Iraqi Journal of Statistical Science (17) 2010
The Second Scientific Conference of Mathematics-Statistics& Information
pp [137-154]

A New Transformed VM-algorithm for Unconstrained Optimization

Abbas Y. AL-Bayati* Basim A. Hassan**

Abstract

The Broyden class of Quasi-Newton updates is an approximation for the inverse Hessian matrix, are transformed to the standard BFGS update, which makes it possible to generalize the well-known t-BFGS formula. One of the variants, the simpler of them, is given in this study and does not require any additional matrix storage by vector multiplications. Experimental results indicate that the new proposed algorithm was more efficient than the standard BFGS algorithm.

خوارزمية جديدة محولة للمتري المتغير في الأمثلية غير المقيدة

الملخص

إن صنف Broyden الشبيه بتحديثات نيوتن عبارة عن تقريب إلى معكوس مصفوفة هسي والتي هي عبارة عن تحويلات تقريبية إلى صيغة BFGS وتعرف بـ t-BFGS. في هذا البحث تم التطرق إلى تحويل جديد لا يحتاج إلى خزن أية مصفوفة إضافية. النتائج العددية أثبتت كفاءة الخوارزمية المقترحة مقارنة بـ BFGS القياسية.

*Prof./ Department of mathematics/College of Computers Sciences and Mathematics
**Lecture/Department of mathematics/College of Computers Sciences and Mathematics
Received:1/7 /2009 ____________________Accepted: 6 /12 / 2009

1. Introduction :

Consider the unconstrained optimization problem :

$$\min\left\{ f(x) \;\middle|\; x \in R^n \right\} \qquad ....(1)$$

where f is a continuously differentiable function of n variables .Quasi-Newton methods for solving (1) often needed the new search direction d_k at each iteration by :

$$d_k = -H_k g_k \qquad ....(2)$$

where $g_k = \nabla f(x_k)$ is the gradient of f evaluated at the current iterate x_k **[11]**. One then computes the next iterate by

$$x_{k+1} = x_k + \alpha_k d_k \qquad ....(3)$$

where the step size α_k satisfies the Wolfe – conditions

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta_1 \alpha_k d_k^T g_k \qquad ....(4)$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \delta_2 d_k^T g_k \qquad ....(5)$$

where $0 \prec \delta_1 \prec 1/2$ and $\delta_1 \prec \delta_2 \prec 1$, and H_{k+1} is an approximation to $\left\{\nabla^2 f(x_k)\right\}^{-1}$ .The matrix H_{k+1} satisfies the actual quasi-Newton condition

$$H_{k+1} y_k = \rho_k v_k \qquad ....(6)$$

where $y_k = g_{k+1} - g_k$, $v_k = x_{k+1} - x_k$, ρ_k is a scalar, for Exact QN condition $\rho_k = 1$ see **[1]** .

To simplify the notation, we now omit the index k and replace the index $k+1$ by + .

The following lemma (1.1) is given in **[6]**.

Lemma (1.1): The BFGS update can be written in the form

$$H_+ = \gamma V^T H V + \frac{\rho}{b} v v^T \qquad ....(7)$$

where

$$V = I - \frac{1}{b} y v^T \qquad ....(8)$$

and where $y = g_+ - g$, $v = x_+ - x$ and $a = y^T H y$, $b = y^T v$, ρ is scalar.

In **[6]** The scaled Broyden class of updates of H with positive value of parameter η can be written in the form :

$$\frac{1}{\gamma} H_+^{Broyden} = H + \frac{w}{b} v v^T - \frac{\eta}{b}\left(H y v^T + v y^T H\right) + \frac{\eta - 1}{b} H y y^T H \qquad ....(9)$$

where

$$w = \frac{\rho}{\gamma} + \frac{a}{b}\eta \;\; . \qquad ....(10)$$

2.A new transformed BFGS update

The matrices storage free BFGS method is an adaptation of the BFGS method. By Malik et al. **[8]** is almost identical to that of standard BFGS method and the only difference is in the matrix update.

Motivated by the work of Barzilai & Borwein **[3]** and Birgin & Martinez **[4],** we use the scalar parameter

$$\lambda_+ = \frac{v^T v}{v^T y} \qquad \text{..........(11)}$$

This parameter is the inverse of the Rayleigh quotient $v^T \nabla^2 f\, v / v^T v$, which lies between the largest and the smallest eigenvalues of the average Hessian $\nabla^2 f$. Rigorous analysis of methods exclusively based on this approximation may be found in **[1,5,9,10]**.

Considering $H = \lambda_+ I$,($B = H^{-1} = (1/\lambda_+)I$) see [2,1], as an approximation to the Hessian of f at x_k, where $\lambda = v^T v / v^T y$. In **[3]** the updated matrix H_+ is defined by

$$H_+^{t-BFGS} = \gamma V^T(\lambda_+ I)V + \frac{\rho}{b} v v^T \quad \text{where } V = I - \frac{1}{b} y v^T \qquad \text{..........(12)}$$

using $H = \lambda_+ I$ and from (9) we get :

$$\frac{1}{\gamma} H_+ = \lambda_+ I + \frac{w}{b} v v^T - \frac{\eta}{b}\left((\lambda_+ I) y v^T + v y^T (\lambda_+ I)\right) + \frac{\eta - 1}{b} (\lambda_+ I) y y^T (\lambda_+ I) \qquad \text{.......(13)}$$

Although we use the unit values of γ and ρ in almost all cases, we will consider also non-unit values in case of VM methods. First we give the simple variant of this transformations. We denote

$$\mu = \eta + (1-\eta)\frac{\rho}{\gamma}\frac{\bar{b}}{\bar{a}} \qquad \text{..........(14)}$$

Theorem (2.1) :

If $\rho \succ 0$, $\gamma \succ 0$. $w \neq 0$, & $\alpha^* = (\bar{\eta} + \sqrt{\mu}) / \bar{w}$ in eq. (13), can be expressed in the following new formula :

$$\frac{1}{\gamma} H_+ = \frac{\rho}{\gamma}\frac{\eta}{\bar{b}} \bar{v}\bar{v}^T + V(\lambda_+ I)V^T, \quad \bar{v} = v - \alpha(\lambda_+ I) y, V = I - \frac{\sqrt{\mu}}{\bar{b}} \bar{v} y^T \qquad \text{..........(15)}$$

Proof : Setting $v = \bar{v} + \xi(\lambda_+ I) y, \xi \in R$, in (13) we obtain

$$\frac{1}{\gamma} H_+ = \lambda_+ I + \frac{\bar{w}}{\bar{b}} \bar{v}\bar{v}^T - \frac{\xi - \bar{w}\eta}{\bar{b}}((\lambda_+ I) y \bar{v}^T + \bar{v} y^T (\lambda_+ I) + (\frac{\eta - 1}{\bar{a}} + \frac{\xi^2 \bar{w} - 2\xi\eta}{\bar{b}})(\lambda_+ I) y y^T (\lambda_+ I)$$

The last term vanishes for $\xi^2 \bar{w} - 2\xi\eta + (\bar{b}/\bar{a})(\eta - 1) = 0$, i.e. for $\xi = (\eta + \sqrt{\bar{\mu}})/\bar{w} = \alpha^*$, then $\xi \bar{w} - \eta = +\sqrt{\bar{\mu}}$ and

$$\frac{1}{\gamma} H_+ = H + \frac{\bar{w}}{b} \bar{v}\bar{v}^T + \frac{+\sqrt{\bar{\mu}}}{b} ((\lambda I) y \bar{v}^T + \bar{v} y^T (\lambda I)) \qquad ....(16)$$

$$\frac{1}{\gamma} H_+ = V(\lambda_+ I)V^T + \left(\bar{w} - \frac{\bar{a}}{\bar{b}} \mu \right) \frac{\bar{v}\bar{v}^T}{\bar{b}} \qquad ....(17)$$

In view of

$$\bar{w} - \frac{\bar{a}}{\bar{b}} \mu = \frac{\rho}{\gamma} + \frac{\bar{a}}{\bar{b}} \eta - \frac{\bar{a}}{\bar{b}} \eta - \frac{\rho}{\gamma}(1 - \eta) = \frac{\rho}{\gamma} \eta \qquad ....(18)$$

and hence eq. (15). is obtained and the proof is completed.

Note that we prefer the minus sign in α and V, to get $\alpha = 0$ and $\bar{v} = v$ when $\eta = 1$, (t-BFGS).

Algorithm (2.1): [new t-BFGS]

Step 0 : Choose an initial point $x_1 \in R^n$, set $k = 1$.

Step 1 : If the hybrid stopping criterion is satisfied stop :

ITERM=1- if $|x_{k+1} - x_k|$ was less than or equal to $1.0D-16$,

ITERM=2- if $|f_{k+1} - f_k|$ was less than or equal to $1.0D-14$,

ITERM=3- if f_{k+1} is less than or equal to $1.0D-16$,

ITERM= 4- if $\|g_k\|$ is less than or equal to $1.0D-6$,

ITERM= 6 if termination criterion was not satisfied, but solution is probably acceptable,

ITERM=12- if NOF exceeded 1000,

ITERM $\prec$ 0- if the method failed.

Step 2 : Solve $d_+ = -H_+ g_+$ to obtain a search direction d where λ is defined by (11).

Step 3 : Find a step size α_k which satisfy the rule (4) and (5) to generate a new iteration point by $x_+ = x + \alpha d$.

Step 4 : Calculate the new updating formula H_+ by

$$H_+ = \frac{\eta}{\bar{b}} \bar{v}\bar{v}^T + V(\lambda_+ I)V^T, \bar{v} \ \&\ V \text{ are defined in } (15) .$$

Step 5 : Set $k = k+1$ and go to step 1 .

3. The global convergence property of the new algorithm :

We assume that f is strongly convex and Lipschitz continuous on the level set

$$L_0 = \{x \in R^n : f(x) \le f(x_0)\}. \qquad ....(19)$$

That is, there exists constants $\mu \succ 0$ and L such that

$$(\nabla f(x) - \nabla f(y))^T (x-y) \ge \mu \|x-y\|^2 \qquad ....(20)$$

and

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x-y\|, \qquad ....(21)$$

for all x and y from L_0 (see **[12]**).

Shanno **[13,14]** proved that the conjugate gradient methods are exactly the BFGS quasi- Newton method, where at every step the approximation to the inverse Hessian is restarted as the identity matrix. Now we extend this result for the scaled conjugate gradient.

The direction d_+ can be computed as :

$$d_+ = -\left\{ \lambda_+ I - \lambda_+ \frac{y\bar{v}^T + \bar{v}y^T}{y^T\bar{v}} + \left[1 + \lambda_+ \frac{y^T y}{y^T \bar{v}}\right] \frac{\bar{v}\bar{v}^T}{y^T\bar{v}} \right\} g_+ \qquad ....(22)$$

$$= -\lambda_+ I + \lambda_+ \left(\frac{g_+^T \bar{v}^T}{y^T \bar{v}} \right) y - \left[\left(1 + \lambda_+ \frac{y^T y}{y^T \bar{v}}\right) \frac{g_+^T \bar{v}}{y^T \bar{v}} - \lambda_+ \left(\frac{g_+^T y}{y^T \bar{v}}\right) \bar{v} \right] \bar{v} \qquad ....(23)$$

If $g_+^T v = 0$, then (15) is reduced to:

$$d_+ = -\lambda_+ g_+ + \lambda_+ \frac{g_+^T y}{y^T \bar{v}} \bar{v} \qquad ....(24)$$

$\lim_{k \Rightarrow \infty} \alpha^* = 0$ because $\alpha^* = (\eta - \sqrt{\mu})/w$, then (24) is reduced to :

$$d_+ = -\lambda_+ g_+ + \lambda_+ \frac{g_+^T y}{y^T v} v \qquad ....(25)$$

Thus, the effect is simply multiplying the Hestenes and Stiefel **[13]** search direction by a positive scalar.

Theorem (3.1) : Suppose that α in (3) satisfies the Wolfe condition (4) and (5), then the direction d_+ given by (23) is a descent direction.
Proof. Multiplying (23) by g_+^T, we have

$$g_+^T d_+ = \frac{1}{(y^T \bar{v})^2}\left[-\lambda_+ \|g_+\|^2 (y^T \bar{v})^2 + 2\lambda_+ (g_+^T y)(g_+^T y)(y^T \bar{v}) - (g_+^T \bar{v})^2 (y^T \bar{v}) - \lambda_+ (y^T y)(g_+^T \bar{v})^2 \right]$$

Applying the inequality $u^T v \le \frac{1}{2}\left(\|u\|^2 + \|v\|^2\right)$ to the second term of the right hand side of the above equality, with $u = (\bar{v}^T y) g_+$ and $v = (g_+^T \bar{v}) y$ we get :

$$g_+^T d_+ \le -\frac{(g_+^T \bar{v})^2}{y^T \bar{v}}. \qquad ....(26)$$

On the other hand, by the strong convexity property (20), we have

$$y^T \bar{v} \ge \mu \left\|\bar{v}\right\|^2 . \qquad ....(27)$$

Now, by Wolfe condition (5) and (27). We have, $g_+^T d_+ \prec 0$ for every $k = 0, 1, 2, ...$, and hence the theorem is proved.

4.Numerical Results :

In this section, we have reported the numerical results for the new formula (15). We tested, using the collection of problems, for general sparse and separable unconstrained optimization test problems from **[7]**. We have in this paper used the dimension of the problem (N), N=10,100,500,1000. Algorithms use a line search technique **[6]** which satisfy Wolfe - condition as in which $\delta_1 = 0.0001$, $\delta_2 = 0.2$. We will test the following two VM-algorithms.

1. Standard BFGS algorithm.
2. New t-BFGS algorithm.

Tables of numerical results show the computational results , where the columns have the following meanings :

Problem : the name of the test problem .

NOI : number of iterations .

NOF : number of function evaluations .

f : value of the objective function at the point x.

g : gradient of the objective function at the point x.

ITERM : the hybrid stopping criterion

From Table **(4-1)** ,we have observed that the average performances of the new formula (15) are better than the standard BFGS and for our selected unconstrained minimization are test problems.

Table (4-1)

Comparison results of all the two algorithms as a total of (15) test functions.

	Standard BFGS algorithm		
N	**NOI**	**NOF**	**TIME**
10	3885	7703	0:00:00.08
100	5463	10332	0:00:00.67
500	3778	8164	0:00:02.20
1000	3747	8001	0:00:04.63
Total	**16873**	**34200**	**0:00:07.58**
	New t-BFGS algorithm with $\eta = 0.5$		
N	**NOI**	**NOF**	**TIME**
10	3005	7253	0:00:00.06
100	4161	9594	0:00:00.53
500	2987	7824	0:00:01.92
1000	2814	7647	0:00:03.94
Total	**12967**	**32318**	**0:00:06.45**
	New t-BFGS algorithm with $\eta = 0.8$		
N	**NOI**	**NOF**	**TIME**
10	3041	7296	0:00:00.06
100	4150	9553	0:00:00.53
500	2997	7830	0:00:01.92
1000	2844	7634	0:00:03.94
Total	**13032**	**32313**	**0:00:06.45**

The details of these test results are fully described in the subsequent tables.

Table (4-2)

Algorithm	standard BFGS algorithm with N=10				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	398	1001	22.3407562	0.163E+01	12
2	654	1001	0.121328883E-03	0.125E-01	12
3	656	1000	0.921077850	0.620E-02	12
4	106	153	0.799804276E-10	0.702E-05	2
5	139	179	0.160886197E-09	0.788E-05	2
6	69	112	3.01929454	0.965E-06	4
7	423	1000	11.4354732	0.217E+01	12
8	57	92	-133.510600	0.233E-05	2
9	374	1002	1.05358706	0.316E-01	12
10	13	23	0.944550269E-13	0.428E-06	4
11	500	1001	0.105143334E-03	0.167E-02	12
12	262	698	1.92460901	0.698E-04	2
13	74	123	-8.05139211	0.877E-06	4
14	74	145	-0.385263183E-01	0.836E-06	4
15	86	173	-0.251419625E-01	0.968E-06	4
Total	**3885**	**7703**			

TIME= 0:00:00.08

Table (4-3)

Algorithm	Standard BFGS algorithm with N=100				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	357	1000	375.772751	0.416E+01	12
2	675	1000	0.417037296E-03	0.229E-01	12
3	642	1000	25.2065330	0.110E-01	12
4	94	126	0.736274182E-10	0.347E-05	2
5	195	259	0.353606890E-08	0.233E-04	2
6	84	145	33.3754297	0.800E-06	4
7	490	725	1039.41617	0.274E-04	2
8	56	57	-98.8560279	0.562E-06	4
9	350	1001	18.9030005	0.500E+00	12
10	7	14	0.512511155E-13	0.667E-07	4
11	500	1001	0.110666145E-05	0.118E-04	12
12	334	1002	2.39784602	0.109E+00	12
13	578	1000	-49.9997833	0.285E-04	12
14	501	1001	-0.133766122E-02	0.207E-01	12
15	600	1001	0.908528629E-02	0.334E-01	12
Total	**5463**	**10332**			

TIME= 0:00:00.67

Table (4-4)

Algorithm	**Standard BFGS algorithm with N=500**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	313	1000	1950.53768	0.506E+01	12
2	654	1000	0.190687684E-03	0.178E-01	12
3	615	1000	133.781367	0.107E-01	12
4	143	179	0.407524183E-10	0.174E-05	2
5	224	298	0.116517131E-08	0.108E-04	2
6	88	151	168.291764	0.134E-05	2
7	83	196	163899.853	0.740E-03	2
8	34	54	90.9672145	0.713E-06	4
9	406	1002	97.5181572	0.102E+01	12
10	6	12	0.315195199E-13	0.251E-06	4
11	132	265	0.101551041E-07	0.100E-05	4
12	334	1002	2.66906251	0.436E-01	12
13	390	1001	-218.394928	0.162E+01	12
14	335	1002	0.311487476	0.209E-01	12
15	334	1002	0.120585472	0.261E-01	12
Total	**4091**	**9164**			

TIME= 0:00:02.18

Table (4-5)

Algorithm	**Standard BFGS algorithm with N=1000**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	318	1002	3920.15325	0.250E+01	12
2	645	1000	0.134675134E-03	0.171E-01	12
3	592	1001	269.500609	0.302E-01	12
4	151	190	0.658806352E-10	0.233E-05	2
5	229	300	0.142991691E-08	0.119E-04	2
6	90	151	336.937181	0.225E-05	2
7	106	265	761774.954	0.242E-02	2
8	34	55	316.436141	0.433E-06	4
9	468	1002	196.256249	0.228E+01	12
10	5	10	0.783883858E-11	0.252E-06	4
11	10	21	0.129032045E-08	0.992E-06	4
12	334	1002	2.68842398	0.300E-01	12
13	414	1000	-423.279033	0.514E+01	12
14	335	1002	0.336253881	0.155E-01	12
15	334	1002	0.128686577	0.194E-01	12
Total	**4065**	**9003**			

TIME= 0:00:04.63

Table (4-6)

Algorithm	**New t-BFGS algorithm with $\eta = 0.5$ and N=10**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	254	1002	18.8964532	0.776E+00	12
2	500	1000	0.737373717E-04	0.151E-01	12
3	444	1001	0.920973871	0.217E-02	12
4	36	67	0.291424252E-11	0.984E-06	4
5	64	103	0.272817745E-11	0.976E-06	4
6	32	66	3.01929454	0.910E-06	4
7	334	1001	11.1394527	0.120E+01	12
8	33	71	-133.510600	0.102E-05	2
9	340	1002	1.05910329	0.245E+00	12
10	11	22	0.125965391E-12	0.416E-06	4
11	500	1001	0.105143334E-03	0.167E-02	12
12	283	567	1.92460901	0.839E-06	4
13	31	63	-8.05139211	0.274E-06	4
14	57	114	-0.385263183E-01	0.799E-06	4
15	86	173	-0.251419625E-01	0.968E-06	4
Total	**3005**	**7253**			

TIME= 0:00:00.06

Table (4-7)

Algorithm	**New t-BFGS algorithm with $\eta = 0.5$ and N=100**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	229	1001	374.048162	0.217E+01	12
2	500	1000	0.903159788E-04	0.175E-01	12
3	479	1001	25.2062519	0.464E-02	12
4	41	77	0.319582393E-11	0.362E-06	4
5	100	186	0.156921851E-10	0.983E-06	4
6	56	119	33.3754297	0.935E-06	4
7	154	382	1057.85018	0.257E-04	2
8	55	57	-98.8560279	0.848E-06	4
9	337	1002	16.3828346	0.257E+01	12
10	7	14	0.348235823E-12	0.173E-06	4
11	500	1001	0.110666145E-05	0.118E-04	12
12	334	1002	2.39784602	0.109E+00	12
13	369	751	-49.9997833	0.499E-05	2
14	500	1000	-0.217226015E-02	0.204E-01	12
15	500	1001	0.142511903E-01	0.206E+00	12
Total	**4161**	**9594**			

TIME= 0:00:00.53

Table (4-8)

Algorithm	**New t-BFGS algorithm with $\eta = 0.5$ and N=500**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	2	21	320193.528	0.197E+06	-6
2	500	1000	0.170900860E-03	0.220E-01	12
3	420	1000	133.781141	0.565E-02	12
4	39	75	0.201232571E-10	0.330E-06	4
5	87	171	0.307767517E-10	0.823E-06	4
6	55	119	168.291764	0.125E-05	2
7	44	134	163899.853	0.613E-03	2
8	20	41	90.9672145	0.780E-06	4
9	330	1000	77.7916010	0.325E+01	12
10	6	12	0.142795749E-12	0.534E-06	4
11	132	265	0.101551041E-07	0.100E-05	4
12	334	1002	2.66906251	0.436E-01	12
13	352	1002	-218.905261	0.149E+00	12
14	334	1001	0.310280282	0.209E-01	12
15	334	1002	0.120585472	0.261E-01	12
Total	**2989**	**7845**			

TIME= 0:00:01.92

Table (4-9)

Algorithm	**New t-BFGS algorithm with $\eta = 0.5$ and N=1000**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	2	21	450850.163	0.256E+06	-6
2	500	1000	0.182658186E-03	0.269E-01	12
3	411	1000	269.499613	0.311E-02	12
4	40	80	0.616107780E-11	0.295E-06	4
5	99	186	0.181714799E-10	0.987E-06	4
6	56	121	336.937181	0.947E-06	4
7	53	170	761774.954	0.252E-02	2
8	27	53	316.436141	0.251E-05	2
9	286	1000	148.400432	0.163E+02	12
10	5	10	0.242975595E-10	0.443E-06	4
11	10	21	0.129032045E-08	0.992E-06	4
12	334	1002	2.68842398	0.300E-01	12
13	325	1001	-427.103778	0.147E+01	12
14	334	1001	0.335617639	0.154E-01	12
15	334	1002	0.128686577	0.194E-01	12
Total	**2816**	**7668**			

TIME= 0:00:03.94

Table (4-10)

Algorithm	**New t-BFGS algorithm with $\eta = 0.8$ and N=10**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	254	1002	18.8964532	0.776E+00	12
2	500	1000	0.737373717E-04	0.151E-01	12
3	460	1000	0.920954442	0.133E-02	12
4	38	70	0.151059113E-11	0.926E-06	4
5	60	102	0.320279426E-11	0.829E-06	4
6	32	66	3.01929454	0.782E-06	4
7	334	1001	11.1394527	0.120E+01	12
8	37	78	-133.510600	0.128E-05	2
9	340	1002	1.05910329	0.245E+00	12
10	10	20	0.293070019E-12	0.752E-06	4
11	500	1001	0.105143334E-03	0.167E-02	12
12	283	567	1.92460901	0.839E-06	4
13	38	76	-8.05139211	0.937E-06	4
14	69	138	-0.385263183E-01	0.871E-06	4
15	86	173	-0.251419625E-01	0.968E-06	4
Total	**3041**	**7296**			

TIME= 0:00:00.06

Table (4-11)

Algorithm	**New t-BFGS algorithm with $\eta = 0.8$ and N=100**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	229	1001	374.048162	0.217E+01	12
2	500	1000	0.903159788E-04	0.175E-01	12
3	480	1001	25.2062513	0.462E-02	12
4	43	73	0.484891937E-11	0.730E-06	4
5	91	178	0.308638605E-10	0.755E-06	4
6	48	102	33.3754297	0.843E-06	4
7	154	382	1057.85018	0.257E-04	2
8	52	54	-98.8560279	0.598E-06	4
9	337	1002	16.3828346	0.257E+01	12
10	7	14	0.920423803E-13	0.892E-07	4
11	500	1001	0.110666145E-05	0.118E-04	12
12	334	1002	2.39784602	0.109E+00	12
13	360	743	-49.9997833	0.783E-05	2
14	500	1000	-0.217226015E-02	0.204E-01	12
15	515	1000	0.100577269E-01	0.103E+00	12
Total	**4150**	**9553**			

TIME= 0:00:00.53

Table (4-12)

Algorithm	**New t-BFGS algorithm with $\eta = 0.8$ and N=500**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	2	21	320193.528	0.197E+06	-6
2	500	1000	0.170900860E-03	0.220E-01	12
3	421	1000	133.781155	0.599E-02	12
4	42	75	0.491093324E-10	0.900E-06	4
5	93	177	0.238605480E-10	0.803E-06	4
6	54	117	168.291764	0.241E-05	2
7	44	134	163899.853	0.613E-03	2
8	21	43	90.9672145	0.752E-06	4
9	330	1000	77.7916010	0.325E+01	12
10	6	12	0.499546372E-13	0.316E-06	4
11	132	265	0.101551041E-07	0.100E-05	4
12	334	1002	2.66906251	0.436E-01	12
13	352	1002	-218.905261	0.149E+00	12
14	334	1001	0.310280282	0.209E-01	12
15	334	1002	0.120585472	0.261E-01	12
Total	**2999**	**7851**			

TIME= 0:00:01.92

Table (4-13)

Algorithm	**New t-BFGS algorithm with $\eta = 0.8$ and N=1000**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	2	21	450850.163	0.256E+06	-6
2	500	1000	0.182658186E-03	0.269E-01	12
3	411	1000	269.499613	0.311E-02	12
4	47	83	0.471625462E-10	0.886E-06	4
5	90	173	0.243391123E-10	0.809E-06	4
6	57	123	336.937181	0.947E-06	4
7	53	170	761774.954	0.252E-02	2
8	25	48	316.436141	0.290E-05	2
9	286	1000	148.400432	0.163E+02	12
10	5	10	0.110653719E-10	0.299E-06	4
11	10	21	0.129032045E-08	0.992E-06	4
12	334	1002	2.68842398	0.300E-01	12
13	358	1001	-427.404476	0.118E-02	12
14	334	1001	0.335617639	0.154E-01	12
15	334	1002	0.128686577	0.194E-01	12
Total	**2846**	**7655**			

TIME= 0:00:03.94

5.Conclusions and Discussions :

In this paper, we have proposed a new Transformation updating formula to the standard BFGS update, call it t-BFGS update for solving unconstrained minimization problems. The computational test shows that the Transformated BFGS approach, given in this paper, is successful. We claim that the new formulae (15) is better than the standard BFGS formula. Namely, for the formula (15) with $\eta = 0.5$ and $\eta = 0.8$ there are about $(23-24)\%$ improvements in NOI , $(6)\%$ improvements in NOF and there are about $(15)\%$ improvements in time overall the calculations and for all different dimensions **(**$10 \le N \le 1000$**).**

Table (5-1)

Relative efficiency of standard BFGS, New t-BFGS.

Methods	NOI	NOF	TIME
standard BFGS	100	100	100
New t-BFGS with $\eta = 0.5$	76.85	94.49	85.09
New t-BFGS with $\eta = 0.8$	77.23	94.48	85.09

6. Test problems for general sparse & partially separable unconstrained optimization

We seek a minimum of either a general objective function $f(x)$ or a partially separable objective function

$$f(x) = \sum_{k=1}^{n_A} f_k(x) \quad , \quad x \in R^n$$

From the starting point $\bar{x}$. For positive integers k & l, we use the notation $div\,(k,l)$ for integer division, i.e., maximum integer not greater than k/l, and $\mathrm{mod}\,(k,l)$ for the remainder after integer division, i.e., $\mathrm{mod}(k,l) = l(k/l - div(k,l))$. The description of individual problems follows.

$problem\ 1$: $Chained\ Wood\ \ function$.

$$f(x) = \sum_{j=1}^{k}[100(x_{i-1}^2 - x_i)^2 + (x_{i-1}-1)^2 + 90(x_{i+1}^2 - x_{i+2})^2$$
$$+ (x_{i+1}-1)^2 + 10(x_i + x_{i+2} - 2)^2 + (x_i - x_{i+2})^2/10]$$

$$i = 2j \quad , \quad k = (n-2)/2$$

$$\bar{x}_i = -3 \; , \; \mathrm{mod}(i,2) = 1 \; , \; i \le 4 \; , \qquad \bar{x}_i = -2 \; , \; \mathrm{mod}(i,2) = 1 \; , \; i \succ 4$$

$$\bar{x}_i = -1 \; , \; \mathrm{mod}(i,2) = 0 \; , \; i \le 4 \; , \qquad \bar{x}_i = 0 \; , \; \mathrm{mod}(i,2) = 0 \; , \; i \succ 4$$

problem 2 :*Chained Powel singular function*.

$$f(x) = \sum_{j=1}^{k} [(x_{i-1} + x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+2})^4]$$

$$i = 2j \quad , \quad k = (n-2)/2$$

$$\bar{x}_i = 3 \; , \; \mathrm{mod}(i,4) = 1 \; , \qquad \bar{x}_i = -1 \; , \; \mathrm{mod}(i,4) = 2$$

$$\bar{x}_i = 0 \; , \; \mathrm{mod}(i,2) = 3 \; , \qquad \bar{x}_i = 1 \; , \; \mathrm{mod}(i,4) = 0$$

problem 3 :*Chained Cragg and Levy function*.

$$f(x) = \sum_{j=1}^{k} [(e^{x_{i-1}} - x_i)^4 + 100(x_i - x_{i+1})^6 + \tan^4(x_{i+1} - x_{i+2}) + x_{i-1}^8 + (x_{i+2} - 1)^2]$$

$$i = 2j \quad , \quad k = (n-2)/2$$

$$\bar{x}_i = 1 \; , \; i = 1 \; , \; \bar{x}_i = 2 \; , \; i \succ 1$$

problem 4 :*Chained Cragg and Levy function*.

$$f(x) = \sum_{i=1}^{n} \left\| (3 - 2x_i)x_i - x_{i-1} - x_{i+1} + 1 \right\|^p$$

$$p = 7/3 \quad , \quad x_0 = x_{n+1} = 0$$

$$\bar{x}_i = -1 \; , \; i \ge 1$$

problem 5 :*generalized Broyden banded function*.

$$f(x) = \sum_{i=1}^{n} \left\| (2 + 5x_i^2)x_i + 1 + \sum_{j \in J_j} x_j(1 + x_j) \right\|^p$$

$$p = 7/3 \quad , \quad J_j = \{ j : \max(1, i-5) \le \min(n, i+1) \}$$

$$\bar{x}_i = -1 \; , \; i \ge 1$$

problem 6 :*Seven – diagonal generaliization of the Broyden tridiagonal function*.

$$f(x) = \sum_{i=1}^{n} \left\| (3 - 2x_i)x_i - x_{i-1} - x_{i+1} + 1 \right\|^p + \sum_{i=1}^{n/2} \left\| x_i + x_{i+n/2} \right\|$$

$$p = 7/3 \quad , \quad x_0 = x_{n+1} = 0$$

$$\bar{x}_i = -1 \; , \; i \ge 1$$

problem 7 : Sparse modification of the Nazareth trigonometric function.

$$f(x)=\frac{1}{n}\sum_{i=1}^{n}\left(n+i-\sum_{j\in J_I}(a_{ij}\sin x_j+b_{ij}\cos x_j)\right)^2$$

$$a_{ij}=5[1+\text{mod}(i,5)+\text{mod}(j,5)]\ ,\ b_{ij}=(i+j)/10$$

$$J_i=\{j:\max(1,i-2)\le\min(n,i+2)\}\cup\{j:\|j-i\|=n/2\}$$

$$\bar{x}_i=1/n\quad,\quad i\ge 1$$

problem 8 : Another trigonometric function.

$$f(x)=\frac{1}{n}\sum_{i=1}^{n}\left(i(1-\cos x_i)-\sum_{j\in J_I}(a_{ij}\sin x_j+b_{ij}\cos x_j)\right)$$

$$a_{ij}=5[1+\text{mod}(i,5)+\text{mod}(j,5)]\ ,\ b_{ij}=(i+j)/10$$

$$J_i=\{j:\max(1,i-2)\le\min(n,i+2)\}\cup\{j:\|j-i\|=n/2\}$$

$$\bar{x}_i=1/n\quad,\quad i\ge 1$$

problem 9 : Chained Wood function.

$$f(x)=\sum_{j\in J}\left\{\exp P\left(\prod_{j=1}^{5}x_{i+1-j}\right)+10\left(\sum_{j=1}^{5}x_{i+1-j}^2-10-\lambda_1\right)^2\right.$$

$$\left.+10\left(x_{i-3}x_{i-2}-5x_{i-1}x_i-\lambda_2\right)^2+10\left(x_{i-4}^3+x_{i-3}^3+1-\lambda_3\right)^2\right\}$$

$$\lambda_1=-0.002008\quad,\quad\lambda_2=-0.001900\quad,\quad\lambda_3=-0.000261$$

$$j=\{i,\ \text{mod}(i,5)=0\}$$

$$\bar{x}_i=-2\ ,\ \text{mod}(i,5)=1\ ,\ i\le 2\ ,\qquad \bar{x}_i=-1\ ,\ \text{mod}(i,5)=1\ ,\ i\succ 2$$

$$\bar{x}_i=2\ ,\ \text{mod}(i,5)=2\ ,\ i\le 2\ ,\qquad \bar{x}_i=-1\ ,\ \text{mod}(i,5)=2\ ,\ i\succ 2$$

$$\bar{x}_i=2\ ,\ \text{mod}(i,5)=3\qquad ,\qquad \bar{x}_i=-1\ ,\ \text{mod}(i,5)=4$$

$$\bar{x}_i=-1\ ,\ \text{mod}(i,5)=0$$

problem 10 : Generalization of the Brown 2 function.

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

$$\bar{x}_i=-1.0\ ,\ \text{mod}(i,2)=1\ ,\ \bar{x}_i=1.0\ ,\ \text{mod}(i,2)=0$$

problem 11 : Discrete boundary value problem.

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

$$h=1/(n+1)\quad,\quad x_0=x_{n+1}=0$$

$$\bar{x}_i=ih\,(1-ih)\quad,\quad i\ge 1$$

problem 12 : Discretization of a variational problem.

$$f(x) = 2\sum_{i=1}^{n}\left[x_i(x_i - x_{i+1})/h + 2h\sum_{i=0}^{n}\left[(e^{x_{i+1}} - e^{x_i})/(x_{i+1} - x_i)\right]\right]$$

$$h = 1/(n+1) \quad , \quad x_0 = x_{n+1} = 0$$

$$\bar{x}_i = ih\,(1 - ih) \quad , \quad i \geq 1$$

problem 13 : Banded trigonometric problem.

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

$$x_0 = x_{n+1} = 0$$

$$\bar{x}_i = 1 / n \ , \quad i \geq 1$$

problem 14 : Variational problem 1.

This problem is a finite difference analogue of a variational problem defined as a minimization of the functional :

$$f(x) = \int_0^1 \left[\frac{1}{2}x^2(t) + e^{x(t)} - 1\right] dt$$

where $x(0) = 0$ & $x(1) = 0$ *.We use the trapeziodal rule together with 3 – point finte differences on a uniforrm grid having* $n+1$ *internal nodes. The starting point is given by the formula*

$\bar{x}_i = x(t_i) = ih\,(1 - ih)$ *, where* $h = 1/(n+1)$.

problem 15 : Variational problem 2.

This problem is a finite difference analogue of a variational problem defined as a minimization of the functional :

$$f(x) = \int_0^1 \left[x^2(t) - x^2(t) - 2t\ x(t)\right] dt$$

where $x(0) = 0$ & $x(1) = 0$ *.We use the trapeziodal rule together with 3 – point finte differences on a uniforrm grid having* $n+1$ *internal nodes. The starting point is given by the formula*

$\bar{x}_i = x(t_i) = ih\,(1 - ih)$ *, where* $h = 1/(n+1)$.

References

[1] Al-Bayati, A.Y. & Abdullah M. R.,(2008):A new family of spectral CG-algorithm, Raf. J. of comp. & Math., Vol. 5, No. 1, pp.69-80.

[2] Apostolopoulou M.S., Sotiropoulos D.G.and Pintelas P,(1999) :A 2-BFGS Updating in a trust region framework .Dep. of math. GR-265 04 Patras, Greece. No. TR07-01PP,pp.1-28.

[3] Barzilai J. and Borwein,(1988):Two point step size gradient method IMA J. Numer. Anal.,Vol. 8, pp.141-148.

[4] Birgin E. G. and Martines J. M., (2001): Apectral conjugate Gradient method for unconstrained optimization. Appl. Math. Optim., Vol. 43, pp.117-128.

[5] Birgin E. G. ,Martines J. M. and Raydan M.,(2000):Non-monotone spectral projected gradient methods on convex sets. SIAM J. Optim., Vol.6,pp.1196-1211.

[6] Luksan L. and Spedicato E. (2000): VM methods for unconstrained Optimization and nonlinear least squares , J. of Computational and Applied Mathe. ,pp. 61-95.

[7] Luksan L.and Vlcek J. (1999):Sparse and separable test problems for unconstrained and equality constrained optimization, Report V-767 Prague, ICS AS CR .

[8] Malik H., Mansor M. & Leong W., (2000): Matrices storage free BFGS method in large scale nonlinear unconstrained optimization, proceeding of International Conference on Math. &its Application in the New Millenninm UPM. pp.430-436.

[9] Raydan M., (1997):The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. Vol.3 ,pp.16-33.

[10] Raydan M., (1993):On the Barzilai and Borwein choice of steplength for the gradient method .SIAM J. Optim. Vol.1 ,pp.321-326.

[11] Storey C. and HU .Y. F. (1993): Preconditioned Low-Order Newton methods , J. of optimization theory and application ,Vol.79, No. 2, pp.311-331.

[12] Hager W.W. and Zhang H. (2005), A new conjugate gradient method with guaranted descent and an efficient line search, SIAM Journal on Optimization, Vol. 16, pp. 170-192.

[13] Shanno D. F. (1978), Conjugate gradient methods with inexact searches, Mathematics of Operations Research, Vol. 3, pp. 244--256.

[14] Shanno D. F. (1978), On the convergence of a new Conjugate gradient algorithm, SIAM J. Nnmer. Anal. Vol. 15, pp. 1247-1257.