n$, and making use of the recurrence relation $x_{k+1} = x_k + \sigma_k p_k$. $p_k$is called the search direction and can be generated by using the relation

$$p_{k+1} = -g_{k+1} + \beta_k p_k, p_0 = -g_0 \tag{2}$$

where $g_k = -\nabla f(x_k)$, that is, the gradient of f at x_k, and β_k is the CG update parameter. Different conjugate gradient methods correspond to different choice of β_k [17].
One notable feature of the nonlinear CGM is the involvement of the line search procedure in its algorithm. For a strictly quadratic objective function, the CG algorithm with exact line search converges finitely [26]. In case of nonlinear (non-quadratic) function, it is more appropriate and cost efficient to adopt the inexact line search procedure although in most cases, accuracy is sacrificed for global convergence. The table below where$y_k = g_{k+1} - g_k$, is presented chronologically for some choices of β_k.

Table 1: Different update parameters for CGM

S/N	Author(s)	Year	CG Parameter
1	Hestenes and Stiefel [17]	1952	$\beta_k^{HS} = \frac{G_{k+1}^T y_k}{d_k^T y_k}$
2	Fletcher and Reeves [13]	1964	$\beta_k^{FR} = \frac{\|G_{k+1}\|^2}{\|G_k\|^2}$
3	Daniel [10]	1967	$\beta_k^{D} = \frac{g_{k+1}^T \nabla^2 f(x_k) p_k}{p_k^T \nabla^2 f(x_k) p_k}$
4	Polak, Ribiere and Polyak [21,22]	1969	$\beta_k^{PRP} = \frac{G_{k+1}^T y_k}{\|G_k\|^2}$
5	Fletcher [14]	1987	$\beta_k^{CD} = \frac{\|G_{k+1}\|^2}{-d_k^T G_k}$
6	Liu and Storey [20]	1991	$\beta_k^{LS} = \frac{G_{k+1}^T y_k}{-d_k^T G_k}$
7	Dai and Yuan [9]	2000	$\beta_k^{DY} = \frac{\|G_{k+1}\|^2}{d_k^T y_k}$
8	Hager and Zhang [16]	2005	$\beta_k^{N} = \left(y_k - 2p_k \frac{\|y_k\|^2}{p_k^T y_k}\right)^T \frac{g_{k+1}}{p_k^T y_k}$
9	Bamigbola, Ali and Nwaeze [6]	2010	$\beta_k^{BAN} = -\frac{G_{k+1}^T y_k}{G_k^T y_k}$

It is more preferable in large scale problems to make use of β_k that do not require the evaluation of Hessian matrix that often require high computer storage capacity. In case of exact line search, equivalence in the methods can be established for strongly convex quadratic functions. For a different class of functions, this important feature is lost. In such instances, the inexact line search approaches are employed. Among these are the Armijo and Wolfe line search procedures. Our aim in this work is to verify the suitability of these two procedures on the nonlinear conjugate gradient algorithm.

The remainder of this article is structured as follows. In section 2, an overview of line search technique as applied to CGM is presented. Focus is given to inexact line search in section 3, while in section 4, the two line search procedures adopted in this paper are considered by establishing their theoretical frameworks and outlining the essential algorithms. In section 5, experiments were carried out by incorporating the search procedures in the nonlinear CG algorithm. The work is concluded by stating few remarks on the results generated and possible future work.

LINE SEARCH IN CONJUGATE GRADIENT METHODS

In its quest to obtain an optimal value to an objective function, what a line search procedure in a CG algorithm does is to compute a search direction p_k and then makes a decision on how far to move along such direction. The line search iteration followed to attain this end is given by where σ_k is the step size. The challenges of finding a good σ_k are both in avoiding that the step size is too long or too short [18]. On common and earliest approach of finding σ_k, known as exact line search, was to find a σ^* such that

$$\sigma^* = argmin\, f(x_k + \sigma p_k) = 0, \sigma > 0. \tag{3}$$

This approach is easily implemented for quadratic objective functions and can be amendable to non-polynomial functions upon expansion in Taylor's series. This method, though usable for nonlinear functions, is considered to be very costly. Researchers are therefore forced to resort to an approximate method called the inexact line search. For this approach, the basic idea is to (i) formulate a criterion that ensures the steps are neither too long or too short (ii) guess a good initial step size and (iii) construct a sequence of updates that satisfy the criterion in (i) after every few steps. A lot of have been done on the formulation of different criteria. Among these as highlighted by Andrei [3] are Wolfe [27], Goldstein [15], Armijo [4], Powell [24], Potra and Shi [23], Dennis and Schnabel [11]. The most widely used is based on the Wolfe line search rule [3]. In what follows, we give more attention to the inexact line search.

THE ARMIJO AND WOLFE LINE SEARCH RULES

Armijo line search rule: of the several inexact line search methods, the Armijo rule among others guarantees a sufficient degree of accuracy, hence ensuring the convergence of a conjugate gradient algorithm. Two parameters are key to this rule: $0 < \varepsilon < 1$and $\alpha > 1$. Assume a minimization of $f(\sigma)$ such that $f'(0) < 0$ (this indeed is true of every line search problems arising in descent algorithm). A first order approximation of $f(\sigma)$ at $\sigma = o$is $f(0) + \sigma f'(0)$. We give the following consequential definition

$$\hat{f}(\sigma) = f(0) + \sigma\varepsilon f'(0), \forall\, \sigma > 0. \tag{4}$$

A step size $\sigma^*$is considered by the Armijo rule to be acceptable provided the following conditions (criteria) are fulfilled:

- $f(\sigma^*) \le \hat{f}(\sigma^*)$
- $f(\alpha\sigma^*) \ge \hat{f}(\alpha\sigma^*)$

The two conditions above are generally referred to as the Armijo line search criteria. While the first of the conditions assures sufficient decrease in the objective function, the other prevents the step size from becoming unbearably too small. These two can be put together as

$$f(x_k + \sigma p_k) \le f(x_k) + \sigma\varepsilon\nabla f(x_k)^T p_k. \tag{5}$$

With the use of this rule, a range of acceptable step sizes is produced. To implement the Armijo rule to find such an acceptable step size usually involves an iterative procedure where fixed values of α and an initial $\sigma^* > 0$ are assumed. The first of the following algorithms describes the Armijo line search rule while the other gives a simple modified form. The modification only involves the parameter α as can be seen.

Algorithm 1: Armijo Line Search Procedure

Step 1: Choose $\varepsilon \in (0,\ \frac{1}{2})$, $\alpha \in (0,\ 1)$ and set $\sigma = 1$

Step 2: While $f(x_k + \sigma p_k) > f(x_k) + \sigma\varepsilon\nabla f(x_k)^T p_k$, set $\sigma = \alpha\sigma$ for some $\alpha \in (\alpha_1,\ \alpha_2) = (0.1,\ 0.5)$, i.e., α is chosen randomly from the open set $(0.1,\ 0.5)$

Step 3: Terminate loop with $\sigma_k = \sigma$ and set $x_{k+1} = x_k + \sigma_k p_k$.

Algorithm 2: Modified Armijo Line Search Procedure

Step 1: choose $\varepsilon \in (0,\ \frac{1}{2})$, $\alpha \in (0,\ 1)$ and set $\sigma = 1$

Step 2: while $f(x_k + \sigma p_k) > f(x_k) + \sigma\varepsilon\nabla f(x_k)^T p_k$, set$\sigma = \alpha\sigma$ for some $\alpha \in (\alpha_1,\ \alpha_2) = (0.1,\ 0.5)$, i.e., α is chosen randomly from the open set $(0.1,\ 0.5)$

Step 3: Terminate loop with $\sigma_k = \sigma$ and set $x_{k+1} = x_k + \sigma_k p_k$.

Wolfe Line Search Rule: As stated earlier, the Armijo rule $f(x_k + \sigma p_k) > f(x_k) + \sigma\varepsilon\nabla f(x_k)^T p_k$ for some $\varepsilon \in (0,\ \frac{1}{2})$, ensures a sufficient decrease in the objective function. If we denote the right hand side of the above inequality, being linear, as $\ell(\sigma)$, the function $\ell(\cdot)$ has a negative slope given by $\varepsilon\nabla f(x_k)^T p_k$, which lies above the graph of a univariate function $\varphi(\sigma) = f(x_k + \sigma p_k)$, $\sigma > 0$, because $\varepsilon \in \left(0,\ \frac{1}{2}\right)$. A condition for sufficient decrease is that σ is acceptable only if $\varphi(\sigma) \le \ell(\sigma)$. This condition alone is not sufficient in itself to ensure that a CG algorithm makes remarkable progress towards an optimal solution. Thus, a second requirement is usually desirable which requires $\sigma_k$to satisfy

$$\nabla f(x_k + \sigma_k p_k)^T p_k \ge \delta\nabla f(x_k)^T p_k \qquad (6)$$

with $\delta \in (\varepsilon,\ 1)$. Observe that the left hand side of () is simply the derivative of $\varphi(\sigma_k)$. Condition () is known as the curvature condition and it ensures that $\varphi'(\sigma_k) > \delta\varphi'(0)$. This assumption is ideal since if $\varphi'(\sigma)$ is strongly negative, it is an indication that f can be reduced significantly by a movement along the chosen p_k. The converse is true. Conditions (5) and (6) are collectively known as the Wolfe condition. We restate it as follows:

- $f(x_k + \sigma_k p_k) \le f(x_k) + \sigma\varepsilon\nabla f(x_k)^T p_k$
- $\nabla f(x_k + \sigma_k p_k)^T p_k \ge \delta\nabla f(x_k)^T p_k$

with$0 < \varepsilon < \delta < 1$

There are however cases where a step size may satisfy the conditions (5) and (6) above without necessarily being close to a minimizer of φ. In order to close this gap, a more stringent two-sided test on the slope of φ which forces σ_k to lie in atleast a neighbourhood of a local minimizer of φ is usually resorted to. This test is referred to as the strong Wolfe condition and is stated as follows:

- $f(x_k + \sigma_k p_k) \le f(x_k) + \varepsilon\sigma_k\nabla f(x_k)^T p_k$
- $|\nabla f(x_k + \sigma_k p_k)^T p_k| \le \delta|\nabla f(x_k)^T p_k|$

With $0 < \varepsilon < \delta < 1$

This stronger condition restrains the derivative $\varphi'(\sigma_k)$ from becoming excessively positive. On the convergence properties of these conditions, [30] and [27] are very good works to consider.

Algorithm 3: Strong Wolfe Line Search Procedure

Step 1: Choose $\varepsilon \in (0,\ 1)$and $\delta = 0.75$, $\alpha = 0.5$. Set $\sigma = 1$

Step 2: If $f(x_k + \sigma_k p_k) > f(x_k) + \varepsilon\sigma_k \nabla f(x_k)^T p_k$
and $|\nabla f(x_k + \sigma_k p_k)^T p_k| \leq \delta |\nabla f(x_k)^T p_k|$
Take $\sigma = \alpha\sigma$

Step 3: Terminate loop with $\sigma_k = \sigma$ and set$x_{k+1} = x_k + \sigma_k p_k$

Algorithm 4: Nonlinear Conjugate Gradient Method

Step 1: Given $x_0 \in \mathbb{R}^n$, set $p_0 = -g_0$. If $g_0 = 0$, stop

Step 2: Find $\sigma_k = argmin\, f(x_k + \sigma p_k)$, with $\sigma > 0$.

Step 3: Update the variables x_k and g_k according to the iterative scheme $x_{k+1} = x_k + \sigma_k p_k$ and Table 1 respectively.

Step 4: Determine β_k and update p_k.

Step 5: Set $k = k + 1$, and go repeat the process from step 2.

Numerical Experiments

The numerical results of our work are reported in Table to. Comparison was made for three well established variants of the nonlinear conjugate gradient methods. These are FR, PRP, and HS. This we achieved by incorporating each of algorithms (1) – (3) into the conjugate gradient algorithm (4). The test functions are drawn from [1]. The problems were tested for different values of n ranging from $n = 500$to $n = 10000$. This is in conformity to the idea of large scale problems. The following symbols have been adopted. FN – function name, n – dimension, I – number of iteration, Ext – computer execution time, f^*- optimal value of the objective function f, $\|g^*\|$ - norm of the gradient of f^*. The stopping criterion was taken as $\|g_k\|_\infty \leq 10^{-6}$. This is very suitable for large scale problems.

The tested functions are:1 – Extended Rosenbrook Function, 2 – Diagonal 4 Function, 3 – Extended Himmelblau Function, 4 – Extended Beale Function, 5 – Modified Extended Beale Function, 6 – Extended Block Diagonal BD1 Function, 7 – Generalized Tridiagonal-1 Function, 8 – Generalized White and Holst Function ($c = 100$), 9 – Extended Tridiagonal-1 Function, 10 – Extended Three Exponential Terms Function.

Table 2: Numerical Results for Armijo Line Search Procedure.

FN	n	FR		PRP		HS	
		I/Ext	$f^*/\|g^*\|$	I/Ext	$f^*/\|g^*\|$	I/Ext	$f^*/\|g^*\|$
1	5000 10000	107/1.19 520/11.58	8.7e – 15/3.5e – 0.7 3.1e– 13/7.9e – 07	174/3.37 178/6.48	1.79e – 13/9.0e – 07 4.14e– 15/5.9e – 07	85/0.95 87/1.44	7.2e – 16/7.6e – 07 4.6e– 16/6.1e – 07
2	5000 10000	30/0.46 27/0.72	1.4e – 16/3.5e – 07 4.00e – 16/5.6e – 07	37/0.92 37/1.43	2.35e – 16/4.0e – 07 4.71e – 16/5.6e – 07	Test Failed Test Failed	Test Failed Test Failed
3	5000 10000	34/0.86 36/1.38	3.9e– 15/7.2e – 07 1.3e – 15/3.4e – 07	34/0.73 34/1.28	3.5e– 15/6.2e – 07 7.0e – 15/8.8e – 07	61/0.95 Test Failed	7.4e – 15/6.2e – 07 Test Failed
4	5000 10000	102/3.30 178/12.09	1.8e – 13/5.5e – 07 3.5e – 14/5.8e – 07	221/16.37 220/33.40	8.5e – 15/9.0e – 07 3.7e – 13/7.6e – 07	Test Failed Test Failed	Test Failed Test Failed
5	5000 10000	195/8.76 806/79.81	1.5e02/8.0e – 07 3.1e02/1.4e – 06	234/13.52 240/26.53	1.5e02/1.2e – 06 3.1e02/1.4e – 06	153/2.88 71/6.13	1.5e02/4.4e– 08 3.1e02/2.8e – 06
6	5000 10000	93/1.64 71/2.78	2.8e – 09/4.6e – 07 1.2e – 12/8.4e – 08	3/0.29 3/0.55	6.1e – 60/3.1e – 45 1.2e – 59/4.4e – 45	9/0.28 Test Failed	4.7e – 01/8.0e – 01 Test Failed
7	5000 10000	189/3.52 97/2.64	6.4e – 02/3.8e – 01 5.2e– 14/6.8e – 07	36/1.21 36/2.37	6.1e – 15/2.2e – 07 9.7e– 17/2.8e – 08	233/2.70 1188/33.62	3.3e – 14/9.3e – 07 3.0e– 07/3.1e – 03
8	5000 10000	82/1.78 82/3.49	4.0e00/5.5e – 07 4.0e00/5.5e – 07	113/3.63 113/7.98	4.0e00/6.2e – 07 4.0e00/6.2e – 07	77/1.64 77/3.20	4.0e00/2.5e – 07 4.0e00/2.5e – 07
9	5000 10000	481/10.37 162/6.76	5.0e03/3.4e – 05 1.0e04/1.3e – 05	563/11.29 626/33.44	5.0e03/7.9e – 07 1.0e04/1.2e – 06	788/30.25 372/10.93	5.0e03/4.4e – 07 1.0e04/3.8e – 06
10	5000 10000	131/2.27 1424/25.82	6.4e03/2.7e – 06 6.4e03/2.1e – 07	348/6.42 129/5.62	6.4e03/1.1e – 06 1.3e04/4.0e – 06	673/9.59 137/3.93	6.4e03/5.5e – 06 1.3e04/7.7e – 06

Table 3: Numerical Results for Modified Armijo Line Search Procedure.

FN	n	FR		PRP		HS	
		I/Ext	$f^*/\|g^*\|$	I/Ext	$f^*/\|g^*\|$	I/Ext	$f^*/\|g^*\|$
1	5000 10000	107/1.25 520/11.59	8.7e – 15/3.5e – 0.7 3.1e– 13/7.9e – 07	174/3.74 178/6.94	1.79e – 13/9.0e – 07 4.14e– 15/5.9e – 07	85/0.90 87/1.59	7.2e – 16/7.6e – 07 4.6e– 16/6.1e – 07
2	5000 10000	25/0.56 27/0.88	1.2e – 15/ 9.7e – 07 4.00e – 16/5.6e – 07	37/0.90 37/1.59	2.3e – 16/4.0e – 07 4.7e – 16/5.6e – 07	Test Failed Test Failed	Test Failed Test Failed
3	5000 10000	53/0.83 53/1.48	6.0e– 15/5.7e – 07 1.2e – 14/8.2e – 07	34/0.73 34/1.28	3.5e– 15/6.2e – 07 7.0e – 15/8.8e – 07	61/1.06 Test Failed	7.4e – 15/6.2e – 07 Test Failed
4	5000 10000	102/3.37 178/11.98	1.8e – 13/5.5e – 07 3.5e – 14/5.8e – 07	221/16.38 220/33.86	8.4e – 15/9.0e – 07 3.7e – 13/7.6e – 07	Test Failed Test Failed	Test Failed Test Failed
5	5000 10000	195/4.80 909/50.88	1.5e02/8.0e – 07 3.1e02/2.8e – 07	234/13.52 240/26.53	1.5e02/1.2e – 06 3.1e02/1.4e – 06	53/1.68 71/3.16	1.5e02/4.4e– 08 3.1e02/2.8e – 06
6	5000 10000	93/1.00 71/1.48	2.8e – 09/4.6e – 07 1.2e – 12/8.4e – 08	3/0.32 3/0.42	6.1e – 60/3.1e – 45 1.2e – 59/4.4e – 45	9/0.23 Test Failed	4.5e – 01/8.0e – 01 Test Failed
7	5000 10000	189/1.85 97/1.55	6.4e – 02/3.8e – 01 5.2e– 14/6.8e – 07	36/0.78 36/1.25	6.1e – 15/2.2e – 07 9.7e– 17/2.8e – 08	233/1.53 2896/39.18	3.3e – 14/9.3e – 07 5.2e– 14/8.7e – 07
8	5000 10000	82/1.40 82/1.99	4.0e00/5.5e – 07 4.0e00/5.5e – 07	113/2.00 113/4.15	4.0e00/6.2e – 07 4.0e00/6.2e – 07	77/0.78 77/1.75	4.0e00/2.5e – 07 4.0e00/2.5e – 07
9	5000 10000	481/5.35 162/2.96	5.0e03/3.4e – 05 1.0e04/1.3e – 05	563/5.65 626/12.99	5.0e03/7.9e – 07 1.0e04/1.2e – 06	788/14.87 372/5.72	5.0e03/4.4e – 07 1.0e04/3.8e – 07
10	5000 10000	301/3.31 99/1.85	6.4e03/3.7e – 06 1.3e04/3.1e – 06	2037/16.85 3005/46.20	6.4e03/9.7e – 07 1.3e04/8.0e – 07	701/4.99 137/2.14	6.4e03/5.0e – 06 1.3e04/7.7e – 06

Table 4: Numerical Results for Strong Wolfe Line Search Procedure.

FN	n	FR		PRP		HS	
		I/Ext	$f^*/\|g^*\|$	I/Ext	$f^*/\|g^*\|$	I/Ext	$f^*/\|g^*\|$
1	5000 10000	107/2.49 520/26.96	8.7e–15/8.9e–0.7 3.1e–13/7.9e–07	174/8.74 178/17.54	1.8e–13/9.0e–07 4.1e–15/5.9e–07	85/1.52 87/2.75	7.2e–16/7.6e–07 4.6e–16/6.1e–07
2	5000 10000	17/0.50 17/0.92	1.0e–16/2.4e–07 2.0e–16/3.3e–07	37/1.31 37/2.49	2.3e–16/4.0e–07 4.7e–16/5.6e–07	Test Failed Test Failed	Test Failed Test Failed
3	5000 10000	53/0.93 53/1.72	6.0e–15/5.7e–07 1.2e–14/8.0e–07	97/4.12 97/8.25	2.1e–15/6.6e–07 4.1e–15/9.3e–07	61/1.15 Test Failed	7.4e–15/6.2e–07 Test Failed
4	5000 10000	102/4.65 178/17.00	1.8e–13/5.5e–07 3.5e–14/5.8e–07	221/29.25 220/61.34	8.4e–15/9.0e–07 3.7e–13/7.6e–07	Test Failed 115/15.14	Test Failed 2.4e–14/7.7e–07
5	5000 10000	362/32.83 882/147.48	1.5e02/3.7e–07 3.1e02/3.6e–06	259/23.14 298/51.09	1.5e02/1.6e–07 3.1e02/8.7e–07	71/5.80 Test Failed	1.5e02/9.5e–07 Test Failed
6	5000 10000	93/1.54 71/2.96	2.8e–09/4.6e–07 1.2e–12/8.4e–08	3/0.49 3/0.69	6.1e–60/3.1e–45 1.2e–59/4.4e–45	Test Failed 17/1.06	Test Failed 8.6e–01/6.0e–01
7	5000 10000	182/4.50 1246/54.32	2.6e–14/6.1e–07 2.2e–13/9.7e–07	95/3.84 100/7.21	1.2e–15/1.0e–07 4.7e–14/9.3e–07	883/13.19 Test Failed	3.1e–14/1.0e–06 Test Failed
8	5000 10000	81/2.21 81/4.39	4.0e00/9.5e–07 4.0e00/9.5e–07	113/4.84 113/10.22	4.0e00/8.5e–07 4.0e00/8.5e–07	77/2.05 77/4.00	4.0e00/2.5e–07 4.0e00/2.5e–07
9	5000 10000	5596/189.19 1700/92.35	5.0e03/1.3e–06 1.0e04/1.3e–06	1044/13.38 712/27.17	5.0e03/3.9e–07 1.0e04/7.8e–07	556/6.16 873/19.81	5.0e03/7.1e–07 1.0e04/6.5e–07
10	5000 10000	2737/56.81 99/3.47	6.4e03/9.3e–07 1.3e04/3.1e–06	2037/31.36 3005/88.79	6.4e03/9.8e–07 1.3e04/8.0e–07	701/9.18 123/3.34	6.4e03/5.1e–06 1.3e04/8.0e–06

REMARKS

From the above tables, we have summarized our observations in order to simplify our inference. Wherever a decrease in the objective function fails to occur or the method fails for a particular function, the value "Test Failed" is assigned.

Table 5: Inference on Armijo Line Search Rule (ALSR)

Inferential Parameter	FR	PRP	HS
Average of I/Ext	242.35/9.06	168.95/8.84	286.50/7.75
Solvability Ratio	20/20	20/20	14/20
Acceptability Ratio	15/20	15/20	8/14

Table 6: Inference on Modified Armijo Line Search Rule (MALSR)

Inferential Parameter	FR	PRP	HS
Average of I/Ext	191.30/5.52	397.20/9.80	403.36/5.68
Solvability Ratio	20/20	20/20	14/20
Acceptability Ratio	15/20	17/20	10/14

Table 7: Inference on Strong Wolfe Line Search Rule (SWLSR)

Inferential Parameter	FR	PRP	HS
Average of I/Ext	708.85/32.36	442.15/19.76	286.31/6.55
Solvability Ratio	20/20	20/20	13/20
Acceptability Ratio	16/20	20/20	10/13

Table 8: Comparing Rules

Inferential Parameter	$ALSR$	$MALSR$	$SWLSR$
Average of I/Ext	697.80/25.65	991.86/21.00	1437/58.67
Solvability Ratio	54/60	54/60	53/60
Acceptability Ratio	38/54	42/54	46/53

It is clear from the above tables that the strong Wolfe line search rule performs better than the other two rules because more it generated the highest number of results that satisfy the stopping criterion$\|g_k\|_\infty \leq 10^{-6}$. It is evident that, though the average number of iterations generated with the Modified ALSR is higher than that of ALSR itself, the average execution time has however reduced.

In the future we hope to focus more attention on the stopping criteria. Several of them are known in literature but the task still remains to find out which is more suitable or to formulate a more robust criteria that suits a wide range of problems.

REFERENCES

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

Andrei, N. (2008). New accelerated conjugate gradient algorithms for unconstrained optimization. ICI Technical Report.

Andrei, N. (2011). Open problems in nonlinear conjugate gradient algorithms for unconstrained optimization. Bull. Malays. Math. Sci. Soc. (2), 34(2), 319-330.

Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives, Pacific J. Math. 16 (1966), 1-3.

Bamigbola, O.M and Ejieji, C.N (2006). A conjugate gradient method for solving discrete Optunal control problems. The Journal of Mathematical Association of Nigeria (M.A.N) vol.33, No213 p.318-329.

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

Beale, E. M. L. (1971). A derivation of conjugate gradients, in Numerical Methods for Non-linear Optimization (Conf., Univ. Dundee, Dundee). 39-43, Academic Press, London.

Birgin, F. G. and Martinez, J. M. (2001) 'A spectral conjugate gradient method for unconstrained optimization. Journal of Applied Mathematics and Optimization, pp.117-128.

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.

Daniel, J. W. (1967). The conjugate gradient method for linear and nonlinear operator equations. SIAM J. Numer. Anal. 4, 10-26.

Dennis, J. E. and Schnabel, R. B. (1983). Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall Series in Computational Mathematics, Prentice Hall, Englewood Cliffs, NJ.

Dixon, L. C. W., Ducksbury, P. G. and Singh, P. (1985). A new three-term conjugate gradient method. J. Optim. Theory Appl. 47, no. 3, 285-300.

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

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

Goldstein, A. A. (1965). On steepest descent. J. Soc. Indust. Appl. Math. Ser. A Control 3, 147-151.

Hager, W. W. and Zhang, H. (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. 16, no. 1, 170-192.

Hager, W. W. and Zhang, H. (2006). A survey of nonlinear conjugate gradient methods. Pac. J. Optim. 2, no. 1, 35-58.

Hauser, R. (2007). Line search methods for unconstrained optimization. Lecture 8, Numerical Linear Algebra and Optimization. Oxford University Computing Laboratory.

Hestenes, M.R. and Stiefel, E. (1952). Method of conjugate gradient for solvinglinearequations. J. Res. Nat. Bur. Stand. 49.

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

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

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

Potra, F. A. and Shi, Y. (1995). Efficient line search algorithm for unconstrained optimization, J. Optim. Theory Appl. 85 (1995), no. 3, 677-704.

Powell, M. J. D. (1975). Some global convergence properties of a variable metric algorithm for minimization without exact line searches, in nonlinear programming. 53-72. SIAM-AMS Proc., IX, Amer. Math. Soc., Providence, RI.

Reid, J. K. (1971).On the method of conjugate gradient for the solutionof large sparse sets of Linear Equations, Large sparse sets of LinearEquation (London and New York). AcademicPress London and New York, pp. 231-254.

Rockafellar, R.T. (1970). Convex analysis Princeton: Princeton University Press.

Wolfe, P. (1969). Convergence conditions for ascent methods. SIAM Rev. 11, 226-235.

Yabe, H. and Takano, M. (2004). Global convergence properties of nonlinear conjugate gradient methods with modified secant condition. Comput. Optim. Appl. 28, no. 2, 203-225.

Yuan, G. and Lu, X. (2009). A modified PRP conjugate gradient method.Ann. Oper. Res. 166, 73-90.

Zoutendijk, G. (1970).Nonlinear programming, computational methods in integerandnonlinearprogramming. J. Jabadie, Ed., pp. 37–86, North-Holland, Amsterdam, The Netherlands.