Weighted Frobenius Norm Diagonal Quasi-Newton Method For Solving Large-Scale System of Nonlinear Equations

Muhammed Yusuf Waziri and Amar Elgaili Mahmoud

1Department of Mathematical Sciences, Faculty of Physical Sciences, Bayero University Kano, Kano, Nigeria, E-mail: mywaziri.mth@buk.edu.ng
2Department of Mathematics and Computer, Faculty of Science, University of Dalanj, Dalanj, South of Kordofan, Sudan E-mail: amarelgeili@gmail.com

Abstract. One of the major shortcomings of the quasi-Newton methods and some of theirs variants in solving systems of nonlinear equations is computing and storing the approximation of the Jacobian matrix or its inverse at each iteration. This paper, presents a new method based on weighted Frobenius norm, with choice of the weighting matrix to be the identity matrix. The proposed method are matrix and derivative free. The basic idea is to incorporate Leong et al., [13] update via diagonal updating by least change secant update strategy. This can be achieved by using weighted Frobenius norm. The local convergence analysis of the scheme is presented. Numerical results carried out on some benchmark test problems show that the proposed method is promising compared to some existing conjugate gradient methods.

Keywords: System of nonlinear equations; Quasi-Newton method; Diagonal update; Least change secant strategy; Weighted Frobenius norm.

1. Introduction
A system of nonlinear equations has the following form:

\[ \begin{align*}
  f_1(x_1, x_2, \ldots, x_n) &= 0 \\
  f_2(x_1, x_2, \ldots, x_n) &= 0 \\
  &\vdots \\
  f_n(x_1, x_2, \ldots, x_n) &= 0
\end{align*} \]  

(1)

The above system can be written in the following form:

\[ F(x) = 0, \]

(2)

where \( F : C \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n \), \( x = (x_1, x_2, \ldots, x_n)^T \) and \( F = (f_1, f_2, \ldots, f_n)^T \). Each function \( f_i (i = 1, 2, \ldots, n) \) can be thought of as mapping a vector \( x = (x_1, x_2, \ldots, x_n)^T \) of the n-dimensional space \( \mathbb{R}^n \) into the real line \( \mathbb{R} \) and is assumed to satisfy the following assumptions:
1. F is continuously differentiable function define on a convex open set C.
2. There exists a locally unique root $x^*$ in $C$.
3. The Jacobian $F'$ is Lipschitz continuous near $x^*$.
4. The Jacobian $F'$ is symmetric, where the symmetry means that the Jacobian $J(x) := F'(x)$ is symmetric; that is, $J(x) = J(x)^T$.
5. $F'(x^*)$ is nonsingular.

There are many ways for finding the solution of equation (2), Newton’s method well known as the best method and some of its modifications \[1,2,9\] for getting the solution. These methods are attractive because of their locally fast convergent rates. However, the method generates an iterative sequence of points $\{x_k\}$ with

$$x_{k+1} = x_k - (F'(x_k))^{-1}F(x_k),$$

where $k = 0, 1, 2, \cdots$ and $F'(x_k)$ is the Jacobian matrix at $x_k$.

The method converges quadratically from any given initial point $x_0$ in the neighborhood of $x^*$, when the Jacobian matrix is nonsingular at a solution of equation (2) \[10,14\], that is

$$\|x_{k+1} - x^*\| \leq \lambda \|x_k - x^*\|^2,$$

for some $\lambda > 0$.

Newton’s method and some of its modification need to solve $n$ linear system of equations at each iteration using the direct Jacobian matrix or an approximation of it, This fact makes them typically unattractive for large-scale nonlinear systems of equations. This fact motivated researches to develop quasi-Newton methods, which is defined as generalization of (2) given by

$$x_{k+1} = x_k - (B(x_k))^{-1}F(x_k),$$

In quasi-Newton methods, the matrices $B_k$ are replacing the approximations of $J(x_k)$. Many quasi-Newton methods, computation of (5) do not require computing derivative at all. Moreover, in many methods, $B_k^{-1}$ is obtained from $B_k^{-1}$ by using simple procedures thanks to which the linear algebra cost involved in (5) is much less than the one involved in $J(x_k)s_k = -F(x_k)$. Various approximations to the Jacobian matrices such as the Broydens method \[3,4\] and Newton-like methods \[11,16\] are proposed. The most vital part of those algorithms are forming and storing a full-matrix approximation to the Jacobian of the given problem, which might be a really overpriced task for giant scale problems. This leads the idea of this paper, where we propose an approximation for Jacobian within the type of a diagonal matrix via least-change secant change strategy, commonly employed within the development of quasi-Newton methods.

The remainder of this paper is organized as follows. In Section 2, we present the fundamentals of our approach. Section 3 is devoted to the analysis of convergence. In Section 4, we report numerical results and comparisons of our method with some widely used algorithms and finally, we conclude in Section 5 with a discussion of future direction for our approach.

2. Main results

In this section, we present a new class of diagonal quasi-Newton method as a result of incorporating the work of Leong et al., \[13\] with weighted Frobenius norm.

2.1. Derivation of Weighted Frobenius Norm Diagonally Quasi-Newton Method (WDQN1)

Consider Leong et al., \[13\] update is defined by

$$D_{k+1} = D_k + \Omega_k,$$

$$\Omega_k = \frac{s_k^T y_k - s_k^T D_k s_k}{tr(A_k^2)} A_k,$$
where \( A_k = \text{diag}(s_k) = (s_k^1, s_k^2, \ldots, s_k^n)^2 \), \( D_{k+1} \simeq J_{k+1} \).

In 2016 Enshaei et al., [12] defined the weighted Frobenius norm by

\[
\|A\|_W = (\text{tr}(WAW^T))^{1/2}
\]

where \( W \) is a positive definite weighted matrix, the target is to approximate the Jacobian matrix diagonally (matrix free) heavily inspired by least change secant update strategy of Dennis and Wolkowicz, [7] with a matrices further restricted to be diagonal. Suppose that \( D_k \) is a bounded diagonal matrix and \( D_{k+1} \) the update version of \( D_k \) as interpretation of the result of the least change second update theory is define as below:

\[
D_{k+1} = D_k + \nabla_k,
\]

where \( \nabla_k \) is the difference between \( D_k \) and \( D_{k+1} \) is also a diagonal matrix. Let us restrict the weighting matrix \( W \) to be positive definite diagonal matrix so that the \( \|\cdot\|_W \) is well defined. Any member of quasi-Newton family, i.e \( D_{k+1} \) should satisfy the weak secant equation of Dennis and Wolkowicz, [7]

\[
s_k^T D_{k+1} s_k = s_k^T y_k,
\]

is satisfied so that guaranteed the Jacobian information along the step is correct. The derivation process of the proposed updating scheme will take the following steps:

Consider the following problem:

\[
\min \frac{1}{2} \|D_{k+1} - D_k\|_W^2 \quad \text{s.t.} \quad s_k^T D_{k+1} s_k = s_k^T y_k
\]

where \( \|\cdot\|_W \) denotes the weighted Frobenius norm. Problem (11) can be stated alternatively as:

\[
\min \frac{1}{2} \sum_{i=1}^{n} (\nabla_k^{(i)} s_k^{(i)})^2, \quad \text{s.t.} \quad \sum_{i=1}^{n} \nabla_k^{(i)} (s_k^{(i)})^2 = s_k^T y_k - s_k^T D_k s_k,
\]

where, \( \nabla_k^{(i)}; 1 \leq i \leq n \) are the diagonal elements of \( \nabla_k \) and \( s_k^{(i)} \); are the components of \( s_k \). Since the objective function on (13) is convex and subset of a convex is also convex, then (13) has a unique solution. It is Lagrangian function given by

\[
L(\nabla_k, \mu) = \frac{1}{2} \text{tr}(W^{(i)} \nabla_k^{(i)} W^{(i)} \nabla_k^{(i)T}) + \mu \left( \sum_{i=1}^{n} (\nabla_k^{(i)} s_k^{(i)})^2 + s_k^T D_k s_k - s_k^T y_k \right)
\]

where, \( \mu \) is the Lagrange multiplier associated with the constraint. Differentiating (14) with respect to each \( \nabla_k^{(i)} \) and setting the results to zero. Note that since \( \nabla_k \) is a diagonal matrix, it gives

\[
\frac{\partial}{\partial \nabla_k^{(i)}} [\text{tr}(W^{(i)} \nabla_k^{(i)})] = W^{(i)}.
\]

Hence, we have

\[
\frac{\partial L}{\partial \nabla_k^{(i)}} = W^{(i)} \nabla_k^{(i)} W^{(i)} + \mu (s_k^{(i)})^2 = 0,
\]

where, \( s_k^{(i)} \) being the \( i^{th} \) component of the vector \( s_k \) and subsequently \( \nabla_k \) becomes

\[
\nabla_k^{(i)} = -\mu W^{-1} (s_k)^2 W^{-1}(i), \quad i = 1, 2, \ldots, n
\]
Next, multiplying each of (17) by \((s_k^{(i)})^2\) and sum them all to obtain
\[
\sum_{i=1}^{n}(\nabla_k^{(i)}(s_k^{(i)})^2) = -\mu \sum_{i=1}^{n}(s_k^{(i)})^2(W^{-1}(s_k)^4W^{-1})^{(i)}.
\] (18)

Rewrite this below equation as it invokes the constraint, we have
\[
\mu = -\frac{s_k^T y_k - s_k^T D_k s_k}{\sum_{i=1}^{n}(s_k^{(i)})^2(W^{-1}(s_k)^4W^{-1})^{(i)}}, \, i = 1, 2, \ldots, n.
\] (19)

Finally, by substituting (19) into (17) gives
\[
\nabla_k^{(i)} = \frac{s_k^T y_k - s_k^T D_k s_k}{\sum_{i=1}^{n}(s_k^{(i)})^2(W^{-1}(s_k)^4W^{-1})^{(i)}}(W^{-1}(s_k)^4W^{-1})^{(i)}, \, i = 1, 2, \ldots, n.
\] (20)

Letting \(E_k = diag((s_k^{(1)})^2, (s_k^{(2)})^2, \ldots, (s_k^{(n)})^2)\), the proposed updating formula can be stated as follow:
\[
D_{k+1} = D_k + \frac{s_k^T y_k - s_k^T D_k s_k}{\sum_{i=1}^{n}(s_k^{(i)})^2(W^{-1}E_k^2W^{-1})^{(i)}}(W^{-1}E_k^2W^{-1}), \, i = 1, 2, \ldots, n.
\] (21)

where, \(W\) define as the following:
\[
W = I_n
\] (22)

\((I_n\) is identity matrix)

Now, the following steps state the algorithm of proposed methods as:

2.2. Algorithm of Weighted Frobenius Norm Diagonally Quasi-Newton Method (WDQN1) with \(W = I_n\)

Where, \((I_n\) is identity matrix)

**Step 1**: Choose an initial point \(x_0 \in \mathbb{R}^n\), a positive definite matrix \(D_0 = I_n\), and a tolerance \(\epsilon\). Set \(k := 0\).

**Step 2**: Compute \(F(x_k)\) if \(\|F(x_k)\| \leq \epsilon\), stop. Otherwise continue with Step 3.

**Step 3**: Compute \(d_k = -D_k^{-1}F(x_k)\), update \(D_k\) using formula (21), (where the index \(k\) is replaced by index \(k-1\)). If \(D_k^{(i)} = 0\), set \(D_k^{(i)} = D_{k-1}^{(i)}\) and go to step 2 and \(W\) has the choose of (22).

**Step 4**: Compute the set \(x_{k+1} = x_k + d_k\).

**Step 5**: Consider \(k = k + 1\) and go to step 2.

2.3. Convergence Analysis

Let us present the following result

Note that \(\|\nabla_k\|_W\) is bounded below if \(\|\nabla_k\|_W\) does and conversely, if \(\|\nabla_k\|_W\) is bounded above, then so does \(\|\nabla_k\|_W\). Hence we can state the following result on the boundedness of \(\|\nabla_k\|_W\) by assuming that, without loss of generality, the updating matrix (11) or (13) (Secant equation) and \(s_k^T y_k = s_k^T F'(\xi)s_k\) is always used:

**Lemma.** Assume that \(\|s_k\| \neq 0\) for all finite \(k\) and \(F\) satisfies Assumptions 1 - 2 and 5. Let \(D_k\) be the sequence generated by (11) or (13) (Secant equation) and \(s_k^T y_k = s_k^T F'(\xi)s_k\). If the given
nonsingular $D_0$ satisfies
\[
\beta \leq D_0^{(i)} \leq \gamma, \quad i = 1, 2, \ldots, n, \tag{23}
\]
for some constants $\beta$ and $\gamma$, then the sequence $\|\nabla_k\|_W$ is bounded for all finite $k$.

**Proof:** Since $\|\cdot\|_W \geq 0$ and the nonsingularity of the Jacobian matrix implies that $\|\nabla_k\|_W \neq 0$, then $\|\nabla_k\|_W$ must be bounded away from zero. Hence it is sufficient to prove that $\|\nabla_k\|_W$ is bounded above.

For $k=0$ by $s_k^T y_k = s_k^T F'(\xi) s_k$ we have
\[
|\nabla_0^{(i)}| = |\sum_{i=1}^{n} (s_0^{(i)})(W^{-1}(s_0)W^{-1}(s_0)W^{-1}(s_0))^{(i)}(s_0^{(i)}) |
\]
so that
\[
|\nabla_0^{(i)}| \leq \frac{|s_k^T F'(\xi) s_k - s_0^T D_0 s_0|}{|s_0^T F'(\xi) s_0 - s_0^T D_0 s_0|} (W^{-1}(s_0)W^{-1}(s_0))^{(i)} (s_0^{(i)})
\]
where, $(W^{-1}(s_0)W^{-1}(s_0))^{(i)}$ is the component of $s_0$ with large magnitude and for some $\xi_0$ between $x_0$ and $x_i$. Multiplying $(W^{-1}(s_0)W^{-1}(s_0))^{(i)}$ by both numerator and denominator of (24) yields
\[
|\nabla_0^{(i)}| \leq \frac{|s_k^T F'(\xi_0) s_0 - s_0^T D_0 s_0|}{(W^{-1}(s_0)W^{-1}(s_0))^{(i)} (s_0^{(i)})} (W^{-1}(s_0)W^{-1}(s_0))^{(i)} (s_0^{(i)})
\]
where, $(W^{-1}(s_0)W^{-1}(s_0))^{(i)}$ is bounded for all finite $k$.

Denote $M = \max \{M_1, |M_2| \}$ and $\Gamma = \max \{\beta, |\gamma| \}$, it follows from assumption $A_5$ and (23) that $|\nabla_0^{(i)}| \leq \frac{n(M + \Gamma)(s_0^{(i)})^2}{(W^{-1}(s_0)W^{-1}(s_0))^{(i)} (s_0^{(i)})} = c,$ \tag{26}
where, $c = n(M + \Gamma)$ is a constant. This implies that
\[
\beta_1 \leq D_0^{(i)} \leq \gamma_1, \quad \text{for all } i,
\]
where, $\beta_1 = \beta - c$ and $\gamma_1 = \gamma + c$. Thus by induction, we can conclude the proof.

To give the convergence of the proposed method, we need the following result which is the special case of more general result given by Broden et al., [5]:

**Theorem 2.** Let $x_k$ be the sequence generated by (21) . Suppose $F$ satisfies Assumptions 1 – 4 , and for all $k$,
\[
\|B_{k+1} - F'(x^*)\|_W \leq \|B_k - F'(x^*)\|_W + \alpha_k
\]
where $B_{k+1}$ is the updated version of $B_k$, $\alpha$ is some constant and $\sigma_k = \max \{\|x_{k+1} - x^*\|, \|x_k - x^*\|\}$. Then if there exists positive constants $\epsilon$ and $\delta$ such that $\|x_0 - x^*\| < \epsilon$ and $\|B_0 - F'(x^*)\|_W < \delta$, the sequence $x_k$ is well defined and converges linearly to $x^*$. We can now state the main result.

**Theorem 3.** Suppose $\|s_k\| \neq 0$ for some finite $k$ and $F$ satisfies Assumptions 1 – 4 and there exist positive constants $\epsilon$ and $\delta$ such that $\|x_0 - x^*\| < \epsilon$ and $\|B_0 - F'(x^*)\|_W < \delta$. Then the sequence $x_k$ generated by Algorithms WDQN1, is well defined and converges linearly to $x^*$.

**Proof:** We shall prove that the updating scheme (21) satisfies (27), by induction. Consider
\[
\|D_1 - F'(x^*)\|_W = \|D_0 + \nabla_0 - F'(x^*)\|_W
\]
and yields after some arrangements

\[ \|D_0 - F'(x^*)\|_W + \frac{\sum_{i=1}^{n} (s_0^{(i)}) (W^{-1}E_0^2W^{-1})^i (s_0^{(i)})}{\sum_{i=1}^{n} |s_D^T D_{is} - s_D^T D_{so}|} (W^1 E_k^2 W^{-1}) \|_W \]

\[ = \|D_0 - F'(x^*)\|_W + \frac{\sum_{i=1}^{n} (s_0^{(i)}) (W^{-1}E_0^2W^{-1})^i (s_0^{(i)})}{\sum_{i=1}^{n} (s_0^{(i)}) (W^{-1}E_0^2W^{-1})^i (s_0^{(i)})} \]

\[ \leq \|D_0 - F'(x^*)\|_W + \frac{\|D_1 - D_0\|_W \|s_0\|^2}{\sum_{i=1}^{n} (s_0^{(i)}) (W^{-1}E_0^2W^{-1})^i (s_0^{(i)})^{1/2}} \]

(29)

Since \( \|s_0\| \neq 0 \), then there exists positive constant \( \epsilon_1 \) such that \( \|s_0\| \geq \epsilon_1 \). This implies that

\[ n(s_0^M)^3 \geq \|s_0\|^3 \geq \epsilon_1, \]

which gives \( (s_0)^3 \geq \epsilon_1^3/n \). Thus, we have

\[ (s_k^M)(W^{-1}E_0^2W^{-1})^i (s_0^{(i)}) = \sum_{i=1}^{n} (s_0^{(i)})^6 \]

\[ \geq (s_0^M)^6 = \frac{1}{2}n(s_0^M)^3(s_0^M)^3 \geq \frac{1}{3} \epsilon_1^3 \|s_0\|^3 \]

and yields after some arrangements

\[ \frac{\|s_0\|}{\sum_{i=1}^{n} (s_0^{(i)}) (W^{-1}E_0^2W^{-1})^i (s_0^{(i)})^{1/2}} \leq \frac{n}{\epsilon_1}. \]

(31)

Using (26) in lemma 1 and (31), inequality (29) becomes

\[ \|D_1 - F'(x^*)\|_W \leq \|D_0 - F'(x^*)\|_W + \frac{n^{3/2} \epsilon_1}{\epsilon_1} \|s_0\|. \]

(32)

Since \( \|s_0\| = \|x_1 - x^* + x^* - x_0\| \leq \|x_1 - x^*\| + \|x^* - x_0\| \leq 2\sigma_0 \), we obtain, finally

\[ \|D_1 - F'(x^*)\|_W \leq \|D_0 - F'(x^*)\|_W + \alpha \sigma_0, \]

(33)

where \( \alpha = n^{3/2} \epsilon_1 \). Therefore from the theorem 2, the sequence generated by Algorithm WDQN1, is well defined and converges linearly to \( x^* \).

3. Numerical results

In this section, the performance of first weighted Frobenius norm diagonal quasi-Newton method (WDQN1) introduced in algorithm (2.2) will be compared with some existing Algorithms for solving (2). Some members of conjugate gradient methods, namely: the method proposed by Fletcher and Reeves (FR-CG), [8], Polak andRibiere (PRP-CG), [15] and Dai and Yuan (DY-CG), [6] in order to check their efficiency of the research using some Benchmark test problems.

The codes written in MATLAB 7.10 R2010a and run on a personal computer of 2.30 GHz CPU processor with Intel COREi5 and 4 GB RAM memory. The work used 10 test functions with dimension of three categories. The first one between 10 to 2,000, the second one is 5,000 to 50,000 and the third one 50,000 to 1,000,000 to test the methods preference in terms of the number of iterations (NI) and central processing unit time (CPU) in seconds. The condition of stopping criteria will be hold if the total number of iterations exceeds 1000 or \( ||F_k|| \leq 10^{-8} \), this symbol "-" will present the failure "if any".
3.1. Test Functions

This section will present the benchmark problems in form of normal mathematical expression used to test the proposed method.

**Problem 1. Generalised function of Rosenbrock.**
For \( i = 1, 2, \cdots, n \),
\[
F_i(x) = e^{x_i} - 1.
\]
Initial guess \( x_0 = (0.5, 0.5, \cdots, 0.5)^T \).

**Problem 2. Generalised function of Rosenbrock.**
For \( i = 1, 2, \cdots, n \),
\[
F_i(x) = x_i^2 - 4.
\]
Initial guess \( x_0 = (1.5, 1.5, \cdots, 1.5)^T \).

**Problem 3. Raydan Function (n is a multiple of 3).**
For \( i = 1, 2, \cdots, (n/3) \),
\[
F_{3i-2} = x_{3i-2}^2 x_{3i-1} - (x_{3i})^2 - 1,
F_{3i-1} = x_{3i-2}^2 x_{3i-1} x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2,
F_{3i} = e^{x_{3i}^2} - e^{x_{3i-1}^2}.
\]
Initial guess \( x_0 = (1.5, 1.5, \cdots, 1.5)^T \).

**Problem 4. Generalised function of Rosenbrock.**
For \( i = 1, 2, \cdots, n \),
\[
F_i(x) = x_i^2 - 1.
\]
Initial guess \( x_0 = (0.5, 0.5, \cdots, 0.5)^T \).

**Problem 5. Generalised function of Rosenbrock.**
For \( i = 1, 2, \cdots, n \),
\[
F_i(x) = cos x_{i-1} + x_i - 1.
\]
Initial guess \( x_0 = (0.5, 0.5, \cdots, 0.5)^T \).
4. Tables
This section will present the Table used to test the proposed method.

Table 1. Numerical result for (WDQN1) defined in Algorithm (2.2), (FR-CG), (PRP-CG) and (DY-CG) for large-scale, when solving problem 1 to 5.

| Problem | Dimension | WDQN1 | FR-CG | PRP-CG | DY-CG |
|---------|-----------|-------|-------|--------|-------|
|         | n         | NI    | CPU   | NI    | CPU   | NI    | CPU   | NI    | CPU   |
| 1       | 50,000    | 6     | 1.9844| 16    | 0.78125| 10    | 0.203125| 19    | 0.2500 |
|         | 100,000   | 6     | 3.2813| 16    | 0.296875| 10    | 0.359325| 20    | 0.5000 |
|         | 500,000   | 6     | 14.7188| 16    | 1.484375| 10    | 1.53125| 20    | 2.76875|
|         | 1,000,000 | 6     | 28.8594| 16    | 3.109375| 10    | 2.859375| 20    | 5.671875|
| 2       | 50,000    | 7     | 4.9375| 7     | 4.015625| 7     | 3.21875| 41    | 0.40625|
|         | 100,000   | 7     | 11.2031| 7     | 16.375  | 7     | 11.40625| 41    | 1.03125|
|         | 500,000   | 7     | 66.9063| 7     | 4.476525| 7     | 2.895625| 43    | 5.46875|
|         | 1,000,000 | 7     | 140.6563| -     | -      | -     | -      | 43    | 9.71875|
| 3       | 50,000    | 7     | 5.9531| 16    | 0.328125| 12    | 0.1875| 21    | 0.546875|
|         | 100,000   | 7     | 11     | 16    | 0.375  | 12    | 0.4375| 21    | 0.65625|
|         | 500,000   | 7     | 65.6875| 17    | 1.859375| 12    | 421875| 22    | 4.140625|
|         | 1,000,000 | 7     | 136.3594| 17    | 4.140625| 12    | 3.765625| 22    | 8.390625|
| 4       | 50,000    | 6     | 9.0313| 11    | 1.65625| 10    | 1.921875| 25    | 0.65625|
|         | 100,000   | 6     | 10.5313| 11    | 6.078125| 10    | 6.3750| 25    | 1.1250 |
|         | 500,000   | 7     | 80.5469| -     | -      | -     | -      | 27    | 6.57825|
|         | 1,000,000 | 7     | 160.0313| -     | -      | -     | -      | 27    | 10.015625|
| 5       | 50,000    | 9     | 11.5469| 4     | 1.4533125| 21    | 1.71875| 12    | 1.78125|
|         | 100,000   | 9     | 15.9688| 4     | 5.390625| 21    | 6.39062| 12    | 5.953125|
|         | 500,000   | 9     | 84.4844| -     | -      | -     | -      | -     | -     |
|         | 1,000,000 | 9     | 167.1406| -     | -      | -     | -      | -     | -     |

From Table 1 the proposed method of weighted diagonal quasi-Newton (WDQN1) defined in algorithm (2.2) is compared with three classes of conjugate gradient methods. It was found that from problem 1, the method is convergence to the solution is different from that of the three methods. However, according to the stopping criteria, we use as $10^{-8}$, the WDQN1 method converges to its own solution with few number of iterations than others by increasing the dimension from 50,000 to 1,000,000.

In problem 2, the method of WDQN1 converges to the solution from dimension 50,000. While FR-CG and PRP-CG are not converged to the solution fast. However, DY-CG has slow converged to solution from dimension 50,000 to 1,000,000 less CPU time than DY-CG. Hence, the method of WDQN1 is more effective in problem 2.

In problem 3 the proposed method converges to the solution from dimension 50,000 as in the problem 2. While in FR-CG, the method converges to its solution from dimension 50,000 up to 1,000,000 but the method is not consistent in getting the solution. However, in PRP-CG the solution is not applicable at all and in DY-CG the method is not approaching the same solution and it is not consistent in getting the solution. Also the method has less CPU time and few number of iterations. Hence, the conclusion is that the method is much better than others.

Finally, in problem 4 the WDQN1 method converges to the solution from dimension 50,000, while FR-CG converges to the solution but slowly. On the other hand, PRP-CG and DY-CG are not consistent in their convergence to the solution. The proposed method has less number of iterations...
and CPU time consumption. Hence, it is more effective compared to others. Generally, the proposed method has less number of iterations and it converges to the solution faster than the other three methods. Similarly, it has less CPU time consumption in most cases.

5. Conclusion
In this paper we presented a new method of the quasi-Newton classes via least change secant update strategy and the weighted Frobenius norm for solving large-scale system of nonlinear equations and compare its performance with three classes of conjugate gradient methods namely: (FR-CG), (PRP-CG) and (DY-CG) by doing some numerical experiments. We however proved the local linearly convergence of the proposed method, and the numerical results show that the method is very efficient.

6. References
[1] Bellavia, S and Morini, B 2001 A Globally Convergent Newton-Gmres Subspace Method For Systems Of Nonlinear Equations. SIAM Journal of sciences computer. 23: 940-960.
[2] Brown, P and Saad, Y 1994 Convergence Theory Of Nonlinear Newton's Krylov Algorithms. SIAM Journal of optimization. 4: 297-330.
[3] Broyden, C G 1965 A Class Of Methods For Solving Nonlinear Simultaneous Equations, Mathematics of computation. 19: 577-593.
[4] Broyden, C G 1967 Quasi-Newton Methods and Their Applications to Function Minimization, Mathematics and computer. 21: 368-381.
[5] Broyden, C G, Dennis, J E and Moré, J J 1973 On The Local And Superlinear Convergence Of Quasi-Newton Methods. Journal institute of mathematics application, 12: 233-246.
[6] Dai, Y H and Yuan, Y X 1999 A Nonlinear Conjugate Gradient Method With A Strong Global Convergence Property. SIAM J. Optim. 10: 177-182.
[7] Dennis, J D and Wolkowicz, H 1993 Least Change Secant Methods , SIAM Journal of numerical analysis. 30: 1291-1313.
[8] Fletcher, R and Reeves, C M 1964 Function Minimization By Conjugate Gradients, Computer Journal. 7: 149-154.
[9] Gasparo, M 2000 A Nomonotone Hybrid Method For Nonlinear Systems. Optimization methods software, 13: 79-94.
[10] Hafiz, M A and Bahgat, S M 2012 An Efficient Two-Step Iterative Method For Solving System Of Nonlinear Equations. Journal of mathematics research, 4: 28-34.
[11] Hassan, M A, Leong, W J and Farid, M 2009 A New Gradient Method Via Quasi-Cauchy Relation Which Guarantees Descent, Journal of computer and applied mathematics. 230: 300-305.
[12] Enshaei, S, Leong, W J and Farid, M 2011 Diagonal Quasi Newton Method Via Varitional Principle Under Generalized Forbenius Norm. Optimization method and software, ISSN:1055-6788(print)1029-4937.
[13] Leong, W J, Hassan, M A and Waziri, M Y 2011 A Matrix Free Quasi-Newton Method For Solving Large-Scale Nonlinear System, Computer and mathematics with applications. 62: 2354-2363.
[14] Mart and nez, JMI1984A Quasi-Newton Method With Modification Of One Column Per Iteration.Computing,33 : 353 – 362.
[15] Polak, E and Ribić re, G 1969 Note Sur La Convergence de Méthodes de Directions Conjuguées, Rev. Francaise Informat. Recherche Opé rationnelle. 3: 35-43.
[16] Thukral, R 2008 Introduction To A Newton-Type Method For Solving Nonlinear Equations, Appl. Math. Comput. 195: 663-668.