



Two improved classes of Broyden's methods for solving nonlinear systems of equations

Mohammad H. Al-Towaiq*, Yousef S. Abu hour

Department of Mathematics and Statistics, Faculty of Science and Arts, Jordan University of Science and Technology, Irbid, Jordan.

Abstract

In this paper, we propose two efficient algorithms based on Broyden's methods by using the central finite difference and modification of Newton's method for solving systems of nonlinear equations. The most significant features of these algorithms are their simplicity and excellent accuracy. Some numerical examples are given to test the validity of the proposed algorithms and for comparison reasons. Superior results show the efficiency and accuracy of the proposed algorithms and a tremendous improvements in Broyden's methods. ©2017 All rights reserved.

Keywords: Nonlinear systems of equations, Newton's method, Broyden's methods, quasi Newton method, finite difference, secant equation.

2010 MSC: 65D15, 65H10, 65L12.

1. Introduction

Systems of nonlinear equations often appear in many application problems in science and engineering, and chemical problems. Exact solutions of the systems usually difficult to find. In recent years, many iterative methods have been developed to find approximate solutions of those systems [1–6, 10, 15, 17, 18]. In this paper we consider the nonlinear system of equation:

$$F(x) = 0, \quad (1.1)$$

where $F(x) = (f_1(x), \dots, f_n(x))^t$, $F : D \rightarrow \mathbb{R}^n$, D convex subset of \mathbb{R}^n , $x \in \mathbb{R}^n$, and $f_i : D \rightarrow \mathbb{R}$ is continuously differentiable in an open neighborhood $D \subseteq \mathbb{R}^n$. For any initial vector $x^{(0)}$ close to x^* where x^* is the exact solution of (1.1), Newton-Raphson method generates the sequence of vectors $\{x^{(k)}\}_{k=0}^{\infty}$ by using the following iterative scheme:

- Set an initial guess $x^{(0)}$.
- Solve the linear system $J(x^{(k)})d^{(k)} = -F(x^{(k)})$ for $d^{(k)}$.
- Compute $x^{(k+1)} = x^{(k)} + d^{(k)}$.

*Corresponding author

Email addresses: towaiq@just.edu.jo (Mohammad H. Al-Towaiq), ysabuhour14@sci.just.edu.jo (Yousef S. Abu hour)

doi:[10.22436/jmcs.017.01.02](https://doi.org/10.22436/jmcs.017.01.02)

Received 2016-04-27

Where $J(x)$ is the Jacobian matrix of $F(x)$, denoted by $J(x) = F'(x)$.

A significant weakness of Newton's method is that, for each iteration a Jacobian matrix must be computed, so this method is very expensive and has the following disadvantages:

1. need a good initial solution $x^{(0)}$ close to the solution x^* .
2. requires $n^2 + n$ function evaluation at each iteration (n^2 for Jacobian matrix and n for $F(x)$).
3. $J(x^{(k)})$ must be nonsingular for all k and $J(x^*)$ is invertible.
4. need to compute n^2 partial derivative for $J(x^{(k)})$ and $J^{-1}(x^{(k)})$ at each step.
5. to solve the linear system at each iteration require $O(n^3)$ arithmetic operation.

The advantage of this method is that $\{x^{(k)}\}_{k=0}^{\infty}$ converges quadratically to x^* and the scheme above is self-corrective when $J(x^{(k)})$ is nonsingular.

Many mathematicians [1, 2, 7, 16–18] developed the above technique to increase the convergence rate of Newton's method.

Quasi Newton methods are the methods which approximate the Jacobian matrix or its inverse with another matrix (i.e., $B_k \cong F'_k$ or $H_k \cong F'^{-1}_k$), where F'_k is the Jacobian matrix evaluated at the k^{th} iteration, B_k and H_k are easily computed. Those methods are used to overcome the disadvantages of Newton's method. A well-known Quasi Newton method proposed by Charles Broyden in 1965 [5], called Broyden's methods.

1.1. Broyden's methods

In [5] Broyden presented two types of methods to compute an approximate solution for the nonlinear systems of equations. The first type of his method gives an approximate matrix for $F'(x^{(k)})$ (i.e., $B_k \cong F'_k$). Two assumptions Broyden's set on B_k :

- (i) B_k must satisfy the secant equation

$$B_k s^{(k)} = y^{(k)},$$

where $s^{(k)} = x^{(k)} - x^{(k-1)}$ and $y^{(k)} = F(x^{(k)}) - F(x^{(k-1)})$.

- (ii) B_k must satisfy the no change condition: Any nonzero vector in \mathbb{R}^n can be expressed as a combination of $s^{(k)}$ and the orthogonal complement of $s^{(k)}$ say q , to uniquely defined the matrix B_k , we also need to specify how it acts on q . No information is available about the change in F in a direction of q , so we specify that no change can be made in this direction $(B_k - B_{k-1})q = 0$, implies that

$$B_k q = B_{k-1} q, \quad \forall q \in \mathbb{R}^n \text{ s.t. } (x^{(k)} - x^{(k-1)})^t q = 0.$$

From the two assumptions (i) and (ii) Broyden defined his method of first type by

$$x^{(k+1)} = x^{(k)} - B_k^{-1} F(x^{(k)}),$$

where $B_k^{-1} = B_{k-1}^{-1} + \frac{(s^{(k)} - B_{k-1}^{-1} y^{(k)})(s^{(k)})^t B_{k-1}^{-1}}{(s^{(k)})^t B_{k-1}^{-1} y^{(k)}}$.

The second type of Broyden's method approximate the inverse of Jacobian matrix (i.e., $H_k \approx F'^{-1}_k$). One can easily verify that the two assumptions of this type follow the assumptions of the first type, so the assumptions (i) and (ii) become:

- (i)' H_k must satisfy the secant equation

$$x^{(k+1)} - x^{(k)} = H_{k+1} (F(x^{(k+1)}) - F(x^{(k)})).$$

- (ii)' H_k must satisfy the no change condition:

$$H_k q = H_{k-1} q, \quad \forall q \in \mathbb{R}^n \text{ s.t. } (F(x^{(k)}) - F(x^{(k-1)}))^t q = 0,$$

therefore, Broyden set the following scheme for his second type

$$x^{(k+1)} = x^{(k)} - H_k F(x^{(k)}),$$

$$\text{where } H_k = H_{k-1} + \frac{(s^{(k)} - H_{k-1}s^{(k)})}{\|y^{(k)}\|^2} (y^{(k)})^t.$$

The main benefits of this approximation is reducing the number of function evaluation at each step from $n^2 + n$ to just n , require $O(n^2)$ arithmetic operation per iteration and need not compute the partial derivative of F . In 1973 Broyden-Dennis-Moré [6] proved the sequence $\{x^{(k)}\}_{i=0}^{\infty}$ converge to x^* with q -superlinear of convergence where $q = \frac{1+\sqrt{5}}{2}$ is the golden ratio.

In this paper, we improve and modify Broyden's methods to increase the accuracy and the order of convergence.

This paper is organized as follows: in Section 2, we present the proposed algorithms; Broyden's with central finite difference of types 1 and 2 (BC1 and BC2). In Section 3, we modified the algorithms of BC1 and BC2. The new algorithms we call them MBC1 and MBC2. In Section 4, we give some illustrative examples to test the validity of the methods and for comparison reasons. We conclude the paper in Section 5.

2. Broyden's classes with central finite difference (BC)

Broyden used forward finite difference to approximate the Jacobian matrix

$$\frac{\partial f_j}{\partial x_k}(x^{(i)}) \cong \frac{f_j(x^{(i)} + e_k h) - f_j(x^{(i)})}{h}.$$

In this section we use central finite difference instead of forward finite difference,

$$\frac{\partial f_j}{\partial x_k}(x^{(i)}) \cong \frac{f_j(x^{(i)} + e_k h) - f_j(x^{(i)} - e_k h)}{2h}.$$

In the next two subsections we discuss our main algorithms BC1 and BC2.

2.1. First type of BC (BC1)

In the first type of Broyden's we will use central finite difference to approximate Jacobian matrix (i.e., $\hat{B}_k \cong F'_k$), to improve the order of convergence and to avoid computing the Jacobian matrix.

Theorem 2.1 (Taylor's expansion [14]). *Let $F : D \rightarrow \mathbb{R}^n$ be p -time Fréchet differentiable in a convex set $D \subseteq \mathbb{R}^n$, then for any $x, h \in \mathbb{R}^n$ it holds that:*

$$F(x+h) = F(x) + F'(x)h + \frac{1}{2!}F''(x)h^2 + \dots + \frac{1}{(p-1)!}F^{(p-1)}(x)h^{p-1} + R_p,$$

where

$$\|R_p\| \leq \frac{1}{p!} \sup_{0 \leq t \leq 1} \|F^{(p)}(x+th)\| \|h\|^p \text{ and } h^p = (h, \dots, h).$$

Then by Taylor's expansion

$$F(x+h) = F(x) + F'(x)h + \frac{1}{2}F''(\xi_1)h^2, \quad (2.1)$$

$$F(x-h) = F(x) - F'(x)h + \frac{1}{2}F''(\xi_2)h^2. \quad (2.2)$$

Subtract (2.2) from (2.1), and by the mean value theorem of vector-valued function, we get

$$F(x+h) - F(x-h) = F'(x)2h + O(\|h\|^2). \quad (2.3)$$

Equation (2.3) indicates that the secant equation of Broyden's method replaced by

$$F(x + h) - F(x - h) = \hat{B}_k 2h,$$

from this, the conditions (i) and (ii) of Broyden's method, become:

(i)'' Central secant equation $F(x + h) - F(x - h) = \hat{B}_k 2h$.

(ii)'' No change condition $\hat{B}_k q = \hat{B}_{k-1} q$, where $h^t q = 0$.

Now we need to determine x , $x - h$ and $x + h$.

Let $x = x^{(k)}$ and $2h = x^{(k)} - x^{(k-2)} = s^{(k)}$ then

$$x + h = \frac{3}{2}x^{(k)} - \frac{1}{2}x^{(k-2)}, \text{ and } x - h = \frac{1}{2}(x^{(k)} - x^{(k-2)}),$$

$$y^{(k)} = F\left(\frac{3}{2}x^{(k)} - \frac{1}{2}x^{(k-2)}\right) - F\left(\frac{1}{2}(x^{(k)} - x^{(k-2)})\right).$$

By (i)'' and (ii)'', we get

$$\hat{B}_k = \hat{B}_{k-1} + \frac{(y^{(k)} - \hat{B}_{k-1}s^{(k)})(s^{(k)})^t}{\|s^{(k)}\|^2}, \quad k = 2, \dots, m.$$

\hat{B}_0 and \hat{B}_1 are not defined in this updating formula, we may define both as Newton's method, such that: $\hat{B}_0 = F'_0$ and $\hat{B}_1 = F'_1$ so $x^{(0)}$ and $x^{(1)}$ are the same. We can use Sherman-Morrison formula [9] to make the above updating depend on the inverse of B_k , that is

$$\hat{B}_k^{-1} = \hat{B}_{k-1}^{-1} + \frac{(s^{(k)} - \hat{B}_{k-1}^{-1}y^{(k)})s^{(k)}\hat{B}_{k-1}^{-1}}{(s^{(k)})^t\hat{B}_{k-1}^{-1}y^{(k)}}.$$

Figure 1 summarizes the algorithm of BC1.

step	process
step 1	input $x^{(0)}$
step 2	solve set $x^{(1)} = x^{(0)} - J^{-1}(x^{(0)})F(x^{(0)})$
step 3	set $x^{(2)} = x^{(1)} - J^{-1}(x^{(1)})F(x^{(1)})$
	for $k=2$
	$\hat{B}_0 = J(x^{(0)}), \hat{B}_1 = J(x^{(1)})$
step 4	$s^{(k)} = x^{(k)} - x^{(k-2)}$
step 5	$y^{(k)} = F\left(\frac{3}{2}x^{(k)} - \frac{1}{2}x^{(k-2)}\right) - F\left(\frac{1}{2}(x^{(k)} - x^{(k-2)})\right)$
step 6	$\hat{B}_k^{-1} = \hat{B}_{k-1}^{-1} + \frac{(s^{(k)} - \hat{B}_{k-1}^{-1}y^{(k)})s^{(k)}\hat{B}_{k-1}^{-1}}{(s^{(k)})^t\hat{B}_{k-1}^{-1}y^{(k)}}$
step 7	$x^{(k+1)} = x^{(k)} - \hat{B}_k^{-1}F(x^{(k)})$
	If $\ x^{(k+1)} - x^{(k)}\ _2 \leq \text{Tol}$, Stop
	otherwise $k = k + 1$ continue

Figure 1: Algorithm of BC type 1

2.2. Second type of BC (BC2)

For this type, we use the central finite difference to approximate F'_k (i.e., $\hat{H}_k \cong F'^{-1}_k$). One can easily

show that the conditions (i)'' and (ii)'' in BC1 method, become:

(i)''' Central secant equation $\hat{H}(F(x+h) - F(x-h)) = 2h$.

(ii)''' No change condition $\hat{H}_k q = \hat{H}_{k-1} q$, where $q^t y^{(k)} = 0$.

Now, we use the same procedures in BC1 to determine x , $x-h$ and $x+h$ for BC2.

By (i)''' and (ii)''', we obtain

$$\hat{H}_k = \hat{H}_{k-1} + \frac{(s^{(k)} - \hat{H}_{k-1} s^{(k)})}{\|y^{(k)}\|^2} (y^{(k)})^t.$$

In this method we use H_0 and H_1 as Newton's method, $\hat{H}_0 = F_0'^{-1}$ and $\hat{H}_1 = F_1'^{-1}$. Figure 2 summarizes the algorithm of BC2.

step	process	
step 1	input $x^{(0)}$	
step 2	solve	$x^{(1)} = x^{(0)} - J^{-1}(x^{(0)})F(x^{(0)})$
step 3		$x^{(2)} = x^{(1)} - J^{-1}(x^{(1)})F(x^{(1)})$
	for $k=2$	$\hat{H}_0 = J^{-1}(x^{(0)}), \hat{H}_1 = J^{-1}(x^{(1)})$
step 4		$s^{(k)} = x^{(k)} - x^{(k-2)}$
step 5		$y^{(k)} = F(\frac{3}{2}x^{(k)} - \frac{1}{2}x^{(k-2)}) - F(\frac{1}{2}(x^{(k)} - x^{(k-2)}))$
step 6		$\hat{H}_k = \hat{H}_{k-1} + \frac{(s^{(k)} - \hat{H}_{k-1} s^{(k)})}{\ y^{(k)}\ ^2} (y^{(k)})^t$
step 7		$x^{(k+1)} = x^{(k)} - \hat{H}_k F(x^{(k)})$
	If	$\ x^{(k+1)} - x^{(k)}\ _2 \leq \text{ToI}$, Stop
	otherwise	$k = k + 1$ continue

Figure 2: Algorithm of BC type 2

This approximation of Jacobian matrix increases the accuracy, and the improved algorithms have the same arithmetic operations and mathematical computations of Broyden's methods up to the first iteration, but with more accuracy.

3. Modified classes of BC (MBC)

In this section, we use Newton's modification method (predictor-corrector), in the modified classes of Broyden's types one and two. We follow the same procedures in [1, 2], by using the following two steps iteration formulas to obtain the approximate solution $x^{(k+1)}$.

Let

$$\hat{x}^{(k+1)} = x^{(k)} - \hat{M}_k F(x^{(k)}). \quad (3.1)$$

Set

$$x^{(k+1)} = \hat{x}^{(k+1)} - \hat{M}_k F(\hat{x}^{(k+1)}), \quad (3.2)$$

where $\hat{x}^{(k+1)}$ is the predictor of $x^{(k+1)}$. This modification requires $(2 \times n)$ -functions evaluation and free of partial derivative.

In (3.1) and (3.2) by replacing $\hat{M}_k^{-1} = \hat{B}_k$, we obtain the following modification of type one of BC class:

$$\begin{aligned} \hat{x}^{(k+1)} &= x^{(k)} - \hat{B}_k^{-1} F(x^{(k)}), \\ x^{(k+1)} &= \hat{x}^{(k+1)} - \hat{B}_k^{-1} F(\hat{x}^{(k+1)}). \end{aligned} \quad (\text{MBC1})$$

Set those equations on the algorithm of BC1, Figure 1, to obtain a new algorithm for MBC1. Figure 3 describes the algorithm of this modification.

step	process
input	$x^{(0)}$, and
step1	solve m : number of steps $x^{(1)} = x^{(0)} - J^{-1}(x^{(0)})F(x^{(0)})$ $x^{(2)} = x^{(1)} - J^{-1}(x^{(1)})F(x^{(1)})$
step2	let $\hat{B}_0 = J(x^{(0)}), \hat{B}_1 = J(x^{(1)})$
step3	for $k = 2$ $s^{(k)} = x^{(k)} - x^{(k-2)}$ $y^{(k)} = F(\frac{3}{2}x^{(k)} - \frac{1}{2}x^{(k-2)}) - F(\frac{1}{2}(x^{(k)} - x^{(k-2)}))$ $B_k^{-1} = B_{k-1}^{-1} + \frac{(s^{(k)} - B_{k-1}^{-1}y^{(k)})s^{(k)}B_{k-1}^{-1}}{(s^{(k)})^t B_{k-1}^{-1}y^{(k)}}$
step4	$\hat{x}^{(k+1)} = x^{(k)} - \hat{B}_k^{-1}F(x^{(k)})$ $x^{(k+1)} = \hat{x}^{(k+1)} - \hat{B}_k^{-1}F(\hat{x}^{(k+1)})$
step5	If $\ x^{(k+1)} - x^{(k)}\ _2 \leq \text{Tol}$, Stop output $x^{(k+1)}$ otherwise $k = k + 1$ continue

Figure 3: Algorithm of MBC1

We follow the same procedures as above by replacing $\hat{M}_k = \hat{H}_k$ in (3.1) and (3.2), we obtain the following modification of type two of BC class:

$$\begin{aligned}\hat{x}^{(k+1)} &= x^{(k)} - \hat{H}_k F(x^{(k)}), \\ x^{(k+1)} &= \hat{x}^{(k+1)} - \hat{H}_k F(\hat{x}^{(k+1)}).\end{aligned}\quad (\text{MBC2})$$

Set those equations on the algorithm of BC1, Figure 2, to obtain a new algorithm for MBC2. Figure 4 describes the algorithm of MBC2.

step	process
input	$x^{(0)}$, and
step1	solve m : number of steps $x^{(1)} = x^{(0)} - J^{-1}(x^{(0)})F(x^{(0)})$ $x^{(2)} = x^{(1)} - J^{-1}(x^{(1)})F(x^{(1)})$
step2	let $\hat{H}_0 = J(x^{(0)})^{-1}, \hat{H}_1 = J(x^{(1)})^{-1}$
step3	for $k = 2$ $s^{(k)} = x^{(k)} - x^{(k-2)}$ $y^{(k)} = F(\frac{3}{2}x^{(k)} - \frac{1}{2}x^{(k-2)}) - F(\frac{1}{2}(x^{(k)} - x^{(k-2)}))$ $\hat{H}_k = \hat{H}_{k-1} + \frac{(s^{(k)} - \hat{H}_{k-1}s^{(k)})}{\ y^{(k)}\ ^2}(y^{(k)})^t$
step4	$\hat{x}^{(k+1)} = x^{(k)} - \hat{H}_k F(x^{(k)})$ $x^{(k+1)} = \hat{x}^{(k+1)} - \hat{H}_k F(\hat{x}^{(k+1)})$
step5	If $\ x^{(k+1)} - x^{(k)}\ _2 \leq \text{Tol}$, Stop output $x^{(k+1)}$ otherwise $k = k + 1$ continue

Figure 4: Algorithm of MBC2

4. Performance evaluation and comparisons

In this section, we give some numerical examples to test the validity of the proposed methods and for comparison reasons.

Example 4.1. Solve $F(x) = 0$, where

$$F(x_1, x_2, x_3) = \begin{pmatrix} x_1^2 + x_2^3 + \sqrt{x_3} - 4 \\ (x_1 + x_2)^2 + \frac{(x_2 + x_3)^2}{25} - 5 \\ (x_1 - x_2)^3 + (x_2 - x_3)^2 - 9 \end{pmatrix} \text{ and } x^{(0)} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Table 1 shows the absolute error $E_k = \|x^{(k)} - x^{(k-1)}\|_2$ at the k -th iteration of our proposed algorithms and Broyden's methods.

We found that BC1 needs 15 iterations to obtain the exact solution, but Broyden's method of type 1 needs 23 iterations.

Table 1: E_k of Example 4.1

method	E_{12}	E_{13}	E_{14}
Broyden's type 1	$3.816873153 \times 10^{-4}$	$9.90750798 \times 10^{-5}$	2.1060461×10^{-5}
Broyden's type 2	1.9715239×10^{-3}	$8.36695487 \times 10^{-4}$	$2.493016279 \times 10^{-4}$
BC1	$6.08447253 \times 10^{-9}$	$1.364686448 \times 10^{-12}$	$4.625781237 \times 10^{-15}$
BC2	$1.259123925 \times 10^{-8}$	$2.868652604 \times 10^{-11}$	$3.948540613 \times 10^{-14}$
MBC1	0	—	—
MBC2	0	0	0

Example 4.2. (Broyden Tridiagonal Function [11, 13]) Solve $H(x) = 0$, where

$$H(x_1, x_2, x_3) = \begin{pmatrix} 1 - x_1 (2x_1 - 3) - 2x_2 \\ 1 - 2x_3 - x_2 (2x_2 - 3) - x_1 \\ 1 - x_3 (2x_3 - 3) - x_2 \end{pmatrix}.$$

Set $x^{(0)} = (-1, -1, -1)^t$.

Table 2 shows the absolute error $E_k = \|x^{(k)} - x^{(k-1)}\|_2$ at the k -th iteration of our proposed algorithms and Broyden's methods.

Table 2: E_k error of Example 4.2

method	E_6	E_7	E_8
Broyden's type 1	5.2766045×10^{-5}	$4.947571122 \times 10^{-6}$	$8.66544107 \times 10^{-7}$
Broyden's type 2	$4.608586696 \times 10^{-5}$	$5.413832406 \times 10^{-6}$	$1.148577881 \times 10^{-6}$
BC 1	$3.159504661 \times 10^{-8}$	$1.4459026456 \times 10^{-10}$	$6.89381544 \times 10^{-13}$
BC2	5.009867×10^{-8}	$2.09119773 \times 10^{-10}$	$8.4725911266 \times 10^{-13}$
MBC1	5.551115×10^{-17}	0	0
MBC2	1.241267×10^{-16}	0	—

Example 4.3. Solve

$$X'(t) = \frac{2t}{3X^2(t)}, \quad X(0) = 1.$$

Solution: integrate the initial value problem above from 0 to s to get

$$X(s) = 1 + \frac{2}{3} \int_0^s \frac{t}{X^2(t)} dt. \quad (\text{nonlinear integral Volterra equation}).$$

We use Trapezoidal method with 10 nodes to approximate the integral term. Let $s_i = t_i = 0.1 \times i$ where $i = 1, \dots, 10$, the corresponding nonlinear equation is

$$X_k = 1 + \frac{2}{3} \left(\frac{0.1}{2} \left(\frac{0.1 \times k}{X_k^2} \right) + \sum_{i=1}^{k-1} \frac{0.1 \times i}{X_i^2} \right), \quad k = 1, \dots, 10.$$

Now, we need to solve $\xi = G(\xi)$,

$$\text{where } \xi = (X_1, \dots, X_{10}) \text{ and } G_j(\xi) = 1 + \frac{2}{3} \left(\frac{0.1}{2} \left(\frac{0.1 \times j}{X_j^2} \right) + \sum_{i=1}^{j-1} \frac{0.1 \times i}{X_i^2} \right).$$

Let $\xi^{(0)} = (1, \dots, 1)^t$. Table 3 shows the absolute error $E_k = \|\xi^{(k)} - \xi^{(k-1)}\|_2$ at the k -th iteration of our proposed algorithms and Broyden's methods.

Table 3: E_k error of Example 4.3

method	E_4	E_5	E_6
Broyden's type 1	1.5×10^{-5}	5.31×10^{-7}	3.92×10^{-9}
Broyden's type 2	1.3×10^{-5}	4.8×10^{-7}	3.5×10^{-9}
BC 1	2.3×10^{-5}	3.02×10^{-8}	1.31×10^{-12}
BC2	2.3×10^{-5}	2.7×10^{-8}	1.01×10^{-12}
MBC1	4.4×10^{-11}	2.2×10^{-16}	0
MBC2	3.8×10^{-11}	2.2×10^{-16}	0

Example 4.4. (Chandrasekhar, $c = 1$, $n = 10$, see [8, 12]). Chandrasekhar introduced the following integral equation in the context of radiative transfer problems. Find $X \in C[0, 1]$ such that

$$X(t) = 1 + \frac{c}{2} \int_0^1 \frac{tX(t)X(s)}{t+s} ds. \quad (4.1)$$

We approximate the integral in this equation by using a composite midpoint rule

$$\int_0^1 g(t) dt \approx \frac{1}{n} \sum_{i=1}^{10} g(t_i), \text{ where } t_i = \frac{1}{10} \left(i - \frac{1}{2} \right), \quad i = 1, \dots, 10. \quad (4.2)$$

The integration term in (4.1) is approximated by (4.2) to get the corresponding nonlinear system,

$$X(t_i) = 1 + \frac{c}{2n} \sum_{j=1}^{10} \frac{t_i X(t_i) X(t_j)}{t_i + t_j},$$

that can be written as

$$X_i = \left(1 - \frac{c}{2n} \sum_{j=1}^{10} \frac{t_i X(t_j)}{t_i + t_j} \right)^{-1}, \quad i = 1, \dots, 10.$$

Now, we need to solve $\xi = G(\xi)$, where $\xi = (X_1, \dots, X_{10})$ and

$$G_i(\xi) = \left(1 - \frac{c}{2n} \sum_{j=1}^{10} \frac{t_i X(t_j)}{t_i + t_j} \right)^{-1}.$$

Let $\xi^{(0)} = (1, \dots, 1)^t$.

Table 4 shows the absolute error $E_k = \|\xi^{(k)} - \xi^{(k-1)}\|_2$ at the k -th iteration of our proposed algorithms and Broyden's methods.

Table 4: E_k error of Example 4.4

method	E_{20}	E_{21}
Newton's method	2.3×10^{-6}	1.1×10^{-6}
Broyden's type 1	8.2×10^{-5}	5×10^{-5}
Broyden's type 2	7.9×10^{-5}	4.9×10^{-5}
BC 1	1.4×10^{-6}	7×10^{-7}
BC2	4.3×10^{-6}	2.1×10^{-6}
MBC1	9×10^{-8}	5.6×10^{-8}
MBC2	8.5×10^{-8}	7.1×10^{-8}

Table 5 shows the numbers of required iterations of the proposed algorithms, Broyden's methods and Newton's method to get error less than 10^{-7} .

Table 5: Number of iterations to get required tolerance, of Example 4.4

method	Number of iteration (k)	E_k
Newton's method	25	0.5×10^{-7}
Broyden's type 1	34	0.9×10^{-7}
Broyden's type 2	34	0.7×10^{-7}
BC 1	24	0.7×10^{-7}
BC2	26	0.5×10^{-7}
MBC1	20	0.9×10^{-7}
MBC2	19	0.3×10^{-7}

The numerical results in Tables 1, 2, 3, 4 and 5 of the above examples show that our proposed algorithms BC1, BC2, MBC1, and MBC2 are very comparable and competitive to Broyden's methods. Also, we observe that the errors of our proposed algorithms decrease rapidly as number of iterations increases.

5. Conclusion

We proposed two improved classes (four types) based on quasi Newton's methods called Broyden's methods with central finite difference (BC) of type one and type two and Modified BC of type one and type two (MBC1 and MBC2). We used central finite difference to approximate the Jacobian matrix. The proposed algorithms do not need to compute partial derivatives (F'_k) and the inverse or partial derivative ($F_k'^{-1}$). The given numerical examples have demonstrated the efficiency and accuracy of the proposed methods. Tables 1, 2, 3, 4 and 5 show that the proposed algorithms converge faster than Broyden's methods and more attractive than Newton's method.

References

- [1] Y. Abu-Hour, *Improved Classes of Broyden methods for solving a nonlinear systems of equations*, MSc. thesis, Department of Mathematics and Statistics, Jordan University of Science and Technology, Jordan, (2016). 1, 1, 3
- [2] M. H. Al-Towaiq, Y. S. Abu Hour, *Two Improved Methods Based on Broyden's Newton Methods for the solution of nonlinear systems of equations*, J. Teknologi, (accepted). 1, 3
- [3] K. Amini, F. Rostami, *Three-steps modified Levenberg-Marquardt method with a new line search for systems of nonlinear equations*, J. Comput. Appl. Math., **300** (2016), 30–42.
- [4] L. Berenguer, D. Tromeur-Dervout, *Developments on the Broyden procedure to solve nonlinear problems arising in CFD*, Comput. & Fluids, **88** (2013), 891–896.
- [5] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., **19** (1965), 577–593. 1, 1.1
- [6] C. G. Broyden, J. E. Dennis, J. J. Moré, *On the local and superlinear convergence of quasi-Newton methods*, J. Inst. Math. Appl., **12** (1973), 223–245. 1, 1.1

- [7] V. Candela, R. Peris, *The rate of multiplicity of the roots of nonlinear equations and its application to iterative methods*, Appl. Math. Comput., **264** (2015), 417–430. [1](#)
- [8] S. Chandrasekhar, *Radiative transfer*, Dover Publications, Inc., New York, (1960). [4.4](#)
- [9] C. Y. Deng, *A generalization of the Sherman-Morrison-Woodbury formula*, Appl. Math. Lett., **24** (2011), 1561–1564. [2.1](#)
- [10] J. E. Dennis Jr., *On some methods based on Broyden's secant approximation to the Hessian*, Numerical methods for non-linear optimization, Conf., Dundee, (1971), 19–34, Academic Press, London, (1972). [1](#)
- [11] C. T. Kelley, *Solving nonlinear equations with Newton's method*. Fundamentals of Algorithms, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, (2003). [4.2](#)
- [12] J. J. Moré, *A collection of nonlinear model problems*, Computational solution of nonlinear systems of equations, Fort Collins, CO, (1988), 723–762, Lectures in Appl. Math., Amer. Math. Soc., Providence, RI, **26** (1990), [4.4](#)
- [13] J. J. Moré, B. S. Garbow, K. E. Hillstom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, **7** (1981), 17–41. [4.2](#)
- [14] J. M. Ortega, W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York-London, (1970). [2.1](#)
- [15] K. Sayevanda, H. Jafari, *On systems of nonlinear equations: some modified iteration formulas by the homotopy perturbation method with accelerated fourth- and fifth-order convergence*, Appl. Math. Model., **40** (2016), 1467–1476. [1](#)
- [16] S. Sharifi, M. Salimi, S. Siegmund, T. Lotfi, *A new class of optimal four-point methods with convergence order 16 for solving nonlinear equations*, Math. Comput. Simulation, **119** (2016), 69–90. [1](#)
- [17] J. R. Sharma, R. Sharma, N. Kalra, *A novel family of composite Newton-Traub methods for solving systems of nonlinear equations*, Appl. Math. Comput., **269** (2015), 520–535. [1](#)
- [18] X.-Y. Xiao, H.-W. Yin, *A simple and efficient method with high order convergence for solving systems of nonlinear equations*, Comput. Math. Appl., **69** (2015), 1220–1231. [1](#), [1](#)