CONSTRUCTION OF A BROYDEN-LIKE METHOD FOR NONLINEAR SYSTEMS OF EQUATIONS

I. A. Osinuga, S. O. Yusuff

Federal University of Agriculture, P.M.B. 2240, Department of Mathematics, Abeokuta, Niheria

Corresponding Author: I. A. Osinuga, osinuga08@gmail.com

ABSTRACT: Broyden-like methods (or hybrid Broyden methods) are one of the efficient modifications of the classical Broyden method proposed to solve nonlinear systems of equations and to overcome the deficiencies of the classical Newton method. In this work, a variant of the Broyden-like method is proposed using the weighted combination of the Trapezoidal, Simpson and Midpoint quadrature rules. Hence a new hybrid Broyden method known as TSMM has been created based on these rules. The numerical tests confirm that TSMM is promising when subjected to comparison with other Broyden-like methods.
KEYWORDS: Broyden method, quadrature formula, predictor corrector, nonlinear systems, convergence, numerical examples.

1. INTRODUCTION

Over the past decades, there has been a growing need and interests for the solution of nonlinear systems. Nonlinear systems of equations may arise for example, in sciences, engineering, mathematics, medicines, robotics as well as in phenomenon such as weather and chaos. Hence, solving such equations is becoming more essential. Since there is no exact solution can be found for vast majority of practical problems involving nonlinear phenomenon, therefore numerical or iterative methods are usually employed to approximate the solutions.
Let us consider numerical schemes for solving nonlinear systems of the form:

$$\begin{aligned} &f_1(x_1,,,,,x_n)=0\\ &f_2(x_1,,,,,x_n)=0\\ &.\\ &.\\ &.\\ &f_n(x_1,,,,,x_n)=0 \end{aligned} \qquad (1)$$

The above equation can be compactly denoted by F(x) = 0; where $F=(f_1,\ldots,f_n):R^n\rightarrow R^n$ is continuously differentiable. The classical Newton method is one of the commonly used iterative schemes for the solution of equation (1) but confronted with many demerits from the need to calculate and invert the Jacobian matrix J(x) at each iteration.
Among the successful modifications proposed by researchers to reduce the computational cost of the classical Newton, is the quasi-Newton method. The most successful and simplest quasi-Newton method for solving nonlinear systems of equations is the Broyden method [MW15]. Broyden methods [Bro65] are another revised quasi-Newton method to reduce the amount of calculation at each iteration but subsequently lead to an increase in the number of iterations to be performed to converge to a solution.
Several efforts have been made recently to improve classical Newton methods and its variants. These methods devised some techniques such as Newton-like schemes [WF00, FS03, SDK10, DS11, Dha14], matrix-free secant method [WLM12], quadrature formulas [CT06, CT07, SAN08, FS04], and composition scheme [CTV12]. Authors such as [MMW13, MMW14, MW15] and the references therein proposed quadrature based Broyden-like methods to solve systems of non-linear equations. For example, [MMW14] devised a Broyden-like method using the Trapezoidal rule in order to reduce the number of iterations of the classical Broyden method. [MW15] used the weighted combination of the Midpoint and Simpson quadrature formulas to achieve the same goal as above.
The rationale behind the use of the Broyden-like methods by [MMW14, MW15] and the need to avert the shortcomings of the classical Newton methods especially on large scale problems are motivations behind this work.
The paper is organized as follows. The proposed method is described in section 2. In section 3, we develop the convergence analysis for the method and section 4 is dedicated for the numerical results obtained by applying the classical Broyden and its variants to several bench-mark systems of nonlinear equations. In the last section, we compare the results with different methods and make some conclusions.

2. DERIVATION PROCESS OF TSMM

Let x^* be a root of the nonlinear equation F(x) = 0; where F is sufficiently differentiable. Newton method originates from the Taylor's series expansion of the function (of a single variable) f(x) about the point x_1 :

$$f(x) = f(x_1) + (x - x_1)f'(x_1) + \frac{1}{2!}(x + x_1)^2 f''(x_1) + \cdots$$

where f, and its first and second derivatives, f' and f'' are calculated at x_1. For multiple variable function F; from the above equation, it is obvious that

$$F(x) = F(x_k) + \int_{x_k}^{x} F'(t)dt \quad (2)$$

Authors such as [MMW14] and [MW15] have approximated the indefinite integral in Equation (2) with different quadrature rules and obtained a nice and better refinement of the classical Newton method. In the same view, approximating the integral in Equation (2) by the weighted combination of Trapezoidal, Simpson and Midpoint quadrature rules yields:

$$x_{k+1} = x_k - 24\left[5F'(x_k) + 14F'\left(\frac{x_k + x_{k+1}}{2}\right) + 5F'(x_{k+1})\right]^{-1} F(x_k) \quad (3)$$

Replacing $F(x_k)$, $F'(x_{k+1})$) by $B(x_k)$, $B(x_{k+1})$ respectively using the same procedure as in ([CT06, CT12, HB12]).

$$x_{k+1} = x_k - 24\left[5B(x_k) + 14B\left(\frac{x_k + x_{k+1}}{2}\right) + 5B(x_{k+1})\right]^{-1} F(x_k) \quad (4)$$

which is an implicit equation because we have x_{k+1} on both sides. In order to avoid the implicit nature of this equation, we use the (k+1)th iteration of the Broyden method in the right hand side. Thus we have

$$x_{k+1} = x_k - 24\left[5B(x_k) + 14B\left(\frac{x_k + m_k}{2}\right) + 5B(m_{k+1})\right]^{-1} F(x_k) \quad (5)$$

where m_k is given as

$$m_k = x_k - B_k^{-1} F(x_k) \quad (6)$$

so we have

$$x_{k+1} = x_k - 24[5B(x_k) + 14B(z_k) + 5B(m_k)]^{-1} F(x_k) \quad (7)$$

for $z_k = \frac{x_k + m_k}{2}$. Suppose we set $B_k = [5B(x_k) + 14B(z_k) + 5B(m_k)]$, then we have

$$x_{k+1} = x_k - 24 B_k^{-1} F(x_k) \quad (8)$$

Hence, with the formulation above and selecting Predictor-Corrector of Broyden method we have the following two-step iterative scheme of the Trapezoidal-Simpson-Midpoint method for solving Equation (1). For a given x_0 using initial matrix $B_0 = I$, compute the approximates solution x_{k+1} by the iterative schemes

$$m_k = x_k - B_k^{-1} F(x_k)$$
$$x_{k+1} = x_k - 24[5B(x_k) + 14B(z_k) + 5B(m_k)]^{-1} F(x_k) \quad (9)$$

for $z_k = \frac{x_k + m_k}{2}, k = 0,1, \ldots$

Algorithm for TSMM

1. Given initial guess x_0, let k = 0 and $B_0 = I$.
2. Compute $F(x_k)$, if $F(x_k) \leq 10^{-12}$ is satisfied stop, Else go to step 3,
3. Compute (m_k), from Equation (6).
4. Compute $B(m_k)$ using.

$$B(m_k) = B(x_k) + \frac{(y_k - B(x_k)s_k)s_k^T}{s_k^T s_k}$$

Where

$$y_k = F(m_k) - F(x_k)$$
$$s_k = m_k - x_k$$

5. Compute $B(z_k)$ using

$$B(z_k) = B(x_k) + \frac{(g_k - B(x_k)b_k)b_k^T}{b_k^T b_k}$$

where

$$z_k = \frac{m_k + x_k}{2}$$
$$g_k = F(z_k) - F(x_k)$$
$$b_k = z_k - x_k$$

6. Compute x_{k+1} from Equation (7)
7. Set k = k + 1 and go to step 2.

$$B_{k+1} = B_k + \frac{(y_k - B(x_k)s_k)s_k^T}{s_k^T s_k}$$

where

$$y_k = F(x_{k+1}) - F(x_k)$$
$$s_k = x_{k+1} - x_k$$

3. CONVERGENCE RESULT OF TRAPEZOIDAL, SIMPSON AND MID-POINT METHOD

The local convergence properties of the proposed method are presented here. We have the following standard assumptions on the nonlinear function F:

Assumptions

1. F is differentiable in an open convex set $D \in R^n$.
2. There exists $x^* \in R^n$ such that $F(x^*) = 0$ and $F(x^*)$ is nonsingular and continuous for every x.
3. $F'(x)$ is Lipschitz continuous and hence satisfies the Lipschitz condition of order 1 such that there exists a positive constant λ such that

$$\| F(x) - F(w) \| \leq \lambda \| x - w \|, \forall\ x, w \in R^n$$

Definition 3.1 (q-superlinearly convergence) [Kel95]
Let $x_k \subset R^n$ and $x^* \in R^n$. Then $x_k \to x^*$ is q-superlinearly if

$$lim_{k\to\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|} = 0$$

Lemma 3.1 [RL03]
Let $F: R^n \to R^n$ be continuously differentiable on an open convex set $D \subset R^n, x \in D$. If $F'(x)$ is Lipschitz continuous with Lipschitz constant λ, then for any $u, v \in D$ $\| F(v) - F(u) - F'(x)(v-u) \| \leq \lambda \max\{\| u - x \|, \| v - x \|$. Moreover, if $F'(x)$ is invertible, then there exists ϵ and $\rho > 0$ such that $\frac{1}{\rho} \| v - u \| \leq \| F(v) - F(u) \leq \rho \| v - u \|$ for all $u, v \in D$ for which $\max\{\| u - x \|, \| v - x \|\} \leq \epsilon$

Lemma 3.2 [RL03]
Let $x_k \in R^n, k \geq 0$. If x_k converges q-superlinearly to $x^* \in R^n$, then

$$lim_{k\to\infty} \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|} = 1$$

Herein, we present the main result which is a modified result in [MW15] to prove that the local order of convergence analysis is superlinear.

Theorem 3.3
Let $F: R^n \to R^n$ satisfy the hypothesis of Lemma 3 on the set D: Let B_k be a sequence of nonsingular matrices in $L(R^n)$ the space of real matrices of order n. Suppose for some x_0 the sequence x_k generated by Equation (8) remains in D and $lim_{k\to\infty} x_k = x^*$ where for each $x_k \neq x^*$. Then $\{x_k\}$ converges q-superlinearly to x^* and $F(x^*) = 0$ if and only if

$$lim_{k\to\infty} \frac{\|\frac{1}{24}B_k - F'(x^*)s_k\|}{\|s_k\|} = 0 \qquad (10)$$

where $s_k = x_{k+1} - x_k$ and $B_k = 5B(x_k) + 14B(z_k) + 5B(m_k)$

Proof
Suppose Equation (10) holds; then Equation (8) becomes

$$0 = \frac{1}{24}B_k s_k + F(x_k)$$
$$0 = \frac{1}{24}B_k s_k + F(x_k) - F'(x^*)s_k + F'(x^*)s_k$$
$$0 = \frac{1}{24}B_k s_k - F'(x^*)s_k + F(x_k) + F'(x^*)s_k$$
$$-F(x_{k+1}) + F(x_{k+1}) = (\frac{1}{24}B_k - F'(x^*))s_k + F(x_k) + F(x^*)s_k$$
$$-F(x_{k+1}) = \left(\frac{1}{24}B_k - F'(x^*)\right)s_k + (-F(x_{k+1}) + F(x_k) + F(x^*)s_k$$

Take the norm of both sides to have:

$$\| -F(x_{k+1}) \| = \| \left(\frac{1}{24}B_k - F'(x^*)\right)s_k + (-F(x_{k+1})) + F(x_k) + F(x^*)s_k \|$$

Using vector norm properties, we have;

$$\| -F(x_{k+1}) \| \leq \| \left(\frac{1}{24}B_k - F'(x^*)\right)s_k \| + \| (-F(x_{k+1})) + F(x_k) + F(x^*)s_k \|$$

Divide through by $\| s_k \|$ to have;

$$\frac{\| -F(x_{k+1}) \|}{\| s_k \|} \leq \frac{\| \left(\frac{1}{24}B_k - F'(x^*)\right)s_k \|}{\| s_k \|} + \frac{(-F(x_{k+1})) + F(x_k) + F(x^*)s_k \|}{\| s_k \|}$$

Using Lemma 3.1, we have;

$$\| -F(x_{k+1}) \| \leq \frac{\| \left(\frac{1}{24}B_k - F'(x^*)\right)s_k \|}{\| s_k \|} + \lambda \max\{\| x_{k+1} - x^* \|, \| x_k - x^* \|\}$$

Since $x_k \to x^* \forall\ k$, then from (10), we have

$$lim_{k\to\infty}\frac{\| F(x_{k+1}) \|}{\| s_k \|} \leq \frac{\| \left(\frac{1}{24}B_k - F'(x^*)\right)s_k \|}{\| s_k \|} + \lambda \max\{\| x_{k+1} - x^* \|, \| x_k - x^* \|\}$$

$$F(x^*) = F\left(\lim_{k\to\infty} x_k\right) = \lim_{n\to\infty} F(x_k) = 0$$

But $F'(x^*)$ is nonsingular, thus by Lemma 3.1 $\exists \rho > 0, k_0 \geq 0$ such that we have;

$$\| F(x_{k+1}) \| = \| F(x_{k+1}) - F(x^*) \| \geq \frac{1}{\rho} \| x_{k+1} - x^* \|$$

For all $k \geq k_0$, Equation (12) and Equation (13) gives

$$0 = \lim_{k\to\infty}\frac{\| F(x_{k+1}) \|}{\| s_k \|} \geq \lim_{k\to\infty}\frac{1}{\rho}\frac{\|x_{k+1}-x^*\|}{\|s_k\|} \geq \lim_{k\to\infty}\frac{1}{\rho}\frac{\|x_{k+1}-x^*\|}{\|x_{k+1}-x^*\|+\|x_{k+1}-x^*\|} = \lim_{k\to\infty}\frac{\frac{1}{\rho}t_k}{1+t_k}$$

And

$$t_k = \frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|}$$

This implies that

$$\lim_{k\to\infty} t_k = 0$$

Therefore x_k converges q-superlinearly to x^*. Conversely, suppose that x_k converges q-superlinearly to x^* and $F(x^*) = 0$. Then by Lemma 3.1, there exist a $\rho > 0$ such that we have

$$\| F(x_{k+1}) \| \leq \rho \| x_{k+1} - x^* \|$$

Then

$$0 = \lim_{k\to\infty}\frac{\| x_{k+1} - x^* \|}{\| x_k - x^* \|} \geq \lim_{k\to\infty}\frac{\| F(x_{k+1}) \|}{\rho \| x_k - x^* \|} = \lim_{k\to\infty}\frac{\| F(x_{k+1}) \|}{\rho \| s_k \|}\frac{\| s_k \|}{\| x_k - x^* \|}$$

Using lemma 3.2, we have

$$\lim_{k\to\infty}\frac{\| F(x_{k+1}) \|}{\| s_k \|} = 0$$

From equation (11), we have

$$\frac{\| \left(\frac{1}{24}B_k - F'(x^*)\right)s_k \|}{\| s_k \|} \leq \lim_{k\to\infty}\frac{\| F(x_{k+1}) \|}{\| s_k \|} + \lim_{k\to\infty}\frac{\left(-F(x_{k+1})\right) + F(x_k) + F(x^*)s_k \|}{\| s_k \|} \leq 0 + \lambda \lim_{k\to\infty} max\{\| x_k - x^* \|, \| x_{k+1} - x^* \|\}$$

Since x_k converges to x^*, then

$$\lim_{k\to\infty} \| x_k - x^* \|$$

which proves that

$$\frac{\| \left(\frac{1}{24}B_k - F'(x^*)\right)s_k \|}{\| s_k \|} = 0$$

4. NUMERICAL RESULTS AND DISCUSSION

In order to evaluate the performance of the proposed method, we apply the method to solve eight (8) benchmark problems using eight (8) dimensions ranging from 5 to 1,065 variables. A comparison of the numerical test results of our new method is made with those of the following three well-known methods:

- Classical Broyden Method
- Trapezoidal Broyden Method ([MMW14])
- Midpoint-Simpson Broyden Method ([MW15])

The comparison was done on the number of iterations and the CPU time in seconds. The computational experiments were carried out using MATLAB 2012b with a double precision arithmetic. The program is designed to terminate whenever the number of iterations reaches 500 and no x_k satisfies $F \| x_k \| \leq 10^{-12}$. A failure is reported (denoted by '-') in the tabulated result.

List of Tested Problems

Problem 1 ([CTV12])

$F_i(x) = x_i x_{i+1} - 1,$

$F_n(x) = x_n x_1 - 1.$

$i = 1,2, \dots, n-1 \; and = x_0 = (0.8, 0.8, \dots, 0.8)^T$

Problem 2 ([SDK10])

$F_i(x) = x_i x_{i+1} - 1,$

$F_n(x) = x_n x_1 - 1.$

$i = 1,2, \dots, n-1 \; and = x_0 = (0.5, 0.5, \dots, 0.5)^T$

Problem 3 ([SDK10])

$F_i(x) = x_i x_{i+1} - 1,$

$F_n(x) = x_n x_1 - 1.$
$i = 1,2, \dots, n-1 \; and = x_0 = (2, 2, \dots, 2)^T$
Problem 4 ([HB12])
$F_i(x) = {x_i}^2 - \cos{(x_1 - 1)},$
$i = 1,2, \dots, n \; and = x_0 = (2, 2, \dots, 2)^T$
Problem 5 ([SDK10])
$F_i(x) = {x_i}^2 - 1,$
$i = 1,2, \dots, n \; and = x_0 = (0.5, 0.5, \dots, 0.5)^T$
Problem 6 ([SDK10])
$F_i(x) = \exp({x_i}^2 - 1) - \cos{(1 - {x_i}^2)},$
$i = 1,2, \dots, n \; and = x_0 = (0.5, 0.5, \dots, 0.5)^T$
Problem 7 ([DS11])
$F_i(x) = \exp(x_i) - 1,$
$i = 1,2, \dots, n \; and = x_0 = (0.5, 0.5, \dots, 0.5)^T$
Problem 8 ([HB12])
$F_i(x) = \exp(x_i) - 1,$
$i = 1,2, \dots, n \; and = x_0 = (0.7, 0.7, \dots, 0.7)^T$

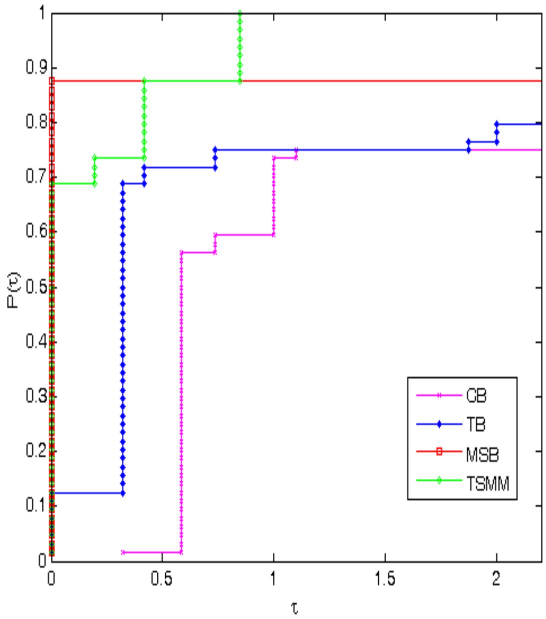

Figure 1: Performance Profile based on the number of iterations

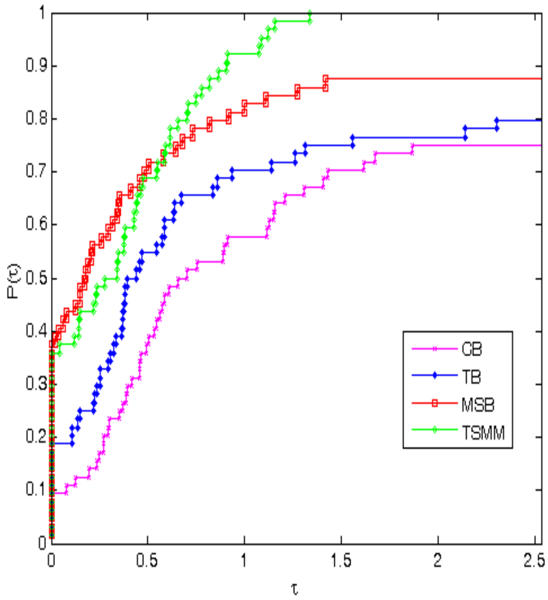

Figure 2: Performance Profile based on CPU time

Table 1: Numerical Results for all the four Methods

Prob	n	1		2		3		4	
		CB		TB		MSB		TSMM	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU
1	5	6	0.051	4	0.048	3	0.039	4	0.051
	15	6	0.066	4	0.084	3	0.062	4	0.078
	35	5	0.155	5	0.164	3	0.103	4	0.142
	65	5	0.215	5	0.389	3	0.247	4	0.236
	165	5	0.515	5	0.574	4	0.437	4	0.482
	365	5	0.515	5	0.574	4	0.437	4	0.482
	665	6	2.794	5	2.469	4	2.351	4	2.126
	1065	6	5.131	5	4.463	4	4.225	4	3.742
2	5	6	0.047	5	0.047	4	0.050	4	0.032
	15	6	0.069	5	0.101	4	0.057	4	0.083
	35	6	0.164	5	0.177	4	0.136	4	0.159
	65	6	0.257	5	0.284	4	0.183	4	0.238
	165	6	1.178	5	0.676	4	0.437	4	0.514
	365	6	2.971	5	2.797	4	3.011	4	1.126
	665	6	2.934	5	2.552	4	2.617	4	2.139
	1065	6	6.884	5	4.351	4	3.894	4	3.729
3	5	-	-	-	-	5	0.058	5	0.052
	15	-	-	-	-	5	0.063	5	0.048
	35	-	-	-	-	5	0.131	5	0.159
	65	-	-	-	-	5	0.241	5	0.392
	165	-	-	-	-	5	0.510	5	0.686
	365	-	-	-	-	5	2.598	5	1.346
	665	-	-	-	-	5	2.663	5	2.526
	1065	-	-	-	-	5	5.648	5	4.438
4	5	12	0.061	22	0.179	6	0.071	8	0.066
	15	12	0.163	24	0.385	6	0.088	8	0.143
	35	12	0.296	-	-	6	0.175	8	0.381
	65	12	1.386	24	10873	6	0.380	8	0.639
	165	12	1.703	-	-	6	0.739	8	1.129
	365	14	6.819	-	-	7	3.069	8	40157
	665	14	13.169	-	-	7	4.311	8	70584
	1065	15	16.180	-	-	7	7.489	8	11.288
5	5	6	0.053	5	0.044	4	0.078	4	0.065
	15	6	0.069	5	0.134	4	0.056	4	0.125
	35	6	0.143	5	0.155	4	0.179	4	0.144
	65	6	1.063	5	0.333	4	0.629	4	0.607
	165	6	0.895	5	0.613	4	1.019	4	1.289
	365	6	3.135	5	1.784	4	2.824	4	1.414
	665	6	4.059	5	3.419	4	3.441	4	3.107
	1065	6	7.439	5	6.685	4	6.497	4	5.139
6	5	-	-	5	0.075	-	-	9	0.098
	15	-	-	5	0.089	-	-	9	0.189
	35	-	-	5	0.280	-	-	9	0.440
	65	-	-	5	0.453	-	-	9	0.848
	165	-	-	5	0.982	-	-	9	1.474
	365	-	-	5	3.596	-	-	9	6.758
	665	-	-	5	6.012	-	-	9	9.214
	1065	-	-	5	8.088	-	-	9	13.845
7	5	6	0.042	5	0.067	4	0.035	4	0.048
	15	6	0.072	5	0.099	4	0.073	4	0.096
	35	6	0.138	5	0.162	4	0.108	4	0.137
	65	6	0.388	5	0.356	4	0.857	4	0.394
	165	6	0.767	5	0.669	4	0.519	4	0.604

Prob	n	1		2		3		4	
		CB		TB		MSB		TSMM	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU
	365	6	2.554	5	3.133	4	3.615	4	2.628
	665	6	6.492	5	3.839	4	4.833	4	3.459
	1065	6	8.312	5	6.393	4	5.578	4	5.119
8	5	6	0.130	5	0.072	4	0.049	4	0.067
	15	6	0.119	5	0.081	4	0.055	4	0.069
	35	6	0.169	5	0.192	4	0.189	4	0.148
	65	6	0.169	5	0.372	4	0.263	4	0.427
	165	6	0.778	5	0.755	4	0.600	4	0.584
	365	6	3.813	5	2.601	4	3.323	4	2.419
	665	6	5.636	5	5.090	4	4.439	4	3.704
	1065	6	7.422	5	6.163	4	5.973	4	5.240

Table 2: Summary of Robustness, Efficiency and Combined Robustness and Efficiency Measures

	CB	TB	MSB	TSMM
R	0.7500	0.7968	0.8750	1.0000
E	0.6313	0.6888	1.0000	0.9034
C_{ER}	0.4734	0.6293	0.8750	0.9034

5. DISCUSSION OF RESULTS

The numerical results in Table 1 clearly show that our new method has better results compared to Classical Broyden (CB) and the Trapezoidal Broyden method (TB) in terms of number of iterations and with regards to the CPU time, the method is competitive with the other methods. To better compare the numerical performance of the four methods, the performance profile for number of iterations is plotted. Figures 1 and 2 shows the performances of the four methods, relative to the number of iterations and CPU time as proposed by [DM02] respectively. Looking at the figure, we see that the new method, TSMM outperform all other methods since it solves 100% of the test problems. In order to have a clearer information about robustness and efficiency of the methods (in terms of number of iterations), we used the performance profile of [BP90]. In the analysis of performance of nonlinear systems solvers, this profile uses three (3) indices namely; Robustness, Efficiency and Combined Robustness and Efficiency indices. Next, we give the report of these analyses. Table 2 shows that in terms of robustness, TSMM is superior to CB, TB and MSB. TSMM has never failed to solve any of the tested benchmark problems. However, in terms of Efficiency, the TSMM is just competitive compared to MSB but performed better then CB and TB. Hence, these observations further authenticate the superiority of the proposed method to CB and TB for solving large scale systems of nonlinear equations. The performance profile of each of the methods for small value of τ, the best possible ratio are shown in Figures 1 and 2. We assume that we have n_s solvers and n_p problems, the performance profile $P : R \to [0,1]$ is defined as follows: let P and S be the set of problems and set of solvers respectively. For each problem $p \in P$ and for each solver $s \in S$ we define $t_{p;s}$ (number of iterations required to solve problem p by solver s) and $t_{p;s}$ (computing time required to solve problem p by solver s) for Figures 1 and 2 respectively. The performance ratio is given by $r_{p;s} = \frac{t_{p;s}}{\min\{t_{p;s},\ s \in S\}}$. Then the performance profile is defined by $P(\tau) := \frac{1}{n_p}\ size\ p \in P : log_2 \leq \tau, \forall \tau \in R$ where $P(\tau)$ is the probability for solver $s \in S$ that a performance ratio $\tau_{p;s}$ is within a factor $\tau \in R$ of the best possible ratio. Both figures showed that TSMM solves all of the test problems successfully. MSB has the probability of 0.87 and 0.38 in Figures 1 and 2 respectively followed by the proposed method with 0.68 and 0.36, TB with 0.13 and 0.19 and CB with 0.03 and 0.09. In Figure 1, TSMM competes with MSB for $\tau \geq 0.4$. For factor $\tau \geq 0.3$, MSB and TSMM have the best probabilities in Figure 2.

6. CONCLUSIONS

We have presented another approximation alternative to Jacobian matrices via the Trapezoidal, Simpson and Midpoint quadrature formulas and preserved the local order of convergence of the classical Broyden method. The numerical tests for the benchmark functions show that the method is robust and competitive.

REFERENCES

[Bro65] **C. G. Broyden** - *A class of methods for solving nonlinear simultaneous equations*. Math Comput; 19: 577-593, 1965.

[BP90] **I. Bogle, J. Perkins** - *A new sparsity preserving quasi-Newton update for solving nonlinear equations*. SIAM Journal on Scientific and Statistical Computing. 11(4): 621-630, 1990.

[CT06] **A. Cordero, J. R. Torregrosa** - *Variants of Newton's method for functions of several variables*. Appl Math Comput; 183: 199-208, 2006.

[CT07] **A. Cordero, J. R. Torregrosa** - *Variants of Newton's method using fifth-order quadrature formulas*. Appl. Math. Comput. 190, 686-698, 2007.

[CTV12] **A. Cordero, J. R. Torregrosa, M. P. Vassileva** - *Pseudocomposition: a technique to design predictor-corrector methods for systems of nonlinear equations*. Appl Math Comput; 218: 11496-11504, 2012.

[Dha14] **A. Dhamacharoen** - *An efficient hybrid method for solving systems of nonlinear equations*. J. Comput Appl Math; 263: 59-68, 2014.

[DM02] **E. D. Dolan, J. J. More** - *Benchmarking optimization software with performance profiles*. Math. Program Ser. A; 91: 201-213, 2002.

[DS11] **M. T. Darvishi, B. Shin** - *High-Order Newton-Krylov Methods to Solve Systems of Nonlinear Equations*. J. KSIAM, 15, 19-30, 2011.

[FS03] **M. Frontini, E. Sormani** - *Some variants of Newton's method with third order convergence and multiple roots*. J. Comput. Appl. Math. 156, 345-354, 2003.

[FS04] **M. Frontini, E. Sormani** - *Third order methods for quadrature formulae for solving system of nonlinear equations*. Appl. Math. Comput. 149, 771-782, 2004.

[HB12] **M. A. Hafiz, S. M. Bahgat** - *An efficient two-step iterative method for solving system of nonlinear equations*. Journal of Mathematical Research; 4, 28-34, 2012.

[Kel95] **C. T. Kelly** - *Iterative Methods for Linear and Nonlinear Equations*. Philadelphia, PA, USA: SIAM, 1995.

[MW15] **H. Mohammad, M. Y. Waziri** - *On Broyden-like update via some quadratures for solving nonlinear systems of equations*, Turkish Journal of Mathematics, 39: 335-345, 2015.

[MMW13] **K. Muhammad, M. Mamat, Y. M. Waziri** - *A Broyden's-like Method for Solving Systems of Nonlinear Equations*. World Applied Sciences Journal 21 (Special Issue of Applied Math): 168-173, 2013.

[MMW14] **M. Mamat, K. Muhammad, M. Y. Waziri** - *Trapezoidal Broyden Method for Solving Systems of Nonlinear Equations*. Applied Mathematical Sciences, 8, 251-260, 2014.

[SDK10] **B. C. Shin, M. T. Darvishi, C. H. Kim** - *A comparison of the Newton-Krylov method with high order Newton-like methods to solve nonlinear systems*. Applied Mathematics and Computation, vol. 217, no. 7, 3190-3198, 2010.

[SAN08] **A. R. Soheili, S. A. Ahmadian, J. Naghipoor** - *A Family of Predictor-Corrector Methods Based on Weight Combination of Quadratures for Solving Nonlinear Equations*. International journal of Nonlinear Science, 6, 29-33, 2008.

[RL03] **B. Van De Rotten, S. V. Lune** - *A Limited Memory Broyden Method to Solve High-Dimensional Systems of Nonlinear Equations, Technical* Report MI 2003-2006. Leiden, the Netherlands: University of Leiden, 2003.

[WLM12] **M. Y. Waziri, W. J. Leong, M. Mamat** - *A Two-Step Matrix-Free Secant Method for Solving Large-Scale Systems of Nonlinear Equations*. Journal of Applied Mathematics, Article ID: 348654, 2012.

[WF00] **S. Weerakoon, T. G. I. Fernando** - *A variant of Newton's method with accelerated third order convergence*. Appl. Math. Lett. 13, 87-93, 2000.