

## A new hybrid scaling parameter for a VM-algorithm

Abbas.Y.Al-Bayati\*

Ghada.M.Al-Naemi\*\*

---

### Abstract

In this paper we have proposed a modified hybrid conjugate direction algorithm which combined a well-known CG-method which was based on a non-quadratic model and a well-known VM-method which based on the quadratic model.

The new modified algorithm was treated both theoretically and numerically and proved to be stable and its convergence was super-linear and it uses an exact line search.

Our numerical results indicate that the modified hybrid method performs well compared to the two well-known used method.

معلمة جديدة للقياس الهجين في خوارزمية المتري المتغير

الملخص

---

\* Prof. /Department of Mathematics/College of computer sciences and Mathematics

\*\*Lecture /Department of Mathematics/College of computer sciences and Mathematics

Received:1/7 /2009 \_\_\_\_\_Accepted: 6 /12 / 2009

## 1-Introduction

We will consider the problem of computing a point  $x = (x_1, x_2, \dots, x_n)^T$  which is a good approximation to a local minimum of a nonlinear twice differentiable function  $f(x)$ . To solve a particular problem of this type, one commonly used either a Conjugate Gradient (CG) algorithm or a Variable Metric (VM) algorithm. Each has its advantages. In general, a CG-algorithm requires more iterations than a VM one to obtain an equally good local minimum, but on the other hand a CG requires little storage for implementation.

We try to solve the unconstrained minimization problem

$$\text{Min}_{x \in R^n} f(x), \quad (1)$$

where  $f$  is twice continuously differentiable function. This problem is usually solved iteratively. Starting with an initial estimate  $x_1$  of the minimum point, each subsequent point  $x_{i+1}$ ,  $i \geq 1$ , will be derived by searching along a descent direction  $d_i$ , such that  $d_i^T g_i > 0$ , where  $g_i = \nabla f(x_i)$  so that  $x_{i+1} = x_i + \lambda_i d_i$  for  $i \geq 1$ , where  $\lambda_i$  is the step-length computed by a line search procedure (Poul and, Kristian, 2004). These line search procedures are traditional and efficient technique for solving unconstrained minimization problems. This convergence has attracted more attention in recent years; consequently, some new line search methods have been proposed, for example, see (Jongen, et al, 2004) & (Nocedal & Wright, 2006).

Or,  $\lambda_i$  may also be computed by Wolfe line search procedure, namely:

$$f(x_i + \lambda_i d_i) \leq f(x_i) + c_1 \lambda_i g_i^T d_i, \quad (2a)$$

and

$$|g_i^T d_i| \leq -c_2 g_i^T d_i, \quad (2b)$$

where  $0 < c_1 < c_2 < 1$ . Conjugate Gradient (CG) method is one of the few practical methods for solving large dimensionality problems because it does not require matrix storage and its iteration cost is very low. Normally the initial direction  $d_1$  is given by  $d_1 = -g_1$ . The search direction

$$d_{i+1} = -g_{i+1} + \beta_i d_i, \quad (3)$$

where  $\beta_i$  is a constant parameter obtained by (Fletcher & Reeves, 1964), or by (Polak-Rib'ere, 1969), or by (Hestenes & Stiefel, 1952), or by (Dixon, 1975).

## 2-The Extended CG-Method

Standard method for solving the problem (1) includes the CG-method which requires  $4n$  locations of computer storage to implement. The CG-method is iterative and it generates a sequences of approximations to the minimizer  $x_{nim}$  of  $f(x)$ .

We can define the classical CG-algorithm as follows (Jongen, et al 2004):

### Algorithm (2.1):

Given  $x_1 \in R^n$ , an initial estimate of the minimizer  $x^*$ ,

**Step (1):** Set  $i=1$ ,  $d_i = -g_i$ .

**Step (2):** Set  $x_{i+1} = x_i + \lambda_i d_i$ , where  $\lambda_i$  minimizes  $f(x_i + \lambda_i d_i)$ .

**Step (3):** Set  $d_{i+1} = -g_{i+1} + \beta_i d_i$ ,  $\beta_i$  is scalar defined in different manners.

**Step (4):** Check for convergence .

**Step (5):** If  $i=n$ , then go to step (1).

**Step (6):** Else, set  $i=i+1$  and go to step (2).

In fact, many attempts have been made to investigate more functions than the quadratic one as a basis for the CG-method. Over years, various authors have published works to solve this problem, for many sorts of objective function, see for example (Fried, 1971),(Goldfard, 1972),(Boland,et al, 1979a, 1979b),(Tassopoulos& Story, 1984a, 1984b), (Al-Bayati, 1993), (Al-Bayati & Al-Naemi, 1995), (Andrei, 2006)

The most popular extended CG-algorithm which based on the logarithmic model

$$f(x) = \varepsilon(\log(q(x)) - 1), \quad \varepsilon < 0, q < 0 \quad (4)$$

Is used as the first part of the new interleaved algorithm, see(Al-Bayati & Al-Naemi, 1995). Now, the outlines of the algorithm are listed below:

**Algorithm (2.2): (An extended\_non-quadratic model algorithm)**

For general function  $f(x)$  with gradient  $g(x)$  and for any starting point  $x_1 \in R^n$ , follow these steps :

**Step (1):** Set  $i=1$ ,  $d_i = -g_i$ .

**Step (2):** Set  $x_{i+1} = x_i + \lambda_i d_i$ ,  $i \geq 1$  where  $\lambda_i$  minimizes  $f(x_i + \lambda_i d_i)$  along  $d_i$ .

**Step (3):** Compute  $\rho_i$  from

$$1 + \frac{y}{2!} + \frac{y^2}{3!} + \frac{y^3}{4!} + \dots + \frac{y^{n-1}}{n!} = \frac{\lambda_i g_i^T g_i / 2}{f_{i+1} - f_i} \quad (5a)$$

$$y_i = \log \rho_i \Rightarrow \rho_i = \exp(y_i) \quad (5b)$$

**Step (4):** Compute  $d_{i+1} = -g_{i+1} + \rho_i \beta_i d_i$ .

**Step (5):** Test for convergence ; if not continue.

**Step (6):** If  $i=n$  or any other restarting criterion is satisfied, go to step (1) else, set  $i=i+1$  and go to step (2).

Now, to ensure that the extended CG-method produces an identical sequence of approximations as a standard CG-algorithm, let us consider the following theorem:

**Theorem (2.1)**

Given an identical starting point  $x_1$ , the method of (Fletcher and Reeves, 1964) defined by

$$d_1 = -g_1, \quad (6)$$

$$d_{i+1} = -g_i + \beta_i d_i, i \geq 1, \quad (7)$$

$$\beta_i = \|g_{i+1}\|^2 / \|g_i\|^2. \quad (8)$$

And  $\| \cdot \|$  is the Euclidean norm applied to  $f(x)=q(x)$  and the extended CG-method using the following search direction:

$$d_1^* = -g_i^*, \quad (9)$$

$$d_{i+1}^* = -g_{i+1}^* + \rho_i \beta_i d_i^*, i \geq 1, \quad (10)$$

$$\rho_i = f_i' / f_{i+1}', \quad (11)$$

$$\beta_i = \|g_{i+1}^*\|^2 / \|g_i^*\|^2. \quad (12)$$

And applied to  $f(q(x))$  generate identical conjugate direction (within a

positive multiple  $f'_i$ ) and the identical sequence of approximations  $x_i$  to the solution  $x^*$  for any function satisfying

$$\frac{df}{dq} = f' > 0 \text{ for } q > 0 \text{ and } \varepsilon > 0. \quad (13)$$

It assumed that the one-dimensional searches are exact. The vectors  $g_1^*, g_i^*$  are gradients of  $f(q(x))$  at  $x_1$  and  $x_i$ , respectively.

### **Proof:**

The theorem is true for  $i=1$ , because

$$d_1^* = -g_1^* = -f'_1 g_1 = f'_1 d_1. \quad (14)$$

Now for  $i=2$ , we have

$$\begin{aligned} d_2^* &= -g_2^* + \rho_1 \beta_1 d_1^* \\ &= -f'_2 g_2 + (f'_1 / f'_2) (\|g_2^*\|^2 / \|g_1^*\|^2) f'_1 d_1 \\ &= -f'_2 g_2 + (f'_1 / f'_2) (f'_2 / f'_1)^2 (\|g_2\|^2 / \|g_1\|^2) f'_1 d_1 \\ &= f'_2 d_2. \end{aligned} \quad (15)$$

Assume that, for  $i \geq 2$ ,

$$\begin{aligned} d_i^* &= f'_i [-g_{i+1} + (\|g_{i+1}\|^2 / \|g_i\|^2) d_i] \\ &= f'_i d_i. \end{aligned} \quad (16)$$

If follows from (10) that

$$\begin{aligned} d_{i+1}^* &= -g_{i+1}^* + \rho_i \beta_i d_i^* \\ &= -f'_{i+1} g_{i+1} + (f'_i / f'_{i+1}) (f'_{i+1} / f'_i)^2 (\|g_{i+1}\|^2 / \|g_i\|^2) f'_i d_i \\ &= -f'_{i+1} d_{i+1}. \end{aligned} \quad (17)$$

### **3-The Self-Scaled VM-Algorithm**

VM algorithm begin an estimate  $x_i$  to the minimizer  $x_{\min}$  and a numerical estimate  $H_1$  of the inverse Hessian matrix  $G^{-1}(x)$ .

A sequence of points

$$x_{i+1} = x_i - \lambda_i H_i g_i \quad (18)$$

where  $H_i$  is updated by a correction of rank-2 matrix of family, i.e. the BFGS update

$$H_{i+1} = [H_i y_i y_i^T H_i / y_i^T H_i y_i + w_i w_i^T] + v_i v_i^T / v_i^T y_i, \quad (19)$$

with

$$\begin{aligned} v_i &= x_{i+1} - x_i, \quad y_i = g_{i+1} - g_i, \\ w_i &= (y_i^T H_i y_i)^{1/2} [v_i / v_i^T y_i - H_i y_i / y_i^T H_i y_i] \end{aligned} \quad (20)$$

The self scaled updating can be defined

$$H_{i+1} = H_i - H_i y_i y_i^T H_i / y_i^T H_i y_i + w_i w_i^T + \rho v_i v_i^T / v_i^T y_i, \quad (21)$$

where

$$\rho = y_i^T H_i y_i / y_i^T v_i \quad (22)$$

(Al-Bayati, 1991). Now, the outlines of the standerd self-scaling Vm-algorithm are:

**Algorithm (3.1): (A self-scaling quadratic model algorithm)**

Start with any initial point  $x_1$ .

**Step (1):** Set  $i=1$  and choose  $H_1$  to be any positive definite matrix (usually  $H_1=I$ ).

**Step (2):** Determine the step size  $\lambda_i$  minimizes  $f(x_i + \lambda_i d_i)$  where  $d_i = -H_i g_i$ , and obtain  $x_{i+1} = x_i + \lambda_i d_i$ .

**Step (3):** Compute the self scaled updating by equation (9) & (10).

**Step (4):** Test for convergence: if not put  $i=i+1$  and go to step (2).

***Theorem (3.1)***

Assume That  $f(x)$  be quadratic function and that line searches are exact: if  $H$  is any symmetric positive definite matrix and we define an updating

$$H_{i+1} = H_i - H_i y_i y_i^T H_i / y_i^T H_i y_i + w_i w_i^T \sigma_i v_i v_i^T / v_i^T y_i, \quad (23)$$

where

$$\sigma_i = y_i^T H_i y_i / y_i^T v_i, \quad (24)$$

Then the search direction

$$d_i^* = -H_i g_i^*, \quad (25)$$

Is identical to the conjugate direction  $d_{CG}^*$  defined by

$$d_{i+1CG}^* = \begin{cases} -g_i & , \text{for } i = 0 \\ -g_i^* + (y_i^T g_i^* / y_i^T d_i) d_i & , \text{for } i \geq 1 \end{cases} \quad (26)$$

***Proof:***

The update (23) can be written as:

$$H_{i+1}^* = H_i - v_i y_i^T H_i / v_i^T y_i - H_i y_i v_i^T / v_i^T y_i + (\sigma_i + y_i^T H_i y_i / v_i^T y_i) v_i v_i^T / v_i^T y_i. \quad (27)$$

Now

$$\begin{aligned} d_{i+1}^* &= -H_i g_i^* + y_i^T H_i g_i^* / v_i^T y_i v_i + v_i^T g_i^* H_i y_i / v_i^T y_i - 2y_i^T H_i y_i v_i^T g_i^* / (y_i^T v_i)^2 v_i \\ &= -H_i g_i^* + y_i^T H_i g_i^* / y_i^T v_i, \end{aligned} \quad (28)$$

Using the property  $v_i^T g^* = 0$ , quoted earlier which holds for exact line searches. The vector  $g^*$  can be substituted for  $Hg^*$  by using the property

### Property (3.2)

Let  $f(x)$  be a quadratic function. Choose an initial approximation  $H_1 = H$ , where  $H$  is any symmetric positive definite matrix appropriate order. Obtain  $H^*$  from  $H$  where  $d = -Hg$  is the search direction and assuming exact line searches then

$$H_{i+1} g^* = H g^*, \text{ for } 0 \leq i < k \leq n. \quad (29)$$

For proof see (Al-Bayati, 1991)

## 4-A Modified CD-Algorithm (based on mixed quadratic & non-quadratic model)

The fundamental strategy which we wish to present the following. If it is based on combining the self-scaling VM- restarts of the form (21) which subsequent ECG-steps will be defined in (5).

The new self-scaled updating can be redefined as:

$$H_{i+1} = [H_i y_i y_i^T H_i / y_i^T H_i y_i + w_i w_i^T] + \rho_i v_i v_i^T / v_i^T y_i \quad (30)$$

where

$$\rho^{CD} = \gamma \rho^{VM} + (1 - \gamma) \rho^{CG} \quad (31)$$

is convex combination of  $\rho^{VM}$  defined in (22) and  $\rho^{CG}$  defined in (5),

where  $0 \leq \gamma \leq 1$ , notice that  $\rho^{VM}$  &  $\rho^{CG}$  have the same effectiveness on the real value  $\rho$  of the parametric  $\rho$  when  $\gamma = 0.5$ .

### Algorithm (4.1): (New algorithm)

Start with any initial point  $x_1$ .

**Step (1):** Set  $i=1$  and choose  $H_1$  to be any positive definite matrix (usually  $H_1 = I$ ).

**Step (2):** Determine the step size  $\lambda_i$  minimizes  $f(x_i + \lambda_i d_i)$  where

$$d_i = -H_i g_i, \text{ and compute } x_{i+1} = x_i + \lambda_i d_i.$$

**Step (3):** Compute the self scaled updating by equation (11) & (12).

**Step (4):** Test for convergence: if not put  $i=i+1$  and go to step (2).

## 5- Some observations on the new algorithm

**1- Positive definite  $H_i$ :** have been proved in (Al-Bayati, 1991) that the self-scaling updating formula  $H_{i+1}$  generates identical search with standard CG-algorithm and it is a positive definite. Hence, it is clear that the new algorithm verifies the positive definite property.

**2- line Search:** In the new algorithm the line search strategy must satisfy the following properties

$$d_{i+1}^T g_i < d_{i+1}^T g_{i+1} \quad (32)$$

This condition is equivalent to that  $H_i$  is a positive definite. For proof see

(Buckley, 1978).

**3-** the new method preserves the super-linear and stability because the scalar  $\rho^{CD}$  depends on the linear combination values of  $\rho^{VM}$  and  $\rho^{CG}$  which have been proved to the algorithms with super-linearly convergence property.

## 6- The Numerical Results

In this section we have compared our new proposed CD-algorithm against the standard well-known BFGS algorithm which was known as the best and effective VM-algorithm. Of course, the scalar  $\rho^{CD}$  depends on a linear combination of  $\rho^{VM}$  and  $\rho^{CG}$ , the scale of the parametric  $\gamma$ ,  $0 < \gamma < 1$  has been noticed, so  $\gamma = 0.5$  is the average of value of the scalar  $\rho^{CD}$  and it performs better than the values of  $\gamma$  is very small tends to zero and  $\gamma$  is very large tends to one. The total Number of Iteration (NOI) and total Number of Function evaluations (NOF).

The comparison test involve sixteen well-known test functions with different dimensions  $2 \leq n \leq 1000$  (see Appendix). From Table (6.1), taking the percentage of 100% for NOI and NOF of the standard BFGS algorithm, we have found that there are a bout overall 15% NOI and 17% NOF



improvements against the standard BFGS algorithm.

Also from Table (6.2) we have found by taking the same consideration that there are about overall 20% NOI and 27% NOF improvements against the standard BFGS algorithm.

The stopping criteria in Table (6.1) has been taken to be

$$\|g_{i+1}\| \leq 1.E-5. \quad (33)$$

while in Table (6.2) the stopping criteria is taken to be

$$|f_{i+1} - f_i| > 5.E-10. \quad (34)$$

**Table (6.1)**

Comparison of the new algorithm against the standard BFGS algorithm using the stopping criteria  $\|g_{i+1}\| < \varepsilon$ .

Test function	N	$\gamma$	New Method NOI [NOF]		Standard BFGS NOI [NOF]	
Rosen	2	0.01	20	[75]	23	[74]
		0.5	21	[58]		
		0.95	22	[61]		
Cubic	2	0.01	7	[22]	7	[25]
		0.5	7	[19]		
		0.95	7	[19]		
Beal	2	0.01	9	[27]	9	[26]
		0.5	8	[20]		
		0.95	8	[20]		
Fred	2	0.01	6	[18]	6	[21]
		0.5	6	[18]		
		0.95	6	[18]		
Bigg	3	0.01	8	[31]	9	[37]
		0.5	11	[36]		
		0.95	11	[33]		
Recip	3	0.01	12	[43]	10	[33]
		0.5	10	[28]		
		0.95	9	[27]		
Powel	4	0.01	23	[80]	21	[87]
		0.5	25	[88]		
		0.95	34	[75]		
Wood	4	0.01	58	[177]	57	[165]
		0.5	61	[196]		
		0.95	20	[46]		
Shallow	4	0.01	8	[24]	8	[25]
		0.5	6	[17]		
		0.95	8	[19]		
Sum	4	0.01	5	[31]	5	[36]
		0.5	4	[22]		
		0.95	4	[22]		
Dixon	4	0.01	13	[39]	15	[39]
		0.5	14	[34]		
		0.95	15	[35]		
Ex-Rosen	6	0.01	43	[132]	44	[139]
		0.5	43	[132]		
		0.95	43	[132]		
Ex-Wolfe	40	0.01	36	[73]	44	[89]
		0.5	37	[75]		
		0.95	38	[77]		
Ex-Full	40	0.01	40	[82]	0	[82]

		0.5	40	[81]	
		0.95	40	[81]	
Ex-Non-Diagonal	300	0.01	52	[126]	63 [149]
		0.5	23	[61]	
		0.95	22	[63]	
Ex-Miele	800	0.01	38	[132]	44 [145]
		0.5	40	[127]	
		0.95	41	[120]	
Ex-Wolfe	800	0.01	47	[96]	85 [172]
		0.5	46	[94]	
		0.95	46	[94]	
Ex-Powell	1000	0.01	70	[167]	39 [111]
		0.5	51	[114]	
		0.95	50	[104]	
Ex-Fred	1000	0.01	8	[22]	10 [27]
		0.5	8	[23]	
		0.95	6	[18]	
Ex-Cubic	1000	0.01	14	[34]	11 [32]
		0.5	8	[21]	
		0.95	7	[19]	
Ex-Beal	1000	0.01	10	[28]	10 [28]
		0.5	9	[23]	
		0.95	8	[20]	40 [82]
Total			478	[1287]	560 [1542]
			85%	[83%]	100% [100%]

**Table (6.2)**

Comparison of the new algorithm against the standard BFGS algorithm using the stopping criteria  $|f_{i+1} - f_i| < \varepsilon$

Test function	N	$\gamma$	New Method NOI [NOF]	Standard BFGS NOI [NOF]
Rosen	2	0.01	20 [75]	22 [72]
		0.5	20 [56]	
		0.95	21 [59]	
Cubic	2	0.01	22 [82]	15 [60]
		0.5	17 [52]	
		0.95	19 [55]	
Beal	2	0.01	8 [24]	8 [60]
		0.5	8 [20]	
		0.95	8 [20]	
Fred	2	0.01	6 [18]	6 [21]
		0.5	6 [18]	
		0.95	6 [18]	
Powell 3	3	0.01	8 [23]	9 [25]
		0.5	9 [25]	
		0.95	11 [30]	
Wood	4	0.01	57 [170]	56 [162]
		0.5	60 [194]	
		0.95	19 [44]	
Powell	4	0.01	21 [73]	18 [76]
		0.5	24 [69]	
		0.95	29 [63]	
Shallow	4	0.01	7 [21]	8 [25]
		0.5	6 [17]	
		0.95	7 [16]	
Sum	4	0.01	6 [37]	6 [43]
		0.5	8 [58]	
		0.95	7 [55]	
Dixon	4	0.01	12 [37]	14 [37]
		0.5	13 [32]	

		0.95	14	[33]	
Ex-Rosen	20	0.01	85	[225]	86 [278]
		0.5	21	[58]	
		0.95	22	[61]	
Ex-Full	40	0.01	38	[77]	38 [89]
		0.5	38	[77]	
		0.95	38	[77]	
Ex-Powel	40	0.01	43	[105]	32 [96]
		0.5	45	[97]	
		0.95	39	[82]	
Ex-Strait	80	0.01	5	[13]	5 [14]
		0.5	5	[14]	
		0.95	5	[13]	
Ex-N0n-Diagonal	800	0.01	55	[131]	56 [134]
		0.5	23	[63]	
		0.95	21	[61]	
Total	Total		303	[850]	379 [1157]
			80%	[73%]	100% [100%]

## Appendix

### 1- Wolfe Function:

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

$$x_0 = (-1; \dots)^T.$$

### 2- Generalized Recipe Function:

$$f(x) = \sum_{i=1}^{n/3} \left[ (x_{3i-1} - 5)^2 + x_{9i-1}^2 + \frac{x_{3i}^2}{(x_{3i-1} - x_{3i-2})^2} \right],$$

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

### 3- Generalized Miele Function:

$$f(x) = \sum_{i=1}^{n/4} \left[ \exp(x_{4i-3} - x_{4i-1})^2 + 100(x_{4i-2} - x_{4i-1})^6 + \left[ \tan(x_{4i-1} - x_{4i}) \right]^4 + x_{4i-3}^8 + (x_{4i} - 1)^2 \right],$$

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

### 4- Generalized Powell Function:

$$f(x) = \sum_{i=1}^{n/4} \left[ (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + \left[ (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4 \right] \right],$$

$$x_0 = (3, -1, 0, 3; \dots)^T.$$

### 5- Generalized Powell 3 Function:

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

$$x_0 = (0, 1, 2; \dots)^T.$$

### 6- Non-Diagonal Variant of Rosenbrok Function:

$$f(x) = \sum_{i=2}^n [100(x_1 - x_i^2)^2 + (1 - x_i)^2]; n > 1,$$

$$x_0 = (-1; \dots)^T.$$

### 7- Dixon Function:

$$f(x) = (1 - x_1)^2 + (1 - x_{10})^2 + \sum_{i=1}^9 (x_i^2 - x_{i+1})^2,$$

$$x_0 = (-1; \dots)^T.$$

### 8- Rosenborck Function:

$$f(x) = \sum_{i=1}^{n/2} 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2$$

$$x_0 = (-1.2, 1; \dots)^T.$$

### 9- Generalized Cubic Function:

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

$$x_0 = (-.2, 1, \dots)^T.$$

### 10- Generalized Beale Function:

$$f(x) = \sum_{i=1}^{n/2} \left\{ [1.5 - x_{2i-1}(1 - x_{2i})]^2 + [2.25 - x_{2i-1}(1 - x_{2i}^2)]^2 \right\} + [2.625 - x_{2i-1}(1 - x_{2i}^2)]^2$$

$$x_0 = (-1, 1, \dots)^T.$$

### 11- Generalized Shallow Function:

$$f(x) = \sum_{i=1}^{n/2} [x_{2i-1}^2 - x_{2i}]^2 + (1 - x_{2i-1})^2,$$

$$x_0 = (-2, 2; \dots)^T.$$

### 12- Generalized Strait Function:

$$f(x) = \sum_{i=1}^{n/2} (x_{2i-1}^2 - x_{2i})^2 + 100(1 - x_{2i-1})^2,$$

$$x_0 = (2, -2; \dots)^T.$$

### 13- Sum of Quadrics (SUM) function:

$$f(x) = \sum_{i=1}^n (x_i - i)^4.$$

[14] \_\_\_\_\_ A new hybrid scaling parameter...

$$x_0 = (1, \dots)^T.$$

14- Full Set of Distinct Eigenvalues Function:

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^n [2x_i - x_{i-1}]^2,$$

$$x_0 = (1, \dots)^T.$$

15- Biggs Function:

$$f(x) = \sum_{i=1}^{10} [\exp(-x_1 z_i) - x_3 \exp(-x_2 z_i) - \exp(-z_i) + 5 \exp(-10z_i)]^2,$$

Where

$$z_i = (0.1)^i, \text{ and}$$

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

16- Freudenstein and Roth Function:

$$f(x) = [-3 + x_1 + ((5 - x_2)x_2 - 2)x_2]^2 + [29 + x_1 + ((1 + x_2)x_2 - 14x_2)]^2,$$

$$x_0 = (30, 3)^T.$$

## 7- References

- 1-Al-Bayati, A.Y. "A new Non-Quadratic Model For Unconstrained Non-Linear Optimization Method", Natural and Applied Series, Mu'tah Journal for Research and Studies, Mu'tah University, Jordan, pp.131-155, 1993.
- 2- Al-Bayati, A. Y. & Al-Naemi "New Extended CG Method for Non-Linear Optimization", Natural and Applied Series, Mu'tah Journal for

Research and Studies, Mu'tah University, Jordan, pp.69-87, 1995.

- 3-Al-Bayati, A.Y. **"A New Family of Self-Scaling Variable Metric Algorithm for unconstrained Optimization"**, J. of Education & Science, Mosul, IRAQ, vol. (12), pp.25-54, 1991.
- 4- Andrei, N. **"Test Function for Unconstrained Optimization"**.  
<http://www.ici.ro/camo/neculai/SCALCG/evalfg.for>
- 5- Boland, W.R.; Kamgnia, E.R. and Kowalik, J.S. **"A Conjugate Gradient Optimization Method Invariant to Non-Linear Scaling"**, Journal of Optimization Theory and Application 27, pp.221-230, 1979a.
- 6-Boland, W.R.; Kowalik, J.S. **" Extended Conjugate Gradient With Restarts"**, Journal of Optimization Theory and Application 28, pp.1-9, 1979b.
- 7- Dixon, L. C. W. **"Conjugate Gradient Algorithm: Quadratic Termination without Linear Search"**, Numerical Optimization Center, The Hatfield Polytechnic, Technical Report No.38, 1975.
- 8- Fletcher, R. and Reeves, C.M. **"Function Minimization by Conjugate Gradient"**, The Computer Journal 7, pp.149-154, 1964.
- 9- Fried, I. **"N-Steps Conjugate Gradient Minimization Scheme for Non-Quadratic Function"**, AIAA Journal 19, pp.2286-2287, 1971.
- 10- Goldfard, D. **"Variable Metric and Conjugate Direction Methods in Unconstrained Optimization "**, Recent Development, ACM, National Meeting, Boston, Massachusetts, pp.496-606, 1972.
- 11- Hestenes, H. R. and Stiefel, E. **"Method of Conjugate Gradients for Solving Linear System"**, J. of Research of National Bureau of Standards, 49, pp.409-436, 1952.
- 12- Jongen, H. TH., Meer, K. and Triesch, E., **"Optimization Theory"**, Springer Science, Business Media, Inc. Kluwer Academic Publishers,

[16] \_\_\_\_\_ A new hybrid scaling parameter...

Boston, 2004.

- 13- Nocedal J., and Wright S. J., "**Numerical Optimization**", Springer series in Operations Research, 2<sup>nd</sup> edition, Springer-Verlag, Inc., New York, USA, (2006).
- 14- Polak, E. "Computational Method in Optimization: A Unified Approach", Academic Press, New Yourk, 1969.
- 15- Poul, E.& Kristian , J. "**Unconstrained Optimization**", Informatics and Mathematical modeling Technical , University of Denmark, Denmark, 2004.
- 16- Tassopoulos, A. and Story, C. "**A Conjugate Direction Method Based on Non-Quadratic Model** ", Journal of Optimization Theory and Application 43, pp.371-381, 1984a.
- 17- Tassopoulos, A. and Story, C. "**Use of a Non-Quadratic Model in a Conjugate Gradient Model of Optimization With Inexact Line Search**", Journal of Optimization Theory and Application 43, pp.357-370, 1984b.