Journal of Mathematics and Statistics 4 (2): 81-87, 2008
ISSN 1549-3644

A New Three-Term Preconditioned Gradient memory Algorithm for Nonlinear Optimization Problems

[1]Abbas Y. AL-Bayati and [2]Ivan Subhi Latif
[1]Department of Mathematics, College of Computers Sciences and Mathematics, University of Mosul, Mosul-Iraq
[2]Department of Mathematics, College of Scientific Education, University of Salahaddin, Erbil, Iraq

Abstract: In the present study, we proposed a three-term of preconditioned gradient memory algorithms to solve a nonlinear optimization problem. The new algorithm subsumes some other families of nonlinear preconditioned gradient memory algorithms as its subfamilies with Powell's Restart Criterion and inexact Armijo line searches. Numerical experiments on twenty one well-known test functions with various dimensions generally encouraged and showed that the new algorithm was more stable and efficient in comparison with the standard three-term CG- algorithm.

Key words: Unconstrained optimization, preconditioned conjugate gradient, self-scaling vm- updates, inexact line searches

INTRODUCTION

We consider the unconstrained optimization problem

$$\min_{x\in\Re} f(x) \qquad x \in \Re^n \qquad (1)$$

where $f:\Re^n \to \Re^1$ is a continuously differentiable function in $\Re^n$ and $\Re^n$ is the n-dimensional euclidean space .Conjugate gradient methods were very useful for solving (1).They were of the form

$$x_{k+1} = x_k + \lambda_k d_k \qquad (2)$$

and

$$d_k = \begin{cases} -g_k & for \quad k=1 \\ -g_k + \beta_k d_{k-1} & for \quad k \geq 2 \end{cases} \qquad (3)$$

where g_k denoted by $\nabla f(x_k)$, λ_k is a step-length obtained by a line search, and β_k is a scalar . The memory gradient algorithm for problem (1) was first presented in Cragg and Levy [4] with the ordinary gradient method. This method has the advantage of high speed convergence since it produces a sequence of quadratic convergent points.

A new three-term memory gradient method for problem (1) whose search directions are defined by

$$d_k = -g_k + \beta_k d_{k-1} + \alpha_k d_{k-2} \qquad (4)$$

and

$$x_{k+1} = x_k + \lambda_k d_k \qquad (5)$$

where β_k and α_k are parameters and λ_k is a step-size considered in Sun [8] and obtained by means of a one dimension search. The self-scaling Variable Metric (VM) algorithms were introduced, showing significant improvement in efficiency over earlier methods. The search direction

$$d_k = -H_k g_k \qquad (6)$$

H_k is an approximation to the inverse Hessian G^{-1}. For a given H_1, the matrix H_k was updated to H_{k+1} by a formula from the class of self-scaling updates satisfying the following QN-like condition given in Cohen [3]

$$H_{k+1} y_k = \rho_k V_k \qquad (7)$$

where

$$V_k = x_{k+1} - x_k \qquad (8)$$

There were infinite numbers of possible updates which satisfy the QN-condition. The class of these updates were written as (See Gill and Murray[5])

$$H_{k+1} = \left(H_k - \frac{H_k Y_k Y_k^{\mathrm{T}} H_k^{\mathrm{T}}}{Y_k^{\mathrm{T}} H_k Y_k} + \theta_k W_k W_k^{\mathrm{T}} \right) \sigma_k + V_k V_k^{\mathrm{T}} / V_k^{\mathrm{T}} Y_k \qquad (9)$$

Corresponding Author: Abbas Y. AL-Bayati, Department of Mathematics, College of Computers Sciences and Mathematics, University of Mosul, Mosul-Iraq

θ_k, σ_k parameters and

$$W_k = \left(Y_k^{\mathrm{T}} H_k Y_k\right)^{1/2}\left[V_k / V_k^{\mathrm{T}} Y_k - H_k Y_k / Y_k^{\mathrm{T}} H_k Y_k\right] \quad (10)$$

The parameter θ_k was chosen such that $\theta_k \in [0,1]$ different choice of θ_k defines different updates and different search conjugate directions (See Al-Assady and Al-Bayati[1]), the Davidon-Fletcher-Powell (DFP) update was defined as (9) where $\sigma_k = 1$ with $\theta_k = 0$ while Broyden-Fletcher-Goldfarb-Shanno (BFGS) was defined as an updated corresponds to $\theta_k = 1$. Oren found that a proper scaling of the objective function improve the performance of algorithms that use Broyden family of update where $\sigma_k = V_k^{\mathrm{T}} Y_k / Y_k^{\mathrm{T}} H_k Y_k$. This choice for the scalar parameter σ_k was made primarily because in this case σ_k requires the quotient of two quantities which were already computed in the updating formula. For more details see Yabe and Takano[9].

In this study, we considered a new three-term PCG algorithm for problem (1) whose search directions were defined by

$$d_k = \begin{cases} -H_k g_k & for \quad k = 1 \\ -g_{k-2} + \beta_k H_{k-1} d_{k-1} + \alpha_k H_{k-2} d_{k-2} & for \quad k \geq 2 \end{cases} \quad (11)$$

by using a new line-search parameter and a positive definite matrix H_k.

The Three-Term Memory Gradient Algorithm (TMG):

Consider the three term memory gradient method (4) and (5). Conditions are given on β_k and α_k to ensure that d_k is a sufficient descent direction at the point x_k. Now let $S_k = -g_k + \beta_k d_{k-1}$ and assume that

$$\begin{cases} g_k^{\mathrm{T}} g_k > \left|\beta_k g_k^{\mathrm{T}} d_{k-1}\right| \\ \left|g_k^{\mathrm{T}} S_k\right| \geq (1+\Delta_1)\left|\beta_k\right|\left\|g_k\right\|.\left\|d_{k-1}\right\| \end{cases} \quad (12)$$

and

$$\begin{cases} \left|g_k^{\mathrm{T}} S_k\right| \geq \left|\alpha_k g_k^{\mathrm{T}} d_{k-2}\right| \\ \left|g_k^{\mathrm{T}} d_k\right| \geq (1+\Delta_2)\left|\alpha_k\right|\left\|g_k\right\|.\left\|d_{k-2}\right\| \end{cases} \quad (13)$$

where $\Delta_1 > 0$ and $\Delta_2 > 0$ are constants it follows from (12) that

$$g_k^{\mathrm{T}} g_k - \beta_k g_k^{\mathrm{T}} d_{k-1} \geq (1+\Delta_1)\left|\beta_k\right|\left\|g_k\right\|\left\|d_{k-1}\right\| \quad (14)$$

Theorem:

If x_k is not a stationary point for problem (1) then:

$$\left\|d_k\right\| \leq (1 + \frac{1}{\Delta_1} + \frac{1}{\Delta_2})\left\|g_k\right\|. \quad (15)$$

Moreover, if d_k is descent then:

$$g_k^{\mathrm{T}} d_k \leq -\frac{1+\Delta_1}{2+\Delta_2} + \frac{1+\Delta_2}{2+\Delta_2}\left\|g_k\right\|^2 \quad (16)$$

for the prove of this theorem see [6].

Outline of the Three-Term Memory Gradient Algorithm (TMG):

Step1: let $x_0 \in \Re^n$ be initial point, $\Delta_1 > 0$, $\Delta_2 > 0$, compute g_0; if $g_0 = 0$ and x_0 is a stationary point of (1) stop; else set $d_0 = -g_0$, let $k = 1$ and go to step2.

Step2: let $x_{k+1} = x_k + \lambda_k d_k$; the step size λ_k is defined in the following way $\lambda_k = \min\left\{\alpha > 0 : g_{k+1}^{\mathrm{T}} d_k = \mu g_k^{\mathrm{T}} d_k \; where \; \mu \in [0,1)\right\}$

Step3: compute g_{k+1}; if $\left\|g_{k+1}\right\| = 0$ and x_{k+1} is a stationary point of (1) stop; else let $k + k + 1$, go to step4.

Step4: set $d_k = -g_k + \beta_k d_{k-1} + \alpha_k d_{k-2}$ where

$\beta_k \in \left[-\underset{-k}{\beta}(\Delta_1), \bar{\underset{k}{\beta}}(\Delta_2)\right]$,

$\alpha_k \in \left[-\underset{-k}{\alpha}(\Delta_1, \Delta_2), \bar{\underset{k}{\alpha}}(\Delta_1, \Delta_2)\right]$, go to step2.

where

$$\bar{\underset{k}{\alpha}}(\Delta_1, \Delta_2) = \frac{1+\Delta_1}{2+\Delta_2} \frac{1}{(1+\Delta_2) + \cos\bar{\theta}_k} \frac{\left\|g_k\right\|}{\left\|d_{k-2}\right\|} \quad (17)$$

$$\underset{-k}{\alpha}(\Delta_1, \Delta_2) = \frac{1+\Delta_1}{2+\Delta_2} \frac{1}{(1+\Delta_2) - \cos\bar{\theta}_k} \frac{\left\|g_k\right\|}{\left\|d_{k-2}\right\|}; \quad (18)$$

$\bar{\theta}_k$ angle between g_k, d_{k-2} and

$$\bar{\beta}_k(\Delta_1) = \frac{1}{(1+\Delta_1)+\cos\bar{\theta}_k}\frac{\|g_k\|}{\|d_{k-1}\|}, \qquad (19)$$

$$\beta_{-k}(\Delta_1) = \frac{1}{(1+\Delta_1)-\cos\bar{\theta}_k}\frac{\|g_k\|}{\|d_{k-1}\|}. \qquad (20)$$

A NEW THREE-TERM PRECONDITIONED GRADIENT MEMORY ALGORITHM

In this section we introduced a line search rule to find the best step-size parameter along the search direction at each iteration . We studied the convergent analysis of the modified Armijo step-size rules fully described in Armijo [2] given in step3 of the following new algorithm.

Outline of the New Three-Term Preconditioned Gradient Memory algorithm (NEW):

Step1: Let $x_0 \in \Re^n$ be initial point, compute g_0; if $g_0 = 0$ and x_0 is a stationary point of (1) stop; else let H_1 is any positive definite matrix usually $H_1 = I$ and ε is a small positive value , let $k = 1$ set $d_1 = -H_1 g_1$

Step2: if $\|g_k\| < \varepsilon$ then stop! Else go to step3

Step3: $x_{k+1} = x_k + \alpha_k d_k$ the step size α_k is chosen by the modified Armijo line search rule, namely: for given $q > 1$, $\mu_1 \in (0,1)$, $\lambda_k = q^{-r^k}$ and r^k is the smallest non negative integer such that

$$f(x_k + q^{-r} d_k) \le f(x_k) + \mu_1 q^{-r} g_k^{\mathrm{T}} d_k \qquad (21)$$

Step4: compute g_{k+1}; if $\|g_{k+1}\| = 0$ and x_{k+1} is a stationary point of (1) stop; else let $k = k+1$, go to step5.

Step5:

$$d_k = \begin{cases} -H_k g_k & for \quad k = 1 \\ -g_{k-2} + \beta_k H_{k-1} d_{k-1} + \alpha_k H_{k-2} d_{k-2} & for \quad k \ge 2 \end{cases}$$

and H_{k+1} is updated by

$$H_{k+1} = \begin{cases} H_k - \dfrac{H_k Y_k Y_k^{\mathrm{T}} H_k^{\mathrm{T}}}{Y_k^{\mathrm{T}} H_k Y_k} + \theta_k W_k W_k^{\mathrm{T}} + V_k V_k^{\mathrm{T}} / V_k^{\mathrm{T}} Y_k \ \text{ if } Y_k^{\mathrm{T}} H_k Y_k / V_k^{\mathrm{T}} Y_k > 0.5 \\ where W_k = (Y_k^{\mathrm{T}} H_k Y_k)^{1/2} [V_k / V_k^{\mathrm{T}} Y_k - H_k Y_k / Y_k^{\mathrm{T}} H_k Y_k], \\ \left(H_k - \dfrac{H_k Y_k Y_k^{\mathrm{T}} H_k^{\mathrm{T}}}{Y_k^{\mathrm{T}} H_k Y_k} + \theta_k W_k W_k^{\mathrm{T}}\right)\sigma_k + V_k V_k^{\mathrm{T}} / V_k^{\mathrm{T}} Y_k \\ where W_k = (Y_k^{\mathrm{T}} H_k Y_k)^{1/2} [V_k / V_k^{\mathrm{T}} Y_k - H_k Y_k / Y_k^{\mathrm{T}} H_k Y_k], \sigma_k = Y_k^{\mathrm{T}} H_k Y_k / Y_k^{\mathrm{T}} V_k \end{cases}$$

Step 7: If the available storage is exceeded, then employ a restart option either with $k = n$ or $g_{k+1}^{\mathrm{T}} g_{k+1} > g_{k+1}^{\mathrm{T}} g_k$.

Step 8: Set $k = k+1$ and go to step 2

The convergence analysis of the new proposed algorithm:

Consider the new three-term Preconditioned gradient memory defined in (11). Let $S_k = -g_k H_k + \beta_k H_{k-1} d_{k-1}$ and order to ensure that d_k is a sufficient descent direction at the point x_k. we assumed that for $\Delta_1 = 0.67$ and $\Delta_2 = 3$ then:

$$\begin{cases} g_k^{\mathrm{T}} g_k > \left|\beta_k g_k^{\mathrm{T}} H_{k-1} d_{k-1}\right| \\ \left|g_k^{\mathrm{T}} S_k\right| \ge 1.067 |\beta_k| \|g_k\| . \|H_{k-1} d_{k-1}\| \end{cases} \qquad (22)$$

and

$$\begin{cases} \left|g_k^{\mathrm{T}} S_k\right| \ge \left|\alpha_k g_k^{\mathrm{T}} H_{k-2} d_{k-2}\right| \\ \left|g_k^{\mathrm{T}} d_k\right| \ge 4 |\alpha_k| \|g_k\| . \|H_{k-2} d_{k-2}\| \end{cases} \qquad (23)$$

from (22) we proposed the following property:

Property 1:

$$g_k^{\mathrm{T}} g_k - \beta_k g_k^{\mathrm{T}} d_{k-1} \ge 1.067 |\beta_k| \|g_k\| . \|H_{k-1} d_{k-1}\| \qquad (24)$$

Proof:

Case 1. To ensure that $\beta_k > 0$ let

$$\beta_k \le \frac{g_k^{\mathrm{T}} g_k}{1.067\|g_k\| . \|H_{k-1} d_{k-1}\| + g_k^{\mathrm{T}} H_{k-1} d_{k-1}} = \frac{1}{1.067 + \cos\theta_{k1}} . \frac{\|g_k\|}{\|H_{k-1} d_{k-1}\|} \qquad (25)$$

where θ_{k1} is the angle between g_k and $H_{k-1} d_{k-1}$

Case 2. To ensure that $\beta_k < 0$ let

$$\beta_k \geq \frac{\|g_k\|^2}{1.067\|g_k\|.\|H_{k-1}d_{k-1}\| - g_k^{\mathrm{T}} H_{k-1} d_{k-1}} \quad (26)$$
$$= \frac{1}{1.067 - \cos\theta_{k1}} . \frac{\|g_k\|}{\|H_{k-1}d_{k-1}\|}$$

where θ_{k1} is the angle between g_k and $H_{k-1}d_{k-1}$. Thus a new choice for β_k will be given by

$$\beta_k \in [-\beta_{k1}, \beta_{k2}] \quad (27)$$

$$\beta_{k1} = \frac{1}{1.067 + \cos\theta_{k1}} . \frac{\|g_k\|}{\|H_{k-1}d_{k-1}\|} \quad (28)$$

$$\beta_{k2} = \frac{1}{1.067 - \cos\theta_{k1}} . \frac{\|g_k\|}{\|H_{k-1}d_{k-1}\|} \quad (29)$$

where θ_{k1} is the angle between g_k and $H_{k-1}d_{k-1}$.

from (23) we proposed the following property:

Property 2:

$$-g_k^{\mathrm{T}} S_k - \alpha_k g_k^{\mathrm{T}} H_{k-2} d_{k-2} \geq 4|\alpha_k| \|g_k\| . \|H_{k-2} d_{k-2}\| \quad (30)$$

Proof:

Case 1. To ensure that $\alpha_k > 0$ let

$$\alpha_k \leq \frac{g_k^{\mathrm{T}} S_k}{4.\|g_k\|.\|H_{k-2}d_{k-2}\| + g_k^{\mathrm{T}} d_{k-2}} \quad (31)$$

It follows from (23) that $g_k^{\mathrm{T}} S_k \leq -\frac{1.067}{4}\|g_k\|^2$ which implies that

$$\alpha_k \leq \frac{1.067}{5} \frac{1}{4 + \cos\theta_{k2}} \frac{\|g_k\|^2}{\|g_k\|.\|H_{k-2}d_{k-2}\|}$$
$$= \frac{1.067}{5} \left(\frac{1}{4 + \cos\theta_{k2}} \frac{\|g_k\|}{\|H_{k-2}d_{k-2}\|}\right)$$

where θ_{k2} is the angle between g_k and $H_{k-2}d_{k-2}$

Case 2. To ensure that $\alpha_k < 0$ let

$$\alpha_k \geq \frac{-g_k^{\mathrm{T}} S_k}{\|g_k\| - 4.\|g_k\|\|H_{k-2}d_{k-2}\|}$$

It follows from (23) that $g_k^{\mathrm{T}} S_k \leq -\frac{1.067}{4}\|g_k\|^2$ which implies that

$$\alpha_k \geq \frac{-1.067}{5} . \frac{1}{4 - \cos\theta_{k2}} \frac{\|g_k\|^2}{\|g_k\|.\|H_{k-2}d_{k-2}\|}$$
$$= \frac{-1.067}{5} . \left(\frac{1}{4 - \cos\theta_{k2}} \frac{\|g_k\|}{\|H_{k-2}d_{k-2}\|}\right)$$

where θ_{k2} is the angle between g_k and $H_{k-2}d_{k-2}$

Thus a new choice for α_k is given by

$$\alpha_k \in [-\alpha_{k1}, \alpha_{k2}] \quad (32)$$

$$\alpha_{k1} = \frac{1.067}{5} . \left(\frac{1}{4 + \cos\theta_{k1}} . \frac{\|g_k\|}{\|H_{k-2}d_{k-2}\|}\right) \quad (33)$$

$$\alpha_{k2} = \frac{1.067}{5} . \left(\frac{1}{4 - \cos\theta_{k2}} . \frac{\|g_k\|}{\|H_{k-2}d_{k-2}\|}\right) \quad (34)$$

where θ_{k2} is the angle between g_k and $H_{k-2}d_{k-2}$.

The descent property of the new proposed algorithm:

If x_k is not a stationary point for problem (1) then the search directions d_k of the new proposed algorithm are descent directions i.e.

$$g_k^{\mathrm{T}} d_k \leq -\frac{1.067}{5}\|g_k\|^2 \quad (35)$$

Proof: For $k = 1$,it is clear that $d_1 = -H_1 g_1$ where $H_1 = I$ identity matrix is a descent direction since for $k \geq 2$ it follows from (22)

$$g_k^{\mathrm{T}} S_k = -g_k^{\mathrm{T}} H_k g_k + \beta_k g_k^{\mathrm{T}} H_{k-1} d_{k-1}$$
$$= -\|g_k H_k\|^2 + \left|\beta_k g_k^{\mathrm{T}} H_{k-1} d_{k-1}\right|$$
$$= -\|g_k H_k\|^2 + \frac{1}{1.067} g_k^{\mathrm{T}} S_k$$

The above inequality $\left|g_k^{\mathrm{T}} S_k\right| = -g_k^{\mathrm{T}} S_k$ imply that

$$g_k^{\mathrm{T}} d_k \le -\frac{1.067}{5}\|g_k\|^2$$

RESULTS AND DISCUSSION

In this section we report some numerical results obtained by newly-written Fortran procedure with double precision.

Table 1: Comparison between the standard Three Term Memory Gradient (TMG) algorithm and New proposed algorithms using different values of 5≤N≤1000 for the 1st group of test functions

N. of test	Test function	TMG NOF(NOI) 5	10	100	1000	New NOF(NOI) 5	10	100	1000
1	GEN-Trid1	117 58	169 84	374 185	365 181	42 27	74 43	100 56	100 56
2	Shanno	1980 990	71 34	50 24	246 272	40 25	56 35	28 21	34 25
3	QF1	54 24	125 60	1187 591	1765 883	16 11	26 16	114 62	532 273
4	Gen-Rosen	1583 778	1655 814	1835 904	1015 508	124 78	120 75	118 79	124 78
5	NON-Digonal	653 306	857 389	791 367	826 385	76 49	150 97	108 64	132 88
6	TPQ	125 62	211 105	1259 5919	1839 920	20 13	30 19	144 77	656 335
7	GQ2	28 13	26 12	26 12	28 13	16 11	18 12	24 15	30 18
8	GEN-Powell	195 97	203 101	223 111	239 119	28 17	28 17	30 18	30 18
9	Tridia	75 37	245 122	2965 1482	3507 1754	22 16	74 43	452 234	1266 641
10	APQ	49 24	119 59	1171 585	1655 828	14 10	28 17	132 72	672 344
11	GEN-Helical	2804 1399	2940 1467	3240 1617	3528 1716	78 46	78 46	80 47	88 47
12	dqudratic	2595 1297	2589 1294	2583 1291	2007 1003	42 28	36 24	30 20	24 17
13	GEN-beal	819 409	859 429	963 481	1067 533	24 15	24 15	24 15	26 16
General total of 7 functions		1107 5494	10069 4970	1667 13569	18087 9160	542 346	742 459	1384 480	3714 1456

In comparison of algorithms the function evaluation is normally assumed to be the most costly factor in each iteration and the number of iterations. The actual convergence criterion employed was $\|g_k\|<1\times10^{-6}$ for the two algorithms, twenty one well-known test functions[9] (Appendices 1 and 2) and with dimensionality ranging (5-1000) are employed in the comparison. We solve each of these test function by the:

- Three Term Memory Gradient algorithm (TMG)
- The New proposed (New) algorithm

All the numerical results are summarized in Table 1, 2 and 3. They present the Number of Iterations(NOI) versus the Number of Function Evaluations (NOF) while Table 3 give the percentage performance of the new algorithm based on both (NOI) and (NOF) against the original (TMG) algorithm.

The important thing is that the new algorithm solves each particular problem measured by (NOI) and (NOF) respectively, while the other algorithm may fail in some cases. Moreover, the new proposed algorithm always performs more stably and efficiently.

Table 2: Comparison between the standard Three Term Memory Gradient(TMG) algorithm and New proposed algorithms using different value of 5≤N≤1000 for the 2nd group of test function

N. of test	Test function	TMG NOF(NOI) 5	10	100	1000	New NOF (NOI) 5	10	100	1000
1	Biggsb	F	F	F	F	8 7	24 15	140 73	1234 620
2	GEN-PowelI	F	F	F	F	94 53	94 54	102 57	106 57
3	GEN-Cubic	F	F	F	F	68 39	68 39	68 39	70 40
4	GEN- QDP	F	F	F	F	18 12	18 13	478 248	128 70
5	Fred	F	F	F	F	32 24	32 24	32 24	32 24
6	Sinquad	F	F	F	F	66 43	58 36	220 130	260 150
7	EX-with host	F	F	F	F	68 39	68 39	68 39	70 40
8	GEN- Wood	F	F	F	F	444 244	426 231	444 246	430 234

Table 3: Percentage performance of the standard Three Term Memory Gradient (TMG) algorithm against and New algorithm for 100% in both NOI and NOF

N	Costs	New
5	NOF NOI	95.107 93.70
10	NOF NOI	92.63 90.76
100	NOF NOI	91.70 96.46
1000	NOF NOI	79.47 84.11

Namely there are about (7-16)% improvements of NOI for all dimensions Also there are (5-21)% improvements of NOF for all test functions.

CONCLUSIONS

In this study, we have three parameter family of preconditioned gradient algorithm suitable to solve nonlinear unconstrained optimization problems. The directions d_k generated by the algorithm satisfy both the sufficient descent and lie search condition, with an

inexact line search under standard Wolfe line search condition. We have proved the global convergence of the new algorithm and examines[7] their computational performances.

Computational experience shows that the new proposed algorithm performs better than the standard three parameter family of preconditioned gradient memory method.

APPENDIX

Appendix 1: All the test functions used in Table (1) are from[2]:

Generalized tridiagonal-1 function:

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

$$x_0=[2.,2.,...,2.,2.].$$

Non-diagonal (Shanno-78) Function (Cute):

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

$$x_0=[-1.,-1.,...,-1.,-1.].$$

Quadratic QF1 function:

$$f(x)=\tfrac{1}{2}\sum_{i=1}^{n}ix_i^2-x_n,$$

$$x_0=[1.,1.,...,1.,1.].$$

Generalized rosen brock banana function:

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

$$x_0=[-1.2,1.,...,-1.2,1.]$$

Generalized Non diagonal function:

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

$$x_0=[-1.,...,-1.].$$

Tri-diagonal perturbed quadratic function:

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

$$x_0=[0.5,0.5.,...,0.5,0.5].$$

Generalized quadratic function GQ2:

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

$$x_0=[1.,1.,...,1.,1.].$$

Generalized Powell3 function:

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

$$x_0=[0.,1.,2.,...,0.,1.,2.].$$

Tri-diagonal function:

$$f(x)=\gamma(\delta x_1-1)^2+\sum_{i=2}^{n}i(\alpha x_i-\beta x_{i-1})^2,$$

$$x_0=[1.,1.,...,1.,1.],\ \alpha=1,\ \beta=1,\ \gamma=1,\ \delta=1.$$

Almost perturbed quadratic function:

$$f(x)=\sum_{i=1}^{n}ix_i^2+\tfrac{1}{100}(x_1+x_n)^2,$$

$$x_0=[0.5,0.5,...,0.5,0.5].$$

General helical function:

$$f(x)=\sum_{i=1}^{n/3}\left(100x_{3i}-10*H_i\right)^2+100(R_i-1)^2+x_{3i}^2,$$

where $R_i=\text{sqrt}(x_{3i-2}^2+x_{3i-1}^2),H_i=\dfrac{\tan^{-1}\dfrac{x_{3i-1}}{x_{3i-2}}}{2.\text{PI}}$

$$x_0=[-1.,0.,0....,-1.,0.],0..$$

D-quadratic Function (CUTE):

$$f(x)=\sum_{i=1}^{n-2}\left(x_i^2+cx_{i+1}^2+dx_{i+2}^2\right),$$

$$x_0=[3.,3.,...,3.,3.]\ ,\ c=100,d=100.$$

Generalized Beale Function:

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

$$x_0=[-1.,-1.,...,-1.,-1.].$$

Appendix 2: All the test functions used in Table (2) are from[2]

Biggsb1 Function (CUTE):

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

$$x_0=[1.,1.,...,1.,1.].$$

Generalized Powell function:

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

$$x_0=[0.,1.,2.,...,0.,1.,2.].$$

Generalized Cubic function:

$$f(x)=\sum_{i=1}^{n/2}[100(x_{2i}-x_{2i-1}^3)^2+(1-x_{2i-1})^2],$$
$$x_0=[-1.2,1.,...,-1.2,1.].$$

Quadratic diagonal perturbed function:

$$f(x)=\left(\sum_{i=1}^{n}x_i\right)^2+\sum_{i=1}^{n}\frac{i}{100}x_i^2,$$
$$x_0=[0.5,0.5,...,0.5,0.5].$$

Extended fred function:

$$f(x)=\sum_{i=1}^{n/2}(-13+x_{2i-1}+(5-x_{2i})+(x_{2i}-2)(x_{2i}))^2+\sum_{j=1}^{n/2}(-29+x_{2i-1}+(1-x_{2i})+(x_{2i}-14)(x_{2i}))^2,$$
$$x_0=(1.,2.,...,n)^T$$

Sinquad Function (CUTE):

$$f(x)=(x_i-1)^4+\sum_{i=1}^{n/2}\left(\sin(x_i-x_n)-x_1^2+x_i^2\right)^2+(x_n^2-x_1^2)^2,$$
$$x_0=[0.1,0.1,...,0.1].$$

Extended white and holst function:

$$f(x)=\sum_{i=1}^{n/2}c(x_{2i}-x_{2i-1}^3)^2+(1-x_{2i-1})^2,$$
$$x_0=[-1.2,1.,...,-1.2,1] \quad , \quad c=100.$$

Generalized wood function:

$$f(x)=\sum_{i=1}^{n/2}\begin{matrix}100(x_{4i-2}-x_{4i-2}^2)^2+(1-x_{4i-3})^2+90(x_{4i}-x_{4i-1}^2)^2\\+(1-x_{4i-1})^2+10.1(x_{4i-2}-1)^2+(x_{4i}-1)^2\\+19.8(x_{4i-2}-1)(x_{4i-2}-1)\end{matrix},$$
$$x_0=[-3.,-1.,-3.,-1.,...,-3.,-1.,-3.,-1.].$$

REFERENCES

1. Al-Assady, N.H., Al-Bayati, A.Y., 1994. Minimization of extended quadratic functions with inexact line Searches. J. Optim. Theo. Appl. 82, 139–147, doi> 10.1007/BF02191784.
2. Armijo, L., 1966. Minimization of function having Lipschitz continuous first partial derivatives, pacific. J. Math., 16, 1-13. http://projecteuclid.org/euclid.pjm/1102995080
3. Cohen, A., 1981. Step-size analysis for descent methods, IEEE Xplore, 12, 417-421. doi> 10.1109/CDC.1973.269201.
4. Cragg, E.E., Levy, A.V., 1969. Study on a super memory gradient method for the minimization of functions, J. Optim. Theo. Appl.4, 191–205. doi> 10.1007/BF00930579.
5. Gill, P.E., Murray, W., 1975. Conjugate gradient methods for large scale nonlinear optimization, Technical Report SOL 79-15, Stanford University, http://stinet.dtic.mil/stinet/jsp/advanced-tr.jsp
6. Hager, W., Zhang, H., 2005. A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. 16, 170–192. http://portal.acm.org/citation.cfm?id=1085581&dl=&coll=
7. More', J.J, Garbow, B.S., Hillstrom, K.E., 1981. Testing unconstrained optimization software. ACM Transactions on Mathematical Software 7, 17–41, http://portal.acm.org/citation.cfm?doid=355934.355936
8. Sun, Q., 2004. Global Convergence of a Class of New Three-term Memory Gradient Methods with Curry-Altman and Armijo Step-size Rules, Soochow Journal of Mathematics 30, 55–66. http://143.248.27.21/mathnet/thesis_content.php?no=367448
9. Yabe, H., Takano, M., 2004. Global convergence properties of nonlinear conjugate gradient methods with modified secant condition. Computational Optim. Appl. 28,203–225. doi>10.1023/B:COAP.0000026885. 81997. 88