Derivative Free Conjugate Gradient Method via Broyden’s Update for solving symmetric systems of nonlinear equations

M.K. Dauda1,2, Mustafa Mamat1, Fatma Susilawati Mohamad1, Abubakar S. Magaji1 and M.Y. Waziri2

1Department of Mathematical Sciences, Kaduna State University, Kaduna, Nigeria.
2Department of Informatics and Computing, Universiti Sultan Zainal Abidin Gong Badak, Terengganu, Malaysia
3Department of Mathematical Science, Bayero University, Kano, Nigeria.

Abstract. The applications of mathematics in many areas of computing, scientific and engineering research mostly give rise to a systems of nonlinear equations. Various iterative methods have been developed to solve such equations, this includes Newton method, Quasi-Newton’s etc. Over the years, there has been significant theoretical study on quasi-Newton methods for solving such systems, but unfortunately the methods suffers setback. To overcome such problems, a Derivative free Method for Solving Symmetric Systems of Nonlinear Equations Using Broyden’s Update is presented. The modification is achieved by simply approximating the inverse Hessian matrix $Q_{k+1}^{-1}$ to $\theta_k I$ with ($\theta_k$ and $I$ represents acceleration parameter and an identity matrix respectively) without computing any derivative. The method uses the symmetric structure of the system sufficiently and the generalized classical Broyden’s update method for unconstrained optimization problems. The squared norm merit function is used, both the direction and the line search technique are derivative-free, this attractive feature of the proposed method makes it to have a very low storage requirement thereby solving large scale problems successfully. In an effort to solve nonlinear problems of the form $F(x) = 0, x \in \mathbb{R}^n$, different initial starting points were used on a set of benchmark test problems, the output is based on number of iterations and CPU time. A comparison between the proposed method and the classical methods were made and found that the proposed method is efficient, robust and outperformed the existing method.

1. Introduction
Consider the system of nonlinear equations

$$g(x) = 0, x \in \mathbb{R}^n$$

(1)

where $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is continuously differentiable mapping. Supposed that for each $x \in \mathbb{R}^n$, the Jacobian $g'(x)$ of $g$ at $x$ is symmetric [9]. The prominent method for finding the solution of (1) is the classical Newton’s method which generates a sequence of iterates $\{x_k\}$ from a given initial point $x_0$ via

$$x_{k+1} = x_k - (g'(x_k))^{-1}g(x_k), k = 0, 1, 2, \cdots$$

where $g'(x_k)$ is the Jacobian matrix of $g$ at $x_k$. Newton’s method converges rapidly, but its main disadvantage is that it requires computation of the first order derivatives of the systems [11]. Practically, computations of derivatives of some functions are quite difficult and/or sometimes not available [5]. In such cases, alternative to Newton’s methods is required, this is the main aim of this article. The CG methods for solving nonlinear systems of equations generates an iterative points $\{x_k\}$ from initial given point $x_0$ via.
\[ x_{k+1} = x_k + \alpha_k d_k \]

where \( \alpha_k > 0 \) is attained via line search and direction \( d_k \) are obtained using

\[ d_k = \begin{cases} -g(x_k) & \text{if } k = 0 \\ (-g(x_{k-1}) + \beta_k d_{k-1}) & \text{if } k \geq 1 \end{cases} \]

where \( \beta_k \) is term as conjugate gradient parameter and \( d_k \) is assumed to be a decent one. Different conjugate gradients methods correspond to different choices for the scalar \( \beta_k \). Substantial effort have been made by numerous researchers in order to eliminate the famous shortcomings of Newton’s method. (see [6],[8]). Most of these modifications of Newton’s method still have some shortfalls. To tackle such problems, a derivative free CG algorithm for solving large scale systems of nonlinear equations is proposed, this is carried out by a modification of the inverse Broyden’s formula and approximate it with \( \theta_k I \) at every step. The method possessed restarting and global convergence properties, thus gives it low memory requirements. Some numerical results are reported in Section 3. While Section 4 gives the conclusion. Finally, future work are made in the Section.

2. A Derivative free Broyden’s Method

This section presents a Derivative free Method for solving large-scale systems of nonlinear equations via Broyden’s update. Broyden’s formula is an iterative method that generates a sequence of points \( \{x_k\} \) from a given initial point \( x_0 \) via the following

\[ x_{k+1} = x_k - \alpha_k (B_k)^{-1} g(x_k), \quad k = 0,1,2,\ldots \]

where \( B_k \) is an approximation to the Jacobian, updated at each iteration for \( k = 0,1,2,\ldots \) [3, 7]. The updated matrix \( B_{k+1} \) is chosen in such a way that it satisfies the secant equation

\[ B_{k+1} s_k = y_k, \quad \text{with } s_k = x_{k+1} - x_k \quad \text{and} \quad y_k = g(x_{k+1}) - g(x_k), \]

the updated formula for the Broyden matrix \( B_k \) is given as [7]:

\[ B_{k+1} = B_k + \frac{(y_k - \theta_k s_k) s_k^T}{s_k s_k^T} \]

The inverse of Broyden’s update is given as [7]

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

(2)

Using the relation \((B_k)^{-1} \approx \theta_k I\) transforms equation (2) to

\[ B_{k+1}^{-1} = \theta_k I + \frac{(s_k - \theta_k y_k) y_k^T}{y_k^T y_k} \]

(3)

where \( \theta_k \) is the acceleration parameter and \( I \) is identity matrix. To obtain new direction, multiply both side of equation (3) by \( g(x_{k+1}) \) to obtain

\[ B_{k+1}^{-1} g(x_{k+1}) = \theta_k g(x_{k+1}) + \frac{(s_k - \theta_k y_k) y_k^T g(x_{k+1})}{y_k^T y_k} \]

(4)

In equation (4), observe that the direction \( B_{k+1}^{-1} g(x_{k+1}) \) is a decent one, thus represented as

\[ d_{k+1} = B_{k+1}^{-1} g(x_{k+1}) \]

(5)

Hence, from (4) and (5), the following is obtained

\[ d_{k+1} = -\theta_k g(x_{k+1}) - \frac{(s_k - \theta_k y_k) y_k^T g(x_{k+1})}{y_k^T y_k} \]

(6)

Thus, the update is given by

\[ x_{k+1} = x_k + \alpha_k d_k \]

where \( \alpha_k > 0 \) is line search strategy while the new direction is obtained via the following
\[ d_{k+1} = \begin{cases} 
- g(x_k) & \text{if } k = 0 \\
- \theta_k g(x_{k+1}) - \frac{(s_k - \theta_k y_k)y_k^T g(x_{k+1})}{y_k^T y_k} & \text{if } k \geq 1 
\end{cases} \quad (6) \]

\[ \theta_k = \frac{y_k^T y_k}{y_k^T s_k}, \quad \text{with } s_k = x_{k+1} - x_k \text{ and } y_k = g(x_{k+1}) - g(x_k). \]

A non-monotone and derivative free line search from the work of Li and Fukushima in [1] was used to compute step size \( \alpha_k \). The interesting idea is that it avoid the necessity of descent directions and guarantee that each iteration is well defined. Using the parameters \( \sigma_1 > 0, \sigma_2 > 0, r \in (0,1) \) as constants and \( \eta_k \) be a given positive sequence such that \( \sum \eta_k < \infty \). Consider \( \alpha_k = \max\{1, r^k\} \) that satisfy

\[ f(x_k + \alpha_k d_k) - f(x_k) \leq -\sigma_1 \| \alpha_k F_k \|^2 - \sigma_2 \| \alpha_k d_k \|^2 + \eta_k F(x_k). \quad (7) \]

Numerous researchers (see [6,9,10,12,13]) used this line search and found that it is effective, thus the line search (7) is used throughout the experiment.

### 2.1 Algorithm of the proposed Method

**Step 1:** Given \( x_0, \alpha > 0, \sigma \in (0,1) \) and \( \epsilon > 0 \) compute \( d_0 = -g_0 \), set \( k = 0 \).

**Step 2:** Compute \( g(x_k) \) and test the stopping criterion, i.e. \( \| g(x_k) \| \leq \epsilon \), if yes, then stop, otherwise continue with step 3.

**Step 3:** Compute \( \alpha_k \) by using the line search condition (7) and go to 4.

**Step 4:** Compute \( \theta_k = \frac{y_k^T y_k}{y_k^T s_k} \), with \( s_k = x_{k+1} - x_k \) and \( y_k = g(x_{k+1}) - g(x_k) \).

**Step 5:** Compute \( x_{k+1} = x_k + \alpha_k d_k \).

**Step 6:** Compute search direction using (6).

**Step 7:** Set \( k = k + 1 \) and go to step 2.

### 3. Numerical Results

The This section presents the numerical results of the entire work, the performance of the proposed method and the classical Broyden methods were reported, this is carried out by solving eight (8) benchmark problems with five (5) different dimensions ranging from 10 to 50000. Each problem with given initial stating point and new dimension is considered as one problem, so entirely a total number of 40 problems were solved.

**Problem 1:** (System of n Nonlinear Equations) [4]

\[ F(i) = e^{x_i} - 1; \quad i = 1, 2, 3, \ldots, n. \]

**Problem 2** (System of n Nonlinear Equations) [6]

\[ f_i(x) = x_i - 0.1x_i^2 + 1; \quad i = 1, 2, 3, \ldots, n. \]

**Problem 3:** (System of n Nonlinear Equations) [12]

\[ f_i(x) = 5x_i^2 - 2x_i - 3, \quad i = 1, 2, 3, \ldots, n. \]

**Problem 4:** (System of n Nonlinear Equations) [9]

\[ f_i(x) = e^{x_i^{i-1}} - \cos(1 - x_i); \quad i = 1, 2, 3, \ldots, n. \]

**Problem 5:** (System of n Nonlinear Equations) [4]

\[ f_i(x) = (x_i - 3) + x_i \left( \frac{\sin x_i}{3} - 0.66 \right) + 2; \quad i = 1, 2, 3, \ldots, n. \]

**Problem 6:** (System of Nonlinear Equations) [6]

\[ f_i(x) = e^{x_i^{i-1}} - \cos(1 - x_i^2); \]
\( i = 1, 2, 3, \ldots, n. \)

**Problem 7:** (Trigonometric system) \[6\]
\[ f_i(x) = x_i^2 - 3x_i + 1 + \cos(x_1 - x_2), \]
\( i = 1, 2, 3, \ldots, n. \)

**Problem 8:** (System of nonlinear equations) \[4\]
\[ f_i(x) = \frac{n(x_i-3)^2+0.5\cos(x_1-3) - x_i-2}{e^{(x_i-3)}+\log_{10}(x_i^2+1)}, \]
\( i = 1, 2, 3, \ldots, n. \)

Table 1 gives the numerical results of all the methods. The performance of these methods are compared in terms of number of iterations and CPU time. A given method is said to perform better if it has less number of iterations or less CPU time compared to its counterpart. The meaning of each column in the tables are stated as "Prob" stands for problem solved,"ISP" is the Initial starting point; "Dim" stands for dimension of the test problems, "Iter" the total number of iterations; "CPU" time taken for the system to solve given problem.

| Prob | ISP | Dim | Iter | CPU     | Iter | CPU     |
|------|-----|-----|------|---------|------|---------|
| 1    | (0.5,0.5,...,0.5)^T | 10  | 6    | 0.011498| 7    | 0.112327|
|      |     | 100 | 6    | 0.004903| 8    | 0.003440|
|      |     | 500 | 6    | 0.001682| 9    | 0.035500|
|      |     | 1000| 6    | 0.002240| 9    | 0.002556|
|      |     | 10000| 7   | 0.015650| 11   | 0.210720|
| 2    | (0.5,0.5,...,0.5)^T | 10  | 3    | 0.005784| 9    | 0.001255|
|      |     | 100 | 3    | 0.000986| 10   | 0.001258|
|      |     | 500 | 3    | 0.001328| 11   | 0.002285|
|      |     | 1000| 4    | 0.001961| 11   | 0.003046|
|      |     | 10000| 4   | 0.040845| 12   | 0.021302|
| 3    | (0.5,0.5,...,0.5)^T | 10  | 7    | 0.066771| 8    | 0.001157|
|      |     | 100 | 6    | 0.001528| 9    | 0.001246|
|      |     | 500 | 6    | 0.002377| 10   | 0.002396|
|      |     | 1000| 6    | 0.002979| 10   | 0.003331|
|      |     | 10000| 6  | 0.021675| 11   | 0.035021|
| 4    | (0.5,0.5,...,0.5)^T | 10  | 9    | 0.001612| 12   | 0.018082|
|      |     | 100 | 11   | 0.001980| 15   | 0.003037|
|      |     | 500 | 13   | 0.006148| 17   | 0.007032|
|      |     | 1000| 13   | 0.010146| 18   | 0.011764|
|      |     | 10000| 18  | 0.147268| 33   | 0.108855|
Table 2: The Numerical Results for DF-Broyden and Classical Broyden on problems 5 to 8.

| Prob | ISP | DFBroyden | Classical Broyden |
|------|-----|-----------|------------------|
|      |     | Dim | Iter | CPU | Iter | CPU |
| 5    | (0.5,0.5,⋯,0.5)^T | 10  | 6    | 0.000966 | 5   | 0.007178 |
|      |     | 100 | 6    | 0.001134 | 6   | 0.001573 |
|      |     | 500 | 6    | 0.001880 | 7   | 0.002371 |
|      |     | 1000| 6    | 0.001928 | 8   | 0.003532 |
| 6    | (0.5,0.5,⋯,0.5)^T | 10  | 12   | 0.013590 | 12  | 0.002554 |
|      |     | 100 | 14   | 0.002539 | 12  | 0.002001 |
|      |     | 500 | 15   | 0.003662 | 22  | 0.004055 |
|      |     | 1000| 16   | 0.003533 | 26  | 0.007380 |
|      |     | 10000| 25  | 0.079324 | 61  | 0.132048 |
| 7    | (0.5,0.5,⋯,0.5)^T | 10  | 7    | 0.001394 | 8   | 0.017004 |
|      |     | 100 | 7    | 0.001563 | 9   | 0.001763 |
|      |     | 500 | 7    | 0.002467 | 10  | 0.002939 |
|      |     | 1000| 7    | 0.003706 | 11  | 0.003809 |
|      |     | 10000| 8   | 0.029091 | 12  | 0.031924 |
| 8    | (0.5,0.5,⋯,0.5)^T | 10  | 23   | 0.014336 | 14  | 0.269722 |
|      |     | 100 | 23   | 0.007543 | 15  | 0.034672 |
|      |     | 500 | 23   | 0.025903 | 16  | 0.054612 |
|      |     | 1000| 23   | 0.040245 | 17  | 0.097769 |
|      |     | 10000| 24  | 0.271139 | 18  | 0.585438 |

Figure 1: The performance profile based on number of iterations for DF-Broyden and Classical Broyden on problems 1 to 8.

The analysis was done using the profile by Dolan and More [2]. The performance ratio is given by

\[ r_{p,s} = t_{p,s}/\min\{t_{p,s}\} \]

Then the performance profile is defined by

\[ P(\tau) = \frac{1}{n_p} \text{size}\{ p \in P : r_{p,s} \leq \tau \} \]

for all \( \tau \in R \) where \( P(\tau) \) is the probability for solver \( s \in S \) that a performance ratio \( r_{p,s} \) is within a factor \( \tau \in R \) of the best possible ratio. From Table 1, the proposed method performs effectively with the given benchmark problems. In terms of number of iterations, out of the 40 problems solved, the proposed method solved 33 successfully (82.5%) while the classical Broyden solves 07 (17.5%) respectively. In terms of CPU time, the proposed method solved 32 out of the 40 problems given, (80%) successfully whereas classical Broyden solves 08 of the problems, i.e.
(20%). This indicates the validity, reliability and accuracy of the method in terms of number of iterations and CPU time.

From Figure 1 and 2, it is obvious that the proposed method performed exceptionally good, which indicated by the upper line.

4. Conclusion
In this paper, a method of solving symmetric systems of nonlinear equations is presented. The method is based on the modification of Broyden’s update via derivative free approach, for which the descent condition is satisfied. Numerical experiments on 40 test problems of different dimension shows that the proposed method is the top performer in almost all the cases compare to some variants of classical Broyden’s algorithm. The method is valid in terms of formulation, reliable in terms of number of iterations and accurate in terms of CPU time. Another interesting feature of the proposed method is that it never fail to converge throughout the numerical experiments. Thus, a conclusion is made that the proposed method is a good alternative update for solving large scale systems of nonlinear equations.

5. Future work
Establishment of the global Convergence of the proposed Algorithm under mild assumptions could be a good work to extend this research. Moreover, an ardent researcher may extend the proposed method to non-smooth equations by adopting the smoothing technique. Applying this technique to solve real life problems is another area for Future work.

Acknowledgements.
The authors would like to thank the administration of Kaduna State University, Kaduna Nigeria for funding this research under the fundamental research grant Scheme of TETFUND 2019.

References
[1] Li D H and Fukushima M 2000 A Derivative-Free Line Search and Global Convergence of Broyden-Like Methods for Nonlinear Equations Optimization Methods and Software 13 pp 181-201
[2] Dolan E and More J 2002 Benchmarking Optimization Software with Performance Profiles Math. Program. Ser. A 91 pp 201-213
[3] Broyden C G A 1965 Class of Methods for Solving Nonlinear Simultaneous Equations Mathematics of Computation 19 pp 577-593
[4] Neculai Andrei 2013 A simple three-term conjugate algorithm for unconstrained Optimization Journal of Computational and Applied Mathematics 241 pp 19-29.
[5] Waziri M Y 2013 Two-Step Derivative-Free Diagonally Newton’s Method for Large-Scale Nonlinear Equations World Applied Sciences Journal 21 (Special Issue of Applied Math) 10 pp 86-94

[6] Dauda M K, Mustafa Mamat, Waziri M Y, Fadhila Ahmad and Fatma Susilawati Mohamad 2016 Improved Quasi-Newton Method Via PSB Update for Solving Systems of Nonlinear equations AIP Conference Proceedings (ICOQSIA 2016), 1782, 030009; doi: 10.1063/1.4966066.

[7] Nocedal J and Wright S J 2006 Numerical Optimization, 2nd ed., Springer, New York

[8] Griewank A 1986 The ‘global’ convergence of Broyden-like methods with suitable line search Journal of Australia Mathematical Society Ser. B 28 pp 75-92.

[9] Dauda M K Mustafa, Mamat M Y, Waziri Fadhila Ahmad and Fatma Susilawati Mohamad 2016 Inexact CG-Method via SR1 Update for Solving Systems of Nonlinear Equations Far East Journal of Mathematical Sciences (FJMS) 100 (11) pp 1787-1804

[10] Jinkui Liu and Shengjie Li 2015 Spectral DY Type Projection Method for Nonlinear Monotone Systems of Equations Journal of Computation of Mathematics 4 pp 341-354

[11] Zhou W and Shen D 2015 Convergence properties of an iterative method for solving symmetric non-linear equations Journal of Optimization Theory and Applications 164 (1) pp 277-289

[12] Waziri M, Leong W and Mamat M 2012 A two-step matrix-free secant method for solving large-scale systems of nonlinear equations Journal of Applied Mathematics Article ID 348654, 6 pages.

[13] Gaohang Y U 2010 Derivative-Free for Solving large-scale Nonlinear Systems of Equations Journal of Management Optimization 6 pp 149-16