A Modified Super-linear QN-Algorithm for
Unconstrained Optimization

Abbas Y. AL-Bayati*
Ivan Subhi Latif**

ABSTRACT

In this paper, we have proposed a modified QN-algorithm for solving a self-scaling large scale unconstrained optimization problems based on a new QN-update. The performance of the proposed algorithm is better than that used by Wei, Li, Yuan algorithm. Our numerical tests show that the new proposed algorithm seems to converge faster as compared with a standard similar algorithm in many situations.

*Prof. Dr/ Dept. of Mathematics/College of Computers Sciences and Mathematics/ University of Mosul. E-mail :profabbaslbayati@yahoo.com

**Dept. of Mathematics, College of Scientific Education, University of Salahaddin, Erbil-Iraq. E-mail :ivansubhi2001@yahoo.com

Received:28/8 /2006 _______________ Accepted: 17 /12 / 2006
KEYWORDS: Unconstrained Optimization, Quasi-Newton updates, Super-linear convergence.

1. Introduction

This paper analyzes the convergence properties of self-scaling QN-methods for solving the unconstrained optimization problem

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

where $f$ is twice continuously differentiable function. The convergence of QN-methods for unconstrained Optimization has been the subject of much analysis. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is generally considered to be the most effective among other variable metric methods for unconstrained Optimization problem. One interesting property of BFGS method is its self correcting mechanism (A detailed explanation for example, in Nocedal (see [11]) with this self correcting property, Powell see[13] shows that the BFGS method with an inexact line search satisfies Wolfe conditions is globally super-linearly convergent for convex problem, and Byrd, Nocedal and Yuan (see [5]) extend Powell's analysis to the restricted Broyden class excluding the DFP method. AL-Bayati's (see [1]) presented a new self-scaling variable-metric algorithm which was based on a known two-parameter family of rank-two updating formulae. The best of these algorithm are also modified to employ inexact line searches with marginal effect thus Wei, Li and Qi (see [15]) have proposed some modified BFGS that the average performance of their algorithm was better than standard BFGS algorithm.

Wei, et al. (see [14]) proved the super-linear convergence of Wei,
Li and Qi (see [15]) algorithm under some suitable conditions. In this paper, a new modified QN-algorithm is proposed. The Basic idea is based on the new QN-equation \( V_k = H_k y_k^* \) where \( y_k^* \) is the sum of \( y_k \) and \( A_k V_k \) and \( A_k \) is some matrix.

This paper is organized as follows in the next section; we represent some basic properties of the modified BFGS algorithm. In section 3 we prove the super-linear convergence for the modified QN-algorithm under some reasonable conditions. The search direction in a VM-method is the solution of the system of equations

\[
d_k = -H_k g_k
\]

where the matrix \( H_k \) is an approximate to \( G_k^{-1} \) the new approximation \( H_{k+1} \) is chosen to take account of this new curvature information which is done by satisfying the condition

\[
H_{k+1} y_k = \zeta_k V_k \quad \text{(called QN-like condition)}
\]

where \( \zeta_k \) is a scalar, generally for the QN-methods \( \zeta_k = 1 \) and hence equation (3) reduces to

\[
H_{k+1} y_k = V_k \quad \text{(called the QN-condition)}
\]

since information has been gained about \( f \) only in one dimension (along \( d_k \)), \( H_{k+1} \) is allowed to differ from \( H_k \) by a correction matrix \( C_k \) of at most rank two, i.e.

\[
H_{k+1} = H_k + C_k
\]

the matrix \( C_k \) is therefore the update to \( H_k \) there are an infinite number of possible rank-two updates which satisfy the QN-condition but our main interest is in updates which form the Broyden one-parametric class (see [3]). The matrix \( H_{k+1} \) is defined by:
\[ H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \theta_k W_k W_k^T + \frac{V_k V_k^T}{V_k^T y_k} \] (6)

with

\[ W_k = (y_k^T H_k y_k) \begin{bmatrix} V_k \\ V_k^T \\ y_k^T H_k y_k \end{bmatrix} \] (7)

where \( \theta_k \) is a scalar chosen such that \( \theta_k \in [0, 1] \) different choice of \( \theta_k \) then defined different updates. The Davidon-Fletcher-Powell (DFP) update (see [7]) is defined as equation (4) with \( \theta_k = 0 \) where the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update corresponds to \( \theta_k = 1 \) (see [6] and [9]). Oren (see [13]) found that a proper scaling of the objective function improve the performance of algorithms that use Broyden family of updates. Hence Oren's family of self-scaling VM-updates can be expressed as:

\[ H_{k+1} = \left[ H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \theta_k W_k W_k^T \right] \eta_k + \frac{V_k V_k^T}{V_k^T y_k} \] (8)

where

\[ \eta_k = \frac{y_k y_k^T V_k}{y_k^T H_k y_k} \] (9)

This choice for the scalar parameter \( \eta_k \) was made primarily because in this case \( \eta_k \) requires the quotient of two quantities which are already computed in the updating formula. Al-Bayati (see [2]) found another interesting family of VM-updates by further scaling of Oren's family of updates with a scalar

\[ \delta_k = \frac{1}{\eta_k} \] (10)

So that the updating formulas becomes
\[ H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + W_k W_k^T + \sigma_k \left( \frac{V_k V_k^T}{V_k^T y_k} \right) \]  

(11)

For more details see [8]

2. Modified BFGS Algorithm:

Wei, Li and Qi proposed a new QN-equation (see [15])

\[ B_{k+1} V_k = y_k^* \]  

(12)

where \( B_k = H_k^{-1} \) where \( y_k^* = y_k + A_k V_k \) and \( A_k \) is some matrix. By using equation (12) they gave BFGS type updates

\[ B_{k+1} = B_k - \frac{B_k V_k y_k^* V_k^T B_k + y_k^* y_k^T}{y_k^T B_k V_k} \]  

(13)

where \( y_k^* = y_k + A_k V_k \) and \( A_k = \frac{2(f(x_k) - f(x_{k+1})) + (g(x_{k+1}) - g(x_k))^T}{\| V_k \|^2} \) using equation (13) and the following Wolf Powell step-size rule

\[ f(x_k + \alpha_d d_k) \leq f(x_k) + \delta \alpha_d g(x_k)^T d_k \]  

(14)

where \( \delta \in (0, \frac{1}{2}) \) and \( \sigma \in (0, 1) \) and

\[ g(x_k + \alpha_d d_k)^T d_k \geq \sigma g(x_k)^T d_k \]  

(15)

2.1. Outline of the Modified BFGS Algorithm (MBFGS):

Corresponding (MBFGS) the outliers of MBFGS algorithms may be listed

Step 1: choose an initial point \( x_0 \in \mathbb{R}^n \) and an initial positive definite matrix \( B_0 \) set \( K = 1 \).
Step 2: if \( \|g_k\| = 0 \) then stop! Go to step 2.

Step 3: solve \( H_k d_k + g_k = 0 \) to obtain a search direction \( d_k \).

Step 4: find \( d_k \) by Wolf-Powell step-size rule

\[
f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g(x_k)^T d_k \quad \text{and} \quad g(x_k + \alpha_k d_k)^T d_k \geq \sigma g(x_k)^T d_k \quad \text{where}
\]

\( \delta \in (0, \frac{1}{2}) \) and \( \sigma \in (\delta, 1) \) and

Step 5: set \( x_{k+1} = x_k + \alpha_k d_k \) calculate \( A_k \) and update \( B_{k+1} \) by formula

\[
B_{k+1} = B_k - \frac{B_k y_k^T B_k}{y_k^T B_k y_k} + \frac{y_k^* y_k^{*T}}{y_k^T y_k}
\]

where \( y_k^* = y_k + A_k y_k \) and \( A_k = \frac{2(f(x_k) - f(x_{k+1})) + (g(x_{k+1}) - g(x_k))^T}{\|y_k\|^2} \).

Step 6: set \( k = k + 1 \), go to step 1.

2.2. Some Properties of the MBFGS Algorithm:

The global convergence of the MBFGS algorithm needs the following three assumptions

**Assumption 2.2.1:** The level set \( \{ x \mid f(x) \leq f(x_0) \} \) is contained in a bounded convex set \( D \).

**Assumption 2.2.2:** The function \( f \) is continuously differentiable on \( D \) and there exists constant \( L \geq 0 \) such that \( \|g(x) - g(y)\| \leq L \|x - y\| \) for all \( x, y \in D \).

**Assumption 2.2.3:** The function \( f \) is uniformly convex that is there are positive constants \( m_1 \) and \( m_2 \) such that

\[
m_1 \|z\|^2 \leq z^T G(x) z \leq m_2 \|z\|^2
\]  

(16)
for all \( x, z \in \mathbb{R}^n \), where \( G \) denotes the Hessian matrix of \( f \).

**Theorem 2.1:** Let \( \{x_k\} \) be generated by MBFGS algorithm then we have (see [11])

\[
\liminf_{k \to 0} \| g_k \| = 0. \tag{17}
\]

The super-linear convergence analysis of the MBFGS algorithm needs the following assumptions:

**Assumption 2.2.4:** \( x_k \to x^* \) at which \( g(x^*) = 0 \) and \( G(x^*) \) is positive definite.

**Assumption 2.2.5:** \( G \) is holder continuous at \( x^* \) that is, there exists constant \( r \in (0,1) \) and \( M > 0 \) such that

\[
\| G(x) - G(x^*) \| \leq M \| x - x^* \|^r \tag{18}
\]

for all \( x \) in neighborhood of \( x^* \) since \( \{f(x_k)\} \) is a decreasing sequence also the sequence \( \{x_k\} \) generated by MBFGS is contained in \( \Omega \) and there exists a constant \( f^* \) such that

\[
\lim_{k \to \infty} f(x_k) = f^* \tag{19}
\]

### 3. A new Modified QN-Algorithm:

In this section we propose a new QN-method based on the following QN-condition

\[
V_k = H_k y_k^* \tag{20}
\]

where \( y_k^* = y_k + A_k V_k \) and \( A_k \) is some matrix defined by

\[
A_k = \frac{y_k^T V_k}{y_k^T H_k y_k} I
\]

using equation (20) and taking \( H_{k+1} \) as Al-Bayati update (see [1]).
\[ H_{k+1} = \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + W_k W_k^T + \sigma_k \left( \frac{V_k V_k^T}{y_k^T H_k y_k} \right) \]

where
\[ \delta_k = \frac{y_k^T H_k y_k}{y_k^T V_k}, \quad W_k = (y_k^T H_k y_k)^{\varepsilon} \left[ \frac{V_k}{y_k^T H_k y_k} - \frac{H_k y_k}{y_k^T H_k y_k} \right] \]

and using also the following Armijo condition
\[ f_k - f_{k+1} \geq c_1 \| V_k^T y_k \|, \quad V_k^T y_k \geq (1 - c_2) (V_k^T y_k) \quad (21) \]

where \( c_1 \in (0, \frac{1}{2}), c_2 \in (c_1, 1) \) in (MBFGS) yields a new QN-algorithm given as below:

3.1. Outline of the Modified QN-Algorithm (NEW):

The outliers of the new algorithm may be given as:

Step 1: choose an initial point \( x_0 \in \mathbb{R}^n \) and use-update positive definite matrix \( H_1 \) set \( K = 1 \).

Step 2: if \( \| g_k \| = 0 \) then stop! Go to step 2.

Step 3: solve \( H_1 d_k + g_k = 0 \) to obtain a search direction \( d_k \)

Step 4: find \( d_k \) by Armijo line search step-size rule
\[ f_k - f_{k+1} \geq c_1 \| V_k^T y_k \|, \quad V_k^T y_k \geq (1 - c_2) (V_k^T y_k) \]
where \( c_1 \in (0, \frac{1}{2}), c_2 \in (c_1, 1) \)

Step 5: set \( x_{k+1} = x_k + \alpha_k d_k \), calculate a new \( A_k \) and update \( H_{k+1} \) by the following formula
\[ H_{k+1} = H_k - \frac{H_k y_k^* y_k^{*\top} H_k}{y_k^{*\top} H_k y_k^*} + W_k W_k^\top + \sigma_k \left( \frac{V_k^T y_k^*}{V_k^T y_k^*} \right) \]  

(22)

where

\[ \delta_k = \frac{y_k^{*\top} H_k y_k^*}{y_k^{*\top} V_k^*}, \quad W_k = (y_k^{*\top} H_k y_k^*)^{-1} \left[ \frac{V_k^T y_k^*}{V_k^T y_k^*} - \frac{H_k y_k^*}{y_k^{*\top} H_k y_k^*} \right] \]

and \( y_k^* = y_k + A_k V_k \) and \( A_k = \frac{y_k^{*\top} V_k^*}{y_k^{*\top} H_k y_k^*} I \) (newly defined)  

(23)

Step 6: set \( k = k + 1 \), go to step 1.

3.2. Some Theoretical Properties of the New Algorithm:

To show the global and super-linear convergence of the new algorithm use the same assumption 2.2.1,2.2.2,2.2.3 and 2.2.4 where used for all \( x \) in neighborhood of \( x^* \) since \( \{f(x_k)\} \) is a decreasing sequence also the sequence \( \{x_k\} \) generated by new algorithm is contained in \( \Omega \) and that there exists a constant \( f^* \) such that

\[ \lim_{k \to \infty} f(x_k) = f^* \]  

(24)

lemma 3.2.1: let \( (\alpha_k, x_{k+1}, g_{k+1}, d_{k+1}) \) be generated by the new algorithm then \( H_{k+1}^{-1} \) is positive definite for all \( k \) provided that \( V_k^T y_k^* > 0 \).

Proof: –

The new algorithm has the following QN-condition

\[ H_{k+1} V_k = y_k \]  

(25)

and preserve positive definiteness of the matrices \( \{H_k\} \) if \( \alpha_k \) is chosen to satisfy the Armijo condition
\[ f_k - f_{k+1} \geq c_1 \|V_k^T y_k\|, \quad V_k^T y_k \geq (1-c_2)(V_k^T y_k) \]

where \( f_k \) denoted \( f(x_k) \), \( c_1 \in (0, 1/2), c_2 \in (1, 1) \). Note that the second condition in (21) guarantees that \( V_k^T y_k > 0 \) whenever \( g_k \neq 0 \).

**Lemma 3.2.2:** let \( \{x_k\} \) be generated by the new algorithm then we have

\[
m_1 \|y_k\|^2 \leq V_k^T y_k^* \leq m_2 \|y_k\|^2, \quad k = 1, 2, \ldots \quad (26)
\]

and

\[
\|y_k^*\| \leq (2L + m_2)\|V_k\|, \quad k = 1, 2, \ldots \quad (27)
\]

**Proof:**

Using assumption 2.2.2 and equation (16)

\[
m_1 \|z\|^2 \leq z^T G(x) z \leq m_2 \|z\|^2
\]

and tailors formula we have

\[
y_k^T V_k = (g_{k+1} - g_k)^T V_k
\]

\[= V_k^T G(\xi) V_k\]

where \( \xi \in (x_k, x_{k+1}) \) thus (26) holds

\[
m_1 \|y_k\|^2 \leq V_k^T y_k^* \leq m_2 \|y_k\|^2.
\]

To prove (27) using the equation (26)

\[
V_k^T y_k^* \leq m_2 \|V_k\|^2
\]

therefore
\[ \|y_k\| \leq m_s\|\tilde{y}_k\|^2 \]  \hspace{1cm} (28)

also use assumption 2.2.2

\[ \|g(x) - g(y)\| \leq L\|x - y\|, \text{ for all } x, y \in D. \]

Thus

\[ \|y^*_k\| \leq L\|\tilde{y}_k\| \]  \hspace{1cm} (29)

from equation (28),(29) we get

\[ \|y^*_k\| \leq (2L + m_s)\|\tilde{y}_k\|. \]

**Theorem 3.1** let \( \{x_k\} \) be generated by the new algorithm then \( x_k \) tends to \( x \) super-linearly.(see[2])

**Lemma 3.2.3:** supose that \((\alpha_k, x_{k+1}, g_{k+1}, d_{k+1})\) be generated by the new algorithm and that \( G \) is continuous at \( x^* \) then we have

\[ \lim_{k \to \infty} A_k = 0 \]  \hspace{1cm} (30)

**Proof:**

By using Taylor's formula, we have

\[ y_k^*V_k = (g_{k+1} - g_k)^TV_k \]
\[ = V_k^TG(\zeta_k)V_k \]

and
\[ f_k - f_{k+1} = (g_{k+1} - g_k)^T V_k \]
\[ = V_k^T G(\zeta_2) V_k \]

and
\[ f_k - f_{k+1} = -g_{k+1}^T V_k + \frac{1}{2} V_k^T G(\zeta_2) V_k \]

where
\[ \zeta_1 = x_k + \theta_{1k} (x_{k+1} - x_k) \]
\[ \zeta_2 = x_k + \theta_{2k} (x_{k+1} - x_k) \]

and \( \theta_{1k}, \theta_{2k} \in (0,1) \). From the definition of \( A \) and lemma 3.2.1 we get
\[ A = \frac{V_k^T H_k^{-1} V_k - V_k^T G(\zeta_1) V_k}{y_k^T H_k y_k} \]

and
\[ V_k^T H_k^{-1} V_k = V_k^T G(\zeta_1) V_k \]

hence
\[ \|A_k\| \leq \|G(\zeta_1) - G(\zeta_2)\| \] Therefore (30) holds.

**Lemma 3.2.4:** Let \( \{x_k\} \) be generated by the new algorithm denoted \( Q = G(x^*)^{-\frac{1}{2}} \) then there are positive constants \( b_i, i = 1, 2, 3, 4 \) and \( \eta \in (0,1) \) such that for all large \( k \)
\[ \|H_k^{-1} G(x^*)^{-1}\| \leq \left( \sqrt{1 - \rho W_k^2} + b_1 \tau_k + b_2 \right) \|H_k - G(x^*)^{-1}\| + b_3 \tau_k + b_4 \|A_k\| \] (31)

where
\[ \|A\| = \|Q^T AQ\|_F \] , \( \|\cdot\|_F \) is the forbenius norm of a matrix and \( W_k \) is defined as
\[ W_k = \frac{\|Q^{-1}(H_k - G(x^*)^{-1}y_k^*)\|}{\|H_k - G(x^*)^{-1}\|\|Qy_k^*\|} \] (32)

In particular \(\|H_k\|, \|H_k^{-1}\|\) are bounded

Proof:

To prove (32)

\[
H_{k+1} = H_k + \frac{(V_k - H_k y_k^*)V_k^T + V_k (V_k - H_k y_k^*)^T}{y_k^{*T}V_k} + \frac{(y_k^*)^T(V_k - H_k y_k^*)V_k V_k^T}{(y_k^{*T}V_k)^2}
\]

\[
= \left( I - \frac{y_k^*V_k^T}{y_k^{*T}V_k} \right) H_k \left( I - \frac{y_k^*V_k^T}{y_k^{*T}V_k} \right) + \frac{V_k V_k^T}{y_k^{*T}V_k}
\]

It is the dual form of the DFP type algorithm in the sense that \(H_{k+1} \rightarrow H_k^{-1}\) and \(V_k \rightarrow y_k\) we also have

\[
\|Qy_k^* - Q^{-1}V_k\| \leq \|Q\| \|y_k^* - Q^{-2}V_k\|
\]

\[
= \|Q\| \|y_k^* - G(x^*)V_k\|
\]

\[
\leq \|Q\| \left\| \int_0^1 G(x_k + \eta V_k) d\eta - G(x^*)V_k \right\| + \|A\| V_k
\]

\[
\leq \|Q\| \|V_k\| \left\| \int_0^1 G(x_k + \eta V_k) V_k - G(x^*) d\eta \right\| + \|A\| V_k
\]

\[
\leq \|Q\| \|V_k\| \left( M_2 \int_0^1 \|x_k - x^* + \eta V_k\| d\eta + \|A\| \right)
\]

\[
\leq \|Q\| \|V_k\| \left( M_2 \left( \int_0^1 \|x_k - x^* + \eta V_k\| d\eta + \|A\| \right) \right)
\]
A Modified Super-linear QN-Algorithm ...

$$\leq \|Q\|\|V_k\|\left(M_2 \sum_0^1 (\eta \|x_k - x^*\| + (1 - \eta)\|x_k - x^*\|)^\beta \eta + \|A\|)$$

$$\leq \|Q\|\|V_k\|(M_2 \tau_k + \|A\|)$$

since \( \tau_k \to 0 \) and \( \|A\| \to 0 \) it is clear that when \( k \) is large enough

$$\|Qy^*_k - Q^{-1}V_k \leq \beta \|QV_k\|,$$

for some constant \( \beta \in (0, \frac{1}{2}) \), therefore from lemma 3.2.1 (with identification \( V \to y_k, y \to V_k, H^{-1} \to H_k, A \to G(x^*)^{-1} \) and \( M \to Q^{-1} \) there are constants \( p \in (0, 1) \) and \( b_7, b_8 > 0 \) such that

$$\|H_{k+1} - G(x^*)^{-1}\| \leq \left( \sqrt{1 - \rho W_k^2} + b_7 \frac{Q^{-1}V_k - Qy^*_k}{\|QV_k\|} \right) H_k - G(x^*)^{-1} + b_6 \frac{V_k - G(x^*)^{-1}y^*_k}{\|Qy^*_k\|}$$

where \( W_k \) is defined as more over, there exists a constant \( b_7 \), such that for all \( k \) large enough

$$\|y^*_{k+1} - Qg_{k+1} - g_k + AV_k\|$$

$$\geq \|Qg_{k+1} - g_k - \|A\|QV_k\|$$

$$\geq b_7 \|x_{k+1} - x_k\| - \|A\|\|V_k\|$$

$$= (b_7 - \|A\|\|Q\|)\|V_k\|$$

using \( \|A\| \to 0 \) the above inequality implies that there is a constant \( C \) such that when \( k \) is sufficiently large

$$\|Qy^*_k\| \geq C\|V_k\|$$

so we may obtain that
\[
\frac{\|Qy_i^* - Q^{-1}V_i\|}{\|Qy_i^*\|} \leq C^{-1}\|Q\|(M_2\tau_k + \|A\|) \quad (33)
\]

and
\[
\frac{\|V_i - G(x^*)^{-1}y_i\|}{\|Qy_i^*\|} = \frac{\|V_k - Q^2y_i\|}{\|Qy_i^*\|} = \frac{\|Q(Qy_i^* - Q^{-1}V_i)\|}{\|Qy_i^*\|} \leq \frac{\|Q\|\|Qy_i^* - Q^2V_i\|}{C\|V_i\|} \leq C^{-1}\|Q\|^2(M_2\tau_k + \|A\|)
\]

from which and (33), we get (31)

**Lemma 3.2.5:** Let \{x_k\} be generated by the new algorithm then the following Dennis More condition holds for the new technique

\[
\lim_{k \to \infty} \frac{\|H_k^{-1} - G(x)V_i\|}{\|V_i\|} = 0 \quad (34)
\]

**Proof:**

Using \(\tau_k \to 0\), \(\|A\| \to 0\) and \(\|H_k\|\) is bounded and following inequality

\[
\sqrt{1 - \tau} \leq 1 - \frac{1}{2}\tau, \quad \forall \tau \in (0,1)
\]

we can deduce that there are positive constants \(M_1\) and \(M_2\) such that for all large \(k\)

\[
\|H_k - G(x^*)^{-1}\| \leq (1 - \frac{1}{2}\rho W_k^2)\|H_k - G(x^*)^{-1}\| + M_1\tau_k + M_2\|A_k\|
\]
that is
\[
\frac{1}{2} \rho W_k^2 \| H_k - G(x^*)^{-1} \| \leq \| H_k - G(x^*)^{-1} \| - \| H_{k+1} - G(x^*)^{-1} \| + M_1 \tau_k + M_2 \| A_k \|
\]

summing the above inequality over \( k \), we get
\[
\frac{1}{2} \rho \sum_{k=k_0}^{\infty} W_k^2 \| H_k - G(x^*)^{-1} \| < +\infty
\]

where \( k_0 \) is sufficiently large index such that (31) holds for all \( k \geq k_0 \). In particular, we have
\[
\lim_{k \to \infty} W_k^2 \| H_k - G(x^*)^{-1} \| = 0
\]

that is
\[
\lim_{k \to \infty} \frac{\| Q^{-1}(H_k - G(x^*)^{-1} y_k^*) \|}{\| Q y_k^* \|} = 0
\] 

(35)

Moreover we have
\[
\| Q^{-1}(H_k - G(x^*)^{-1} y_k^*) \| = \| Q^{-1}(H_k (G(x^*) - H_k^{-1})) G(x^*)^{-1} y_k^* \|
\]
\[
\geq \| Q^{-1}(H_k (G(x^*) - H_k^{-1}) V_k) - Q^{-1}(H_k (G(x^*) - H_k^{-1}) (V_k - G(x^*))^{-1} y_k^* \|
\]
\[
\geq \| Q^{-1}(H_k (G(x^*) - H_k^{-1}) V_k) - Q^{-1}(H_k (G(x^*) - H_k^{-1}) (V_k - G(x^*))^{-1} y_k^* \|
\]
\[
\geq \| Q^{-1}(H_k (G(x^*) - H_k^{-1}) G(x^*)^{-1} V_k \|
\]

using the fact that \( \| H_k \| \) and \( \| H_{k+1} \| \) are bounded and that \( G(x) \) is continuous, we have
\[ \left\| Q^{-1}(H_k(G(x^*) - H_k^{-1})V_k - G(x^*)^{-1}y^*_k) \right\| \\
= \left\| Q^{-1}(H_k(G(x^*) - H_k^{-1})G(x^*)^{-1}(G(x^*)V_k - y^*_k) \right\| \\
= \left\| Q^{-1}(H_k(G(x^*) - H_k^{-1})G(x^*)^{-1}[G(x^*) - G(x_k)]V_k + (G(x_k)V_k - y_k) \right\| \\
\leq \left\| Q^{-1}(H_k(G(x^*) - H_k^{-1})G(x^*)^{-1} V_k \right\| G(x^*) - G(x_k) \right\| V_k \right\| G(x_k)V_k - y_k \right\| \\
= O(\|V_k\|) \\
\]

and
\[ \left\| A\left\| Q^{-1}(H_k(G(x^*) - H_k^{-1})V_k \right\| \right. \]
\[ \leq \left\| A\left\| Q^{-1}(H_k(G(x^*) - H_k^{-1})G(x^*)^{-1} \right\| V_k \right\| \\
= O(\|V_k\|) \] (36)

therefore, there exists a positive constant \( k > 0 \) such that
\[ \left\| Q^{-1}(H_k - G(x^*)^{-1}y^*_k) \right\| \geq k\left\| G(x^*) - H_k^{-1} \right\| \leq O(\|V_k\|) \]
on the other hand, from equation (36) and lemma 3.2 we have
\[ \|Qy^*_k\| \leq Q\|y^*_k\| \leq (2L + m)\|Q\|\|y^*_k\| \]
from the above inequality (35) and (36) we conclude that the Dennis-More condition holds.

\textbf{Theorem 3.2}

Let \( \{x_k\} \) be a sequence generated by the new algorithm then the \( x_k \) tends to \( x \) super-linearly.

Proof:-

We will verify that \( \alpha_k = 1 \) for large \( k \). since the sequence \( H_k \) is bounded we have
\[ \|d_k\| = \|H_k g_k\| \leq \|H_k\| \|g_k\| \rightarrow 0 \]

by Taylor's expansion, we get

\[ f_{k+1} - f_k - \delta g_k^T d_k = (1 - \delta) g_k^T d_k + \frac{1}{2} d_k^T G(x_k + \delta d_k) d_k \]

\[ = -(1 - \delta) d_k^T H^{-1} d_k + \frac{1}{2} d_k^T G(x_k + \delta d_k) d_k \]

\[ = -(1 - \delta) d_k^T H^{-1} d_k - \frac{1}{2} d_k^T (H_k^{-1} - G(x_k + \delta d_k)) d_k \]

\[ = -(1 - \delta) d_k^T G(x^*) d_k + O(\|d_k\|^2) \]

where \( \theta \in (0,1) \) and the last equality follows the Dennis More condition (34) thus

\[ f_{k+1} - f_k - \delta g_k^T d_k \leq 0 \]

for all large \( k \). In other words \( \alpha_k = 1 \) the first inequality of the Armijo equation (21) for all \( k \) sufficiently large on the other hand

\[ g_k^T d_k - \delta g_k^T d_k = (g_{k+1} - g_k)^T d_k + (1 - \sigma) g_k^T d_k \]

\[ = d_k^T G(x_k + \delta d_k) d_k - (1 - \delta) g_k^T H_k^{-1} d_k \]

\[ = d_k^T G(x_k + \delta d_k) d_k - (1 - \delta) d_k^T G(x_k) d_k + O(\|d_k\|^2) \]

\[ = \delta d_k^T G(x_k) d_k + O(\|d_k\|^2) \]

where \( \theta \in (0,1) \) so we have

\[ g_{k+1}^T d_k \geq \delta g_k^T d_k \]

which means that \( \alpha_k = 1 \) satisfies the Armijo equation (21) for all sufficiently large \( k \). Therefore we assert that \( \alpha_k = 1 \) for large \( k \).

Consequently, we can deduce that \( x_k \) converges super-linearly.
4. Numerical Results:

In this section, we compare the numerical behavior of the new algorithm with the MBFGS algorithm for different dimensions of test functions. Comparative tests were performed with (41)(specified in the Appendices 1 and 2) well-known test functions (see [10]). All the results are shown in Table (1), (2) while Table (3) give the percentage of NOI and NOF. All the results are obtained with newly-programmed FORTRAN routines which employ double precision. The comparative performances of the algorithms taken in the usual way by considering both the total number of function evaluations (NOF) and the total number of iterations required to solve the problem (NOI) . In each case the convergence criterion is that the value of \( \|g_i\| < 1 \times 10^{-5} \) the Armijo fitting by Frandsen (see [2]) and Powell line search (see [3]) used as the common linear search subprogram.

Each of the function was solved using the following algorithms

(1) MBFGS Algorithm:

(2) The new algorithm

The important thing is that the new algorithm needs less iteration, fewer evaluations of \( f(x) \) and \( g(x) \) than MBFGS. We can see that other algorithm may fail in some cases while the new algorithm always converges. Moreover numerical experiments also show that the new algorithm always convergence stabiles. Namely
there are about (60-87) % improvements of NOI for all dimensions
Also there are (30 -78) % improvements of NOF for all test
functions.

Table (1):Comparison between the New algorithm and MBFGS
algorithms using different value of 12< N <4320 for 1st test
function.

<table>
<thead>
<tr>
<th>N. OF Test</th>
<th>TEST FUNCTION</th>
<th>MBFGS</th>
<th>NOI(NOIF)</th>
<th>NEW</th>
<th>NOI(NOIF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GEN- Shallow</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>2</td>
<td>GEN- Beal</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>3</td>
<td>Arhad</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>4</td>
<td>GEN-Edger</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>5</td>
<td>Digonal4</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>6</td>
<td>EX- Denschnb</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>7</td>
<td>EX-BD1</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>8</td>
<td>Digonal5</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>9</td>
<td>GEN-Strail</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>10</td>
<td>Digonal6</td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>11</td>
<td>Digonal7</td>
<td>12</td>
<td>12</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>EX- Denschnf</td>
<td>16</td>
<td>25</td>
<td>61</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
<td>15</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>13</td>
<td>GEN-PSC1</td>
<td>32</td>
<td>34</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>14</td>
<td>GEN- Quadratic</td>
<td>39</td>
<td>40</td>
<td>62</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>19</td>
<td>24</td>
<td>49</td>
<td>42</td>
</tr>
<tr>
<td>15</td>
<td>Digonal8</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>GEN- penal1</td>
<td>19</td>
<td>25</td>
<td>139</td>
<td>527</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>11</td>
<td>52</td>
<td>180</td>
<td>180</td>
</tr>
<tr>
<td>17</td>
<td>GEN-TRI</td>
<td>37</td>
<td>101</td>
<td>78</td>
<td>222</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>45</td>
<td>76</td>
<td>87</td>
<td>89</td>
</tr>
<tr>
<td>General TOTAL of first 17 functions</td>
<td>752</td>
<td>918</td>
<td>2780</td>
<td>2438</td>
<td>2601</td>
</tr>
</tbody>
</table>
A Modified Super-linear QN-Algorithm

Table (2): Comparison between the New algorithm and MBFGS algorithms using different value of $12 < N < 4320$ for 2nd test function.

<table>
<thead>
<tr>
<th>N. OF Test</th>
<th>TEST FUNCTION</th>
<th>MBFGS</th>
<th>NOI(NO)</th>
<th>NEW</th>
<th>NOI(NO)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>36</td>
<td>360</td>
<td>1080</td>
</tr>
<tr>
<td>1</td>
<td>EX-freudenst</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Shanno</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Liarwhd</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>EX-BD2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>WX-Powell</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Engval</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Cosin</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Biggs</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>GEN-Cubic</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>EX-Himmebil</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>EX-Host</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>EX-three exponential</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>---</td>
<td>---------------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34</td>
<td>34</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>dquadratic</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>14</td>
<td>Perturbed</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>Raydan1</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>16</td>
<td>GEN-Helical</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>17</td>
<td>EX-Fred</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>18</td>
<td>GEN-Non digonal</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>19</td>
<td>Maratos</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>20</td>
<td>Full Hessian</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>21</td>
<td>Sincos</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>22</td>
<td>GQ2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>23</td>
<td>Raydon2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Table(3): Percentage performance of the new algorithm against MBFGS algorithm for 100% in both NOI and NOF.

<table>
<thead>
<tr>
<th>N</th>
<th>Costs</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>NOI</td>
<td>60.11</td>
</tr>
<tr>
<td></td>
<td>NOF</td>
<td>30.23</td>
</tr>
<tr>
<td>36</td>
<td>NOI</td>
<td>67.87</td>
</tr>
<tr>
<td></td>
<td>NOF</td>
<td>41.54</td>
</tr>
<tr>
<td>360</td>
<td>NOI</td>
<td>87.88</td>
</tr>
<tr>
<td></td>
<td>NOF</td>
<td>78.46</td>
</tr>
<tr>
<td>1080</td>
<td>NOI</td>
<td>85.81</td>
</tr>
<tr>
<td></td>
<td>NOF</td>
<td>71.24</td>
</tr>
<tr>
<td>4320</td>
<td>NOI</td>
<td>86.31</td>
</tr>
<tr>
<td></td>
<td>NOF</td>
<td>66.47</td>
</tr>
</tbody>
</table>

5. Conclusions:

In this Paper, a new modified QN-algorithm for solving a self-scaling algorithm for solving large-scale unconstrained optimization problems is proposed. The new algorithm is a self-scaling QN-algorithm. The basic idea is based on a new QN-update proved to have super-linear convergence property. Our numerical results supports our claim and also indicate that the new algorithm may be competitive with the MBFGS algorithm in most cases of test function.
Appendix1:

All the test functions used in Table (1) for this paper are from general literature:

1. Generalized Shallow Function:

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

\[ x_0 = [-2,-2,\ldots,-2,-2] . \]

2. Generalized Beale Function:

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

\[ x_0 = [-1,-1,\ldots,-1,-1] . \]

3. Arwhead Function (CUTE):

\[ f(x) = \sum_{i=1}^{n-1} (-4x_i + 3) + \sum_{i=1}^{n-1} (x_i^2 + x_n^2) , \]

\[ x_0 = [1,1,\ldots,1,1] . \]

4. Generalized Edger Function:

\[ f(x) = \sum_{i=1}^{n/2} (x_{2i-1} - 2)^4 + (x_{2i-1} - 2)^2 x_{2i}^2 + (x_{2i} + 1)^2 , \]

\[ x_0 = [1,0,\ldots,1,0] . \]

5. Diagonal4Function:

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

\[ x_0 = [1,1,\ldots,1,1] , \ c = 100 . \]
6. Extended Denschnb Function (CUTE):

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

\[ x_0 = [0.1,0.1,...,0.1,0.1] . \]

7. Extended Diagonal BDI Function:

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

\[ x_0 = [0.1,0.1,...,0.1,0.1] . \]

8. Diagonal5 Function:

\[ f(x) = \sum_{i=1}^{n} \log(\exp(x_i) + \exp(-x_i)) , \]

\[ x_0 = [1.1,1.1,...,1.1,1.1] . \]

9. Generalized Strait Function:

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

\[ x_0 = [-2,...,-2.] . \]

10. Diagonal 6 Function:

\[ f(x) = \sum_{i=1}^{n} (\exp(x_i) - (1 + x_i)) , \]

\[ x_0 = [1.,1...,1.,1.] . \]
11. Diagonal 7 Function:

\[ f(x) = \sum_{i=1}^{n} (\exp(x_i) - 2x_i - x_i^2), \quad x_0 = [1.1, \ldots, 1.1]. \]

12. Extended Denschnf Function (CUTE):

\[ f(x) = \frac{\mu^2}{2} \left( \sum_{i=1}^{n/2} \left( 2(x_{2i-1} + x_{2i})^2 + (x_{2i-1} - x_{2i})^2 - 8 \right)^2 + \left( 5x_{2i-1} + (x_{2i} - 3)^2 - 9 \right)^2 \right), \]

\[ x_0 = [2.0, 2.0, \ldots, 2.0]. \]

13. Generalized pscl Function:

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

\[ x_0 = [3.0, 1.0, \ldots, 3.0, 1.0]. \]

14. Generalized quartic Function GQ1

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

\[ x_0 = [1.1, \ldots, 1.1]. \]

15. Diagonal 8 Function:

\[ f(x) = \sum_{i=1}^{n} x_i \exp(x_i) - 2x_i - x_i^2, \]

\[ x_0 = [1.1, \ldots, 1.1]. \]
16. Generalized Penal Function:

\[ f(x) = \sum_{i=1}^{n} (x_i - 1)^2 + \text{eps}(x_i^2 - 0.25)^2, \]

\[ x_0 = [1., 2., ..., n], \quad \text{eps} = 1.0 \times 10^{-5}. \]

17. Generalized Tridiagonal Function:

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

\[ x_0 = [2., 2., ..., 2., 2]. \]

**Appendix 2:**

All the test functions used in Table (2) for this paper are from general literature:

1. Extended Freudenstein & Roth Function:

\[ f(x) = \sum_{i=1}^{n/2} \left( -13 + x_{2i-1} + ((5 - x_{2i})x_{2i} - 2) x_{2i} \right)^2 + \left( -29 + x_{2i-1} + ((x_{2i} + 1) - 14) x_{2i} \right)^2, \]

\[ x_0 = [0.5, -2, 0.5, -2, ..., 0.5, -2]. \]

2. Nondia (Shanno-78) Function (Cute):

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

\[ x_0 = [-1, -1, ..., -1, -1]. \]
3. Liarwhd Function (cute):

\[ f(x) = \sum_{i=1}^{n} 4(-x_i + x_i^2)^2 + \sum_{i=1}^{n} (x_i - 1)^2, \]

\[ x_0 = [4., 4., ..., 4.]. \]

4. Extended Block-Diagonal BD2 Function:

\[ f(x) = \sum_{i=1}^{\alpha/2} (x_{2i-1}^2 + x_{2i}^2 - 2.)^2 + (\exp(x_{2i-1} - 1.) + x_{2i}^3 - 2.)^2, \]

\[ x_0 = [1.5, 2., ..., 1.5, 2.]. \]

5. Extended Powell Function:

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

\[ x_0 = [3, -1, 0, 1, ..., 3, -1, 0, 1]. \]

6. Engval1 Function (CUTE):

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

\[ x_0 = [2., 2., ..., 2.]. \]

7. Cosine Function (CUTE):

\[ f(x) = \sum_{i=1}^{\alpha/2} \cos(-0.5x_{i+1} + x_i^2), \]

\[ x_0 = [1., 1., ..., 1., 1.]. \]
8. Biggsb1 Function (CUTE):

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

\[ x_0 = [1.,1.,...,1.,1.]. \]

9. Generalized Cubic function:

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

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

10. Extended Himmelblau Function:

\[ f(x) = \sum_{i=1}^{n/2} \left( x_{2i-1}^2 + x_{2i} - 11 \right)^2 + \left( x_{2i-1} + x_{2i}^2 - 7 \right)^2 , \]

\[ x_0 = [1.1,1.1,...,1.1,1.1]. \]

11. Extended White & Holst Function:

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

\[ x_0 = [-1.2,1,...,-1.2,1], \quad c = 100. \]

12. Extended Three Exponential Terms Function:

\[ f(x) = \sum_{i=1}^{n/2} \left( \exp(x_{2i-1} + 3x_{2i} - 0.1) + \exp(x_{2i-1} - 3x_{2i} - 0.1) + \exp(-x_{2i-1} - 0.1) \right) , \]

\[ x_0 = [0.1,0.1,0.1,...,0.1,0.1]. \]
13. Quadratic Function (CUTE):

\[ f(x) = \sum_{i=1}^{n} (x_i^2 + cx_i^2 + dx_{i+1}^2), \]

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

14. Perturbed Penalty Function:

\[ f(x) = \sum_{i=1}^{n} ix_i^2 + \frac{1}{100} \left( \sum_{j=1}^{n} x_j \right)^2, \]

\[ x_0 = [0.5,0.5,...,0.5,0.5]. \]

15. Raydan 1 Function:

\[ f(x) = \sum_{i=1}^{n} \frac{1}{100} (\exp(x_j) - x_j), \]

\[ x_0 = [1.,1.,...,1.,1]. \]

16. General Helical Function:

\[ f(x) = \sum_{i=1}^{n/3} (100x_{3i} - 10 \ast H_i)^2 + 100(R_i - 1)^2 + x_{3i}^2, \]

where

\[ R_i = \sqrt{x_{3i-2}^2 + x_{3i-1}^2}, \]

\[ H_i = \begin{cases} \arctan \frac{x_{3i-1}}{x_{3i-2}}, & \text{if } x_{3i-2} > 0 \\ 0.5 + \arctan \frac{x_{3i-1}}{x_{3i-2}}, & \text{if } x_{3i-2} < 0 \end{cases} \]

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

17. Extended Fred Function:

\[ f(x) = \sum_{i=1}^{n/2} \left( -13 + x_{2i-1} + (5 - x_{2i}) + (x_{2i} - 2)(x_{2i}) \right)^2 + \sum_{j=1}^{n/2} \left( -29 + x_{2j-1} + (1 - x_{2j}) + (x_{2j} - 14)(x_{2j}) \right)^2, \]

\[ x_0 = [1.,2.,...,n] \]
18. Generalized Non diagonal function:

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

\[ x_0 = [-1, \ldots, -1]. \]

19. Extended Martos Function:

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

\[ x_0 = [1.1, 0.1, \ldots, 1.1, 0.1]. \]

20. Full Hessian Function:

\[ f(x) = \left( \sum_{i=1}^{n} x_i \right)^2 + \sum_{i=1}^{n} (x_i \exp(x_i) - 2x_i - x_i^2), \]

\[ x_0 = [1, 1, \ldots, 1, 1]. \]

21. SINCOS Function:

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

\[ x_0 = [3, 0.1, \ldots, 3, 0.1]. \]

22. Generalized Quartic Function GQ2:

\[ f(x) = (x_i^2 - 1)^2 + \sum_{i=2}^{n} (x_i^2 - x_{i-1} - 2)^2, \quad x_0 = [1, 1, \ldots, 1, 1]. \]

23. Raydan 2 Function:

\[ f(x) = \sum_{i=1}^{n} (\exp(x_i) - x_i), \quad x_0 = [1, 1, \ldots, 1, 1]. \]
References:


