Qatar Univ. Sci J. (1997), 17(2): 205 - 215

NEW SWITCHING ALGORITHM FOR COMBINING NEW MULTI-STEP CG AND SELF-SCALING VM ALGORITHMS FOR NONLINEAR OPTIMIZATION

By

Abbas Y. Al Bayati,

Dept. of Mathematics, College of Science

University of Mosul, Iraq.

Muna M. Mohammad Ali,

Dept. of Mathematics, College of Science

University of Mosul, Iraq.

خوارزميتان جديدتان في مجال الأمثلية غير الخطية

عباس البياتي و منى محمد علي

قسم الرياضيات، كلية العلوم، جامعة الموصل - العراق

في هذا البحث تم استحداث خوارزميتين جديدتين في مجال الأمثلية غير الخطية الأولى هي خوارزمية التدرج المترافق متعدد الخطوات والثانية هي خوارزمية المتري المتغير ذاتي القياس وباستعمال خطوط بحث غير تامة.

تم مقارنة الخوارزميات الجديدة بمثيلاتها من الخوارزميات في هذا المجال مع الحصول على نتائج مشجعة جداً وباستعمال عدد كبير من الدوال غير الخطية ذات أبعاد مختلفة.

Key Words : Switching algorithm, Monlinear Optimization.

ABSTRACT

New interleaved CG and self-scaling VM algorithm is presented in this research which combines Al Bayati's self-scaling VM algorithm with another new multi-step CG algorithm with inexact line searches (ILS). In an interleaving algorithm a VM-update can be initiated between each CG-step and the technique improves the rate of convergence of the new proposed algorithms. The new algorithms are tested against the standard Hestenes-Stiefel and Buckley's algorithms for a number of well-known test functions, with encouraging results.

1 - INTRODUCTION

Two of the most important classes of algorithms for function minimization are Variable Metric (VM) and Conjugate Gradient (CG) algorithm. It is well-known that VM algorithms require fewer iterations and do not require very accurate line searches. On the other hand, CG algorithms do not require matrix storage and have proved themselves appropriate for large problems where the matrix of second derivatives is not spare (see for example Fletcher, [1]. However, the class of interleaved algorithms which incorporate both a CG and a VM algorithms are "intermediate" in that they use only few storage locations and they accelerate the rate of convergence of CG algorithms. We now define explicitly the general VM and CG algorithms, and hence the new algorithms with their inexact line search.

2. WELL-KNOWN RESULTS:

Let $f : R^n \to R$ be a twice differentiable function defined on an open set E. We consider the problem of finding $x \in E$ such that:

$$\min_{x \in E} f(x) = f(x_{min}) \qquad (1)$$

$$\text{and} \quad v_k = x_{k+1} - x_k \qquad (3)$$

Then the Broyden [2] θ-class VM-family can be expressed as follows: given X_1 and an arbitrary nxn positive definite matrix H_1, iterate for $k = 1,2,3,\ldots$ with

$$x_{k+1} = x_k - \lambda_k H_k g_k , \qquad (4)$$

where λ_k is a steplength and H_k is defined by:

$$H_{k+1} = H_k - \frac{H_k \gamma_k \gamma_k^T}{\gamma_k^T H_k \gamma_k} + \theta_k \omega_k \omega_k^T + \rho_k \frac{v_k v_k^T}{v_k^T \gamma_k}$$

where:

$$\omega_k = (\gamma_k^T H_k \gamma_k)^{1/2} \; [(v_k v_k^T) - (H_k \gamma_k / \gamma_k^T H_k \gamma_k)] \quad (6)$$

and θ_k, ρ_k are parameters ≥ o. The well known BFGS (Broyden, Fletcher, Goldfarb and Powell) update corresponds to $\rho_k = 1$; $\theta_k = 1$, while the DFP (David, Fletcher and Powell) update arises from $\rho_k = 1$; $\theta_k = 0$. In this research we are mainly concerned with Al Bayati's (1991) self-scaling VM update, for which $\theta_k = 1$,

$$\rho_k = (\gamma_k^T H_k \gamma_k) / v_k^T \gamma_k .$$

ALGORITHM I (Hestenes and Stiefel):

Now, the CG-method was originally developed by Hestenes and Stiefel, (1952) to solve systems of linear equations; it has been updated to solve minimization problem (1) in the following form given arbitrary x_1, let $d_1 = -g_1$, iterate for $k = 1, 2, 3, \ldots$ with

$$d_{k+1} = -g_{k+1} + \beta_k d_k , \qquad (7a)$$

$$x_{k+1} = x_k + \lambda_k d_k , \qquad (7b)$$

where λ_k minimizes f in the direction d_k. The scalar λ_k is called an exact minimizer (ILS) if:

$$f(x_k + \lambda_k d_k) \le f(x_k + t d_k) \text{ for all } t \in (0, \omega). \qquad (8)$$

Different choice of the parameter β_k gives several different algorithms, including:

$$\beta_k = \begin{cases} (g_{k+1}^T g_{k+1}) / (g_k^T g_k) & \text{Fletcher-Reeves (FR)} \quad (9) \\ (g_{k+1}^T g_k) / (\gamma_k^T g_k) & \text{Polak- Ribiere (PR)} \quad (10 \\ (g_{k+1}^T \gamma_{k+1}) / (d_k^T \gamma_k) & \text{Hestenes-Stiefel (HS)} \quad (11) \end{cases}$$

We note that for a quadratic function the three formulae for β_k are the same provided ELS are used (see Fletcher, 1987).

3. MULTI-STEP CG-METHODS:

Let f be a strictly convex quadratic function defined on R^n:

$$f(x) + (x^T A \; x)/2 + b^T x , \qquad (12)$$

where A is a symmetric and strictly positive definite nxn matrix and $b \in R^n$. A set of non-zero vectors $(d_1, d_2, \ldots\ldots\ldots, d_n)$ is defined as mutually conjugate with respect to A if:

$$d_i^T A \, d_k = 0 \text{ for all } i \neq k \qquad (13)$$

Now, The CG method can be looked upon as being a particular specialisation of the Gram-Schmidt orthogonalisation of a given set of vectors. The Gram-Schmidt process can be expressed:

$$d_1 = -g_1 \qquad (14)$$

and for $k = 1, 2, 3, \ldots$ iterate with

$$d_{k+1} = -g_{k+1} = +\sum_{i=1}^{k} \beta_k d_k . \qquad (15)$$

For the conjugacy condition to hold for the set d with respect to A then

$\beta_{ik} = (\gamma_i^T g_{k+1}) / (\gamma_i^T d_i);\quad i = 1, 1, 2, 3, \ldots\ldots, k.$ (16)

Particular CG-algorithms that do not require ELS have been introduced by Dixon (1975); Nazareth (1977); Shanno (1978). Nazareth-Nocedal (1978) developed, a multistep CG-method which does not need ELS by defining matrices

$D = (d_1, d_2 \ldots\ldots\ldots, d_n)$

and $G = (g_1, g_2 \ldots\ldots\ldots, g_n)$ and expressing (15) by

$-G = DB,$ (17)

where B is an nxn upper triangular matrix with $\beta_{ii} = 1$ for $i=1, 2, \ldots\ldots\ldots, k$. Using ELS in (15) yields:

$\gamma_i^T g_{k+1} = 0 \quad \text{for} \quad i = 1, 2, \ldots\ldots, k-1,$ (18)

and hence

$\beta_{ik} = 0 \quad \text{for} \quad i = 1, 2, 3, \ldots\ldots, k-1.$ (19)

Hence, the conjugate search directions of the multi-step CG-method can be summerized as follows:

ALGORITHM II (Nazareth-Nocedal):

$d_1 = -g_1,$

iterate for $k = 1, 2, 3, \ldots\ldots,$ with

$\rho_{k+1} = -g_{k+1} + (\gamma_k^T g_{k+1}) / (\gamma_k^T d_k)\, d_k,$ (20a)

$d_{k+1} = -p_{k+1} + c_k,$ (20b)

$(c_{k-1} + [(\gamma_{k-1}^T g_{k-1}) / (\gamma_{k-1}^T d_{k-1})]\, d_{k-1}; \quad k > 1,$

$c_k = 10 \quad ; \quad k = 1$ (20c)

In the above algorithm using ILS not all the coefficients of the Gram-Schmidt process have to be computed at every iteration. For the $(k+1)^{th}$ step, when computing d_{k-1} the coefficients for $d_1\, d_2, d_2, \ldots\ldots\ldots, d_{k-2}$ are already known since (see Nazareth-Nocedal 1978) it is established that:

$\gamma_i^T \gamma_{i+k} = 0 \quad \text{for} \quad k \geq 2.$ (21)

Hence

$\gamma_i^T \gamma_{i+2} = \gamma_i^T (g_{i+3} - g_{i+2}) = 0$ (22a)

$\gamma_i^T \gamma_{i+3} = \gamma_i^T (g_{i+4} - g_{i+3}) = 0$ (22b)

and so

$\gamma_i^T g_{i+2} = \gamma_i^T g_{i+3} = \gamma_i^T g_{i+4} = \ldots\ldots\ldots$ (22c)

Hence, only two new coefficients have to be computed and only two previous search directions must be stored; the contribution of the components along $d_1, d_2, \ldots\ldots\ldots, d_{k-2}$ to the new direction d_{k+1} can be accumulated in a single vector c_k. Similarly, the correction vector (e_k) to the current iterate x_k can also be accumulated in a single vector such that for the quadratic function (12) the correction term is defined by:

$e_{k+1} = e_k + e_k\, d_k \quad \text{for} \quad k \geq 1;$ (23)

and so

$x_{min} = x_{n+1} + e_{n+1},$ (24)

where

$e_k = -\lambda_k [(g_{k+1}^T d_k) / (\gamma_k^T d_k)]$, with $e_1 = 0$ initially (25)

for further details see Nazareth (1977).

Algorithm II generates mutually conjugate search directions with respect to the Hessian matrix A for the system of equations defined in (20). For more details see Nazareth and Nocedal (1978).

ALGORITHM III (NEW1):

However, in this research we have assumed that A=I, the identity matrix, Eq. (20) generates I-conjugate vectors, i.e. they are also multually orthogonal. Based on the above idea we have constructed a new set of mutually orthogonal vectors $g_1^*, g_2^*, \ldots\ldots, g_n^*$ which are a linear combination of the normal gradient terms $g_1, g_2, \ldots\ldots, g_n$ defined as follows:

$g_1^* = g_1,$ (26a)

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

$g_k^* = g_k - [(g_k^T g_{k-1}^*) / (g_{k-1}^{*T} g_{k-1}^*)]\; g_{k-1}^*,$ (26b)

$g_k^* = g_k^* + c_{k-1},$ (26c)

$$c_{k-1} = \begin{cases} (c_{k-2} + [(g_k^T g_{k-2}^*) / (g_{k-2}^{*T} g_{k-2}^*)]\, g_{k-2}^*; & k = 3,4,.. \\ 0 & k = 1, 2 \end{cases} \quad (26d)$$

We now will prove that the set of orthogonal vectors defined in (26) will estimate the set of normal gradient vectors as follows:

The mutually conjugate search directions for algorithm III can be expressed as follows:

$d_1 = -g_1^*,$ (27a)

iterate for $k = 1, 2, 3, \ldots\ldots$ with

$$d_{k+1} = -g^*_{k+1} + \beta^*_k d_k \tag{27b}$$

where

$$\beta^*_k = (g^{*T}_{k+1} \gamma^*_k) / (d^T_k \gamma^*_k) \tag{27c}$$

and $$\gamma^*_k = g^*_{k+1} - g^*_k \tag{27d}$$

where g^*_{k+1} is a vector defined in (26).

However, this algorithm also generates mutually conjugate directions for the quadratic function (12) and the set $\{g^*_1, g^*_2, \ldots, g^*_n\}$ is identical with the normal set $\{g^*_1, g^*_2, \ldots, g^*_n\}$ provided that ELS are used (see theorem 1).

THEOREM 1: The set of orthogonal vectors $\{g^*_i : i=1, 2, \ldots, n\}$ defined in (26) coincide with the set $\{g_i : i=1, 2, \ldots, n\}$ in the case of the quadratic function (12) using ELS.

Proof : Proceed by induction:

$g^*_1 = g_1$, by assumption.

For $i = 2$, we have

$$g^*_2 = g_2 - [\,(g^T_2 g^*_1) / (g^{*T}_1 g^*_1)\,]\, g^*_1 \tag{28}$$

since $g^*_1 = g_1$ and $g^T_2 g_1 = 0$ (orthogonality property is satisfied for the quadratic function with ELS); hence

$g^*_2 = g_2$.

Suppose now that the theorem is true for i, i .e.

$$g^*_j = g_j, \quad \text{for } j = 1, 2, 3, \ldots, i. \tag{29}$$

To complete the proof of the theorem we must show it is true for i+1. Consider

$$208 = g_{i+1} - \frac{g^T_{i+1} g^*_i}{g^{*T}_i g^*_i} g^*_i + \frac{g^T_{i+1} g^*_{i-1}}{g^{*T}_{i-1} g^*_{i-1}} g^*_{i-1} + \ldots + \frac{g^T_{i+1} g^*_1}{g^{*T}_1 g^*_1} g^*_1. \tag{30}$$

Since all the gradients are mutually conjugate with respect to A in the set $\{g_i : i = 1, 2, 3, \ldots, n\}$ and (29) is true, then clearly

$g^*_{i+1} = g_{i+1}$.

For ILS and for general functions, the set $\{g^*_i : i = 1, 2, 3, \ldots,\}$ will only estimate the normal set of gradient vectors $\{g_i : i = 1, 2, 3, \ldots, n)$. For the quadratic function (12) the search directions defined in (27) are identical to the standard CG-method provided that ELS are used. But for general functions, algorithm III restarts with the stepest descent direction every n interation or whenever Powell's (1977) restarting criterion is satisfied, i.e. if either of the inequalities:

$$\left| g^{*T}_{k+1} g^*_k \right| \geq 0.2 \left| g^{*T}_{k+1} g^*_{k+1} \right| \tag{31a}$$

or

$$d^T_{k+1} g^*_{k+1} > -0.8\,(g^{*T}_{k+1} g^*_{k+1}) \tag{31b}$$

are satisfied.

4. NEW INTERLEAVING VM-CG METHOD ALGORITHM (IV):

In this section we have to present another new algorithm which uses both VM and CG-steps. Algorithm (IV) is particularly suitable for cases where less than nxn storage locations are conveniently available to store the VM matrix, and they perform VM-steps and CG-steps sequentially. This type of algorithms was originally studied by Buckley (1978).

For the outlines of the this new algorithm we consider the general iteration:

$$d_1 = -Hg_1. \tag{32a}$$

where g_1 is arbitrary, and H is any symmetric and positive definite matrix, iterate for $k = 1, 2, 3, \ldots$ with

$$d_{k+1} = -Hg^*_{k+1} + \beta^*_k d_k, \tag{33a}$$

$$x_{k+1} = x_k + \lambda_k d_k, \tag{33b}$$

$$\beta^*_k = (g^{*T}_{k+1} H \gamma^*_k) / (d^T_k \gamma^*_k), \tag{33c}$$

where λ_k is a steplength determined by ILS.

The new interleaved algotrithm uses iteration (33) and Al-Bayati's (1991) self-scaling VM updates as follows:

Given arbitrary x_1 and matrix H_1 (usually $H_1 = 1$), set t =1, i = 1 initially.

step 1: set $d_t = -H_i g_t$.

step 2: for $k = t, t+1, t+2, \ldots$ iterate with (33)

step 3: check if

$$\left| d^T_k \gamma_k \right| > 0.0015 \left\| y_k \right\| \cdot \left\| d_k \right\| \tag{34}$$

is satisfied then reset t to the current k. Eq (34) was given by Dixon (1985).

Update H_i by

$$H_{i+1} = H_i - \frac{H_i \gamma_t v^T_t}{v^T_t \gamma_t} + v_t \left(2 \frac{\gamma^T_t H_i \gamma_t}{v^T_t \gamma_t} - \frac{H_i \gamma_t}{v^T_t \gamma_t}\right)^T \tag{35}$$

step 4: replace i by i+1 and repeat from step 1.

For the VM-steps it is necessary that $v^T_t \gamma_t > 0$ to ensure that H_{i+1} will be positive definite. Replace this condition by the equivalent one:

$$d_{k+1}^T g_{k+1} < \mu_k d_{k+1}^T g_k , \quad (36)$$

where μ_k is constant less than 1. Hence, we adopt the condition (36) in implementing our new algorithm but use it explicitly with $\mu_k = 0.9$ which corresponds to the best value for ILS in this case. Buckley (1978b) also proved that his combined QN-CG algorithm and the standard CG method generate identical sequences of points x_k, if the two algorithms start from the same starting point. This means that the change in the metric along the QN-step does not prevent the mixed algorithm from terminating in n steps or less in the case of quadratic function. provided that the BFGS update is used.

For this new algorithm it is clear that Al-Bayati's (1991) self-scaling VM update (7) will not effect the quadratic termination property.

ALGORITHM (V) BUCKLEY:

To measure the performance of the new propsoed interleaving algorithm (IV) we have compared it with Buckley's (1978a) interleaving algorithm. Buckley's algorithm differes than the new algorithm in two major steps:

First: Buckley's algorithm uses the BFGS update while the new algorithm uses Al-Bayati's self-scaling VM-update.

Second: Buckley's switching criterion is

$$\left| g_{k+1}^T g_k \right| > 0.2 \left| g_{k+1}^T g_{k+1} \right| \quad (37)$$

while the new algorithm uses eq (34) as new switching criterion.

5. NUMERICAL COMPARISONS.

Thirty standard test functions (see Appendix) were tried with a range of dimensions in order to examine the effectiveness of the new proposed algorithms. The numerical experiments were performed on the IBM-Elonex personal computer with double precision arithmetic with programs written in FORTRAN. The line search routine used was a cubic interpolation which uses function and gradient values and it is an adaptation of the routine published in (12).

The following five algorithms were tested: the first (HS) corresponds tot he standard Hestenes and Stiefel CG-method, the second (NEW1) is the new multi-step CG-method, the third is Buckley's method. (BUCKLEY), the fourth is the second new interleaving algorithm (NEW2) and the fifth is Al-Bayati's self-scaling VM-method (BAYATI). The numerical results for these five algorithms are presented in four tables. The performance indicators employed are; total number of function calls (NOF), and total number of iterations (NOI) required to solve each test function, using the following stopping criterion.:

$$(g_{k+1}^T g_{k+1}) < 1\text{x}10^{-5} \quad (38)$$

Finally, the computational results documented in this section show that the new algorithms generally performs more efficiently than the other algorithms, in terms of NOF, for most of our test functions. The new algorithms almost always require a lower number of iterations and function calls than the well-know standard HS-method with overall savings in the range (12-25)% for NOF and (11-39)% for NO1 for small dimensionality test functions (see Table-1-). Now for moderate dimensionality test functions there are (10-35)% for NOF and (27-50)% for NO1. For large dimensionality test functions there are improvements in the NOF about (35-38)% and (44-66)% for NO1. However, the new interleaved algorithms uses only moderate storage and they generally accelerate the CG-method. However, CG-methods remain valuable for the very large problems since only few vector storage locations are required.

TABLE (1)

TEST FUNCTIONS	N	HS	NEW1	BAYAT1	BUCKLEY	NEW2
		NO1 (NOF)	NO1 (NOF)	NO1 (NOF)	NO1 (NOF)	NO1 (NOF)
CUBIC	2	14 (53)	19 (52)	13 (36)	14 (47)	14 (48)
BEALE	2	10 (26)	10 (26)	9 (23)	20 (70)	9 (30)
EDGAR	2	6 (20)	6 (18)	7 (16)	7 (17)	5 (18)
RECIPE	3	5 (16)	5 (16)	7 (16)	6 (16)	6 (21)
POWLL3	3	14 (30)	10 (24)	11 (22)	10 (30)	11 (33)
BIGGS	3	14 (42)	14 (52)	12 (30)	15 (48)	10 (40)
HELICAL	3	32 (74)	32 (67)	20 (67)	19 (66)	19 (62)
POWELL	4	65 (170)	46 (122)	32 (72)	17 (73)	10 (57)
WOOD	4	26 (60)	29 (70)	22 (51)	38 (114)	22 (72)
CANTERLL	4	25 (148)	16 (115)	14 (61)	19 (93)	16 (96)
TOTAL	NO1 (NOF)	211 (639)	187 (562)	147 (394)	165 (574)	128 (477)

PERFORMANCE OF THE NEW ALGORITHMS IN RELATION TO STANDARD HS-CG METHOD

	HS	NEW1	BAYATI	BOCKLEY	NEW2
NO1	100	88.6	69.6	78.1	60.6
NOF	100	87.9	61.6	89.8	71.6

ALL the algorithms terminate when $\| g^* \| < 1\mathrm{x}10^{-5}$

TABLE (2)

TEST FUNCTIONS	N	HS	NEW1	BAYAT1	BUCKLEY	NEW2
		NO1 (NIF)	NO1 (NOF)	NO1 (NOF)	NO1 (NOF)	NO1 (NOF)
DIXON	10	22 (50)	30 (62)	15 (31)	17 (40)	16 (49
SHALLOW	20	8 (19)	7 (19)	8 (18)	9 (28)	8 (27)
MIELE	20	54 (141)	38 (122)	32 (89)	25 (84)	29 (102)
EX-CANTRL	20	20 (132)	17 (120)	15 (74)	19 (93)	16 (96)
EX-POWELL	20	60 (162)	47 (129)	38 (79)	39 (119)	17 (60)
PN2	30	16 (40)	18 (146)	29 (64)	38 (109)	16 (50)
EX-MIELE	40	82 (197)	43 (141)	34 (94)	28 (90)	29 (102)
EX-CANTRL	40	20 (132)	17 (126)	15 (74)	19 (93)	16 (96)
EX-POWELL	40	85 (213)	47 (129)	41 (85)	66 (198)	18 (65)
FULL	40	50 (100)	39 (79)	41 (81)	53 (107)	40 (120)
TOTAL	NO1 (NOF)	417 (1185)	303 (1073)	268 (689)	313 (961)	205 (767)

PERFORMANCE OF THE NEW ALGORITHMS IN RELATION TO STANDARD HS-CG METHOD

	HS	NEW1	BAYATI	BOCKLEY	NEW2
NO1	100	72.6	64.2	75.1	49.1
NOF	100	90.5	58.1	81.1	64.7

ALL the algorithms terminate when $\| g^* \| < 1x10^{-5}$

TABLE (3)

TEST FUNCTIONS	N	HS	NEW1	BAYAT1	BUCKLEY	NEW2	BFGS
		NO1 (NIF)	NO1 (NOF)	NO1 (NOF)	NO1 (NOF)	NO1 (NOF)	NO1 (NOF)
EX-POWELL	60	125 (303)	47 (129)	45 (96)	72 (206)	21 (70)	71 (203)
FREUD	60	6 (18)	10 (25)	7 (28)	16 (51)	6 (22)	15 (49)
STRAIT	70	6 (20)	6 (18)	6 (15)	5 (20)	5 (18)	5 (20)
EX-POWELL	80	112 (303)	47 (129)	40 (83)	63 (169)	18 (63)	60 (165)
EX-CANTRLL	80	20 (132)	18 (137)	15 (74)	19 (93)	17 (110)	22 (95)
WOLFE	80	50 (99)	47 (95)	42 (83)	63 (127)	41 (123)	65 (130)
EX-RECIPE	90	12 (33)	6 (18)	7 (16)	6 (16)	6 (21)	6 (16)
PNI	90	8 (25)	10 (41)	8 (21)	10 (31)	8 (29)	10 (31)
EX-POWELL	100	105 (276)	48 (140)	44 (91)	62 (175)	18 (63)	60 (170)
EX-CUBIC	100	14 (40)	19 (52)	13 (36)	47 (118)	13 (43)	43 (115)
TOTAL	NO1 (NOF)	458 (1249)	258 (784)	227 (543)	363 (1006)	153 (562)	966 (1010)

PERFORMANCE OF THE NEW ALGORITHMS IN RELATION TO STANDARD HS-CG METHOD

	HS	NEW1	BAYATI	BUCKLEY	NEW2	BFGS
NO1	100	56.3	49.5	79.2	33.4	78.4
NOF	100	62.7	43.4	80.5	44.9	80.9

ALL the algorithms terminate when $\| g^* \| < 1x10^{-5}$

TABLE (4)

A statistical comparison between the new algorithm (NEW2) with [HS ; BAYATI; and NEW1 algorithm by using paired t-test.

N	ALGORITHMS	NO1 / NOF	SIGNIFICANCE LEVEL
2 - 4	HS	NO1 NOF	0.112614 0.214132
	NEW1	NO1 NOF	0.081546 0.243499
	BAYATI	NO1 NOF	0.269737 0.0861184
	BUCKLEY	NO1 NOF	0.0654013 0.116255
10 - 40	HS	NO1 NOF	0.0233992 0.0428506
	NEW1	NO1 NOF	0.0279254 0.0366562
	BAYATI	NO1 NOF	0.0610346 0.241029
	BUCKLEY	NO1 NOF	0.0619183 0242282
60 - 100	HS	NO1 NOF	0.0596574 0.9825033
	NEW1	NO1 NOF	0.0262952 00710468
	BAYATI	NO1 NOF	0.0729792 0.802641
	BUCKLEY	NO1 NOF	0.011004 0.0362787

For the above Table one sided test is considered when the significance level is less than 0.05 and we have consideredtwo sided tests otherwise, conventionally the levels are (see Robert and James 1980)

0.05 and less (significant)
0.01 and less (highly significant)
0.001 and less (very highly significant).

6. REFERENCES :

Broyden, C. G. (1970) "The convergence of a class of double-rank minimization algorithms", Journal of Institute of Mathematics and its Application, 6, pp. 76-90.

Buckley, A. G. (1978) "A combined conjugate gradient quasi-Newton minimization algorithm", Mathematical programming, 15, pp. 200-210.

Buckley, A. G. (1978) "Extending the relationship between the conjugate gradient and BEGS algorithm", Mathematical Programming, 15, pp. 343-348.

Bunday, B. (1984)" Basic Optimization Methods", Edward Arnold, London.

Dixon, L. C. W. (1975) "Conjugate gradient algorithms: quadratic termination without linear searches", Journal of Institute of Mathematics and its Application, 15, pp. 9-18.

Dixon, L.C.W. Ducksbury, P.G., Singh, p (1985), "A new three term conjugate gradient method", Technical Report No.130 (2nd issue).

Eletcher, R. and Reeves, C. M. (1964) "Function minimisation by conjugate gradients". Computer Journal, 7 pp. 149-154.

Fletcher, (1987) " practical methods of optimizations", John Wiley and sons, Chichester, New York.

Hestenes, H. R. and Stiefel, E. (1952) "Methods of conjugate gradient for solving linear systems", Journal of Research of the National Bureau of standards, 49, pp. 409-436.

Nazareth L. (1977) "A conjugatedirection algorithm without line searches", Journal of Optimization Theory and Application, 23, pp. 373-388.

Nazareth, L. and Nocedal, J. (1978) "Properties of conjugate gradient methods with inexact linear searches", Report 780-1, Systems Optimization Laboratory, Department of Operation Research, Standford University.

Polak, E. (1971) "Computational Methods in Optimization :a Unified Approach", Academic Press, London and New York.

Powell, M. J. D. (1977) " Restart Procedures for the conjugate gradient method", Mathematical Programming, 12, pp. 241-254.

Robert, G. D. S. and James, H. T. (1980" " Principles and procedures and statistics", A Biometrical Approach, International Student Edition.

Shanno, D. F. (1978)" Conjugate gradient methods with inexact searches", Mathematics of Operation Research, 3, pp. 244-256.

APPENDIX

All the presented test functions are from the general literature:

1- Cubic Function:

$f = 100(x_2 - x_1^3)^2 + (i-x_1)^2, \quad x_o = (-1.2,1)^T.$

2- Strait Function:

$f = [(x_{2i-1})^2 - x_{2i}]^2 + 100 (1 - x_{2i-1})^2$

3- Generalized Wood Function:

$f = \sum_{i=1}^{n/4} 100[(x_{4i-2} - x_{4i-3}^2)^2] + (1-x_{4i-3})^2 + 90(x_{4i} - x_{4i-1}^2)^2 + (1-x_{4i-1})^2 + 10.1 [(x_{4i-2} -1)^2 + (x_{4i}-1)^2] + 19.8 (x_{4i-2}-1)(x_{4i}-1), \quad x_o=(-3, -1, -3, -1 ;...)^T.$

4- Generalized Powell Function:

$f = \sum_{i=1}^{n/4} [x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1}-x_{4i})^2 + (x_{4i-2} -2x_{4i-1})^4 + 10(x_{4i-3}-x_{4i})^4], \quad x_o=(3, -1, 0, 1 ; ...)^T.$

5- Beale Function:

$f = (1.5-x_1 (1-x_2^2) + (2.25-x_1 (1-x_2^2)^2 + [2.625- x_1 (1-x_2^2)^2], \quad x_0=(0, 0)^T.$

6- Pen 1. (Penalty Function):

$f = \sum_{i=1}^{n}[(x_i-1)^2 + \exp(x_i^2- 0.25)^2], \quad x_0=(1, 2, ...,n)^T.$

7- Pen 2. (Generalized PenaltyFunction):

$f = \sum_{i=1}^{n}[\exp(x_i-1)^2 + (x_i^2-0.25)^2], \quad x_0=(1, 2, ... ,n)^T.$

8- Dixon Function:

$f = \sum_{i=1}^{n}[(1-x_1)^2 + (1-x_n)^2 + \sum_{i=1}^{n-1} (x2i-xi+1)2], \quad x0=(-1 ; ...)T.$

9- Generalized Edger & Himmel Function:

$f = \sum_{i=1}^{n} [(x_{2i-1}-2)^4 + (x_{2i}-2)^2 \times x_{2i}^2 + (x_{2i} +1)^2], \quad x_0=(1, 0; ...)^T.$

10- Generalized Recip Function:

$$f = \sum_{i=1}^{n/3} \left\{ (x_{3i-1} -5)^2 + x_{3i-1}^2 + \frac{x_{3i}^2}{(x_{3i-1} - x_{3i-2})^2} \right\},$$

$x_0=(2, 5, 1; ...)^T.$

11- Wolfe Function:

$f = [-x_1 (3-x_1/2) + 2x_2-1]^2 + \sum_{i=1}^{n-1} (x_{i-1} - x_i (3-x_i /2) + 2x_{i+1}-1]^2 + [x_{n-1} - x_n (3-x_n /2)-1]^2, \quad x_0=(-1, ...)^T.$

12- Biggs Function:

$f = \sum_{i=2}^{10} [\exp(-x_1z_i) - (x_3 \exp(-x_2z_i) - \exp(-z_i)+5 \exp(-10z_i)]^2,$

where $z_i = (0.1)^i$, and $x_0=(1, 2, 1,)^T.$

13- Generalized Freudenstien and Roth Function:

$f = \sum_{i=1}^{n/2} \left\{ [-3+x_{2i-1} + (5-x_{2i}) x_{2i}-2)x_{2i}]^2 + [29+x_{2i-1} + (1+x_{2i})x_{2i}-14x_{2i}]^2 \right\}, \quad x_0=(30, 3; ...)^T.$

14- Full Set of Distinct Eigen values Function:

$f = (x_1-1)^2 + \sum_{i=2}^{n} [2x_i - x_{i-1}]^2, \quad x_0 =(1; ...)^T.$

15- Generalized Miele Function:

$f = \sum_{i=1}^{n/4} [\exp(x_{4i-3} - x_{4i-1})^2 + 100(x_{4i-2}-x_{4i-1})^\sigma + \left(\tan (x_{4i-1}-x_{4i})\right)^4 + x_{4i-3}^3 + (x_{4i}-1)^2], \quad x_0=(1, 2, 2, 2; ...)^T.$

16- Generalized Helical Valley. Function:

$f = \sum_{i=1}^{n/3} \left\{ 100[(x_{3i} - 10)^2 + (r-1)^2] + x_{3i}^2 \right\}$

$$\in = \begin{cases} 1/2 \arctan (x_{2i} / x_i) & \text{for } x_i > 0 \\ 1/2 + 1/2 \arctan (x_{2i} / x_i) & \text{for } x < 0_i \end{cases}$$

$r= (x_i^2 + x_{2i}^2)^{1/2}, \quad x_0=(-1, 0, 0; ...)^T.$

17- Generalized Powell 3. (Powell Three Variable Function):

$$f = \sum_{i=1}^{n/3} \left\{ 3-[\frac{1}{1+(x_i-x_{2i})^2} \; 1 -\sin (\frac{\pi x_{2i} x_{3i}}{2}) -\exp[-(\frac{x_i + x_{3i}}{x_{2i}} -2)^2] \right\}, \quad x_0=(0, 1, 2; ...)^T.$$

18- Generalized Cantreal Function:

$f = \sum_{i=1}^{n/4} [(\exp(x_{4i-3})-x_{4i-2})^4 + 100(x_{4i-2}-x_{4i-1})^\sigma + (\arctan(x_{4i-1} -x_{4i})^4 + x_{4i-3}], \quad x_0=(1, 2, 2,2; ...)^T.$