



# A Comparison of Newly Developed Broyden – Like Methods for Solving System of Nonlinear Equations

Azure Isaac<sup>1,\*</sup>, Twum Boakye Stephen<sup>2</sup>, Baba Seidu<sup>2</sup>

<sup>1</sup>Mathematics Department, University for Development Studies, Tamale, Ghana

<sup>2</sup>Mathematics Department, C. K. Tedam University of Technology and Applied Sciences, Novrongo, Ghana

## Email address:

azureike@yahoo.com (A. Isaac), stwum@cktutas.edu.gh (T. B. Stephen), bseidu@cktutas.gh (B. Seidu)

\*Corresponding author

## To cite this article:

Azure Isaac, Twum Boakye Stephen, Baba Seidu. A Comparison of Newly Developed Broyden – Like Methods for Solving System of Nonlinear Equations. *International Journal of Systems Science and Applied Mathematics*. Vol. 6, No. 3, 2021, pp. 77-94.

doi: 10.11648/j.ijssam.20210603.11

Received: July 10, 2021; Accepted: July 20, 2021; Published: July 24, 2021

---

**Abstract:** The search for a more efficient and robust numerical method for solving problems have become an interesting area for many researchers as most problems resulting into nonlinear system of equations would require a very good numerical method for its computation. The introduction of the Broyden method has served as the foundation to developing several others, which are referred to as Broyden – like methods by some authors. These methods, in most cases, have proven to be superior to the original classical Broyden method in terms of the number of iterations needed and the CPU time required to reach a solution. This research sought to develop new Broyden – like methods using weighted combinations of quadrature rules (i.e., Simpson -1/3 and Simpson -3/8 rules against Midpoint, Trapezoidal, and Simpson quadrature rules). The weighted combination of the quadrature rules in the development of the new methods led to the discovery of several new methods. Some of which have proven to be more efficient and robust when compared with some existing methods. A comparison of these newly developed methods with the classical Broyden method together with some existing improved Broyden method revealed that, one of the newly developed methods namely, Midpoint–Simpson-3/8 (MS–3/8) method, outperformed all the others, with the MS–3/8 method giving the best of numerical results in all the benchmark problems considered in the study.

**Keywords:** Broyden Method, Newton-Raphson Method, Quadrature Rules, Simpson – 1/3 Rule, Simpson - 3/8 Rule, Nonlinear Systems, Convergence

---

## 1. Introduction

Finding solutions to equations is an important quest in mathematical computations. The roots of equations provide answers to many practical problems under study. Finding the most efficient numerical method for the purpose is very critical since the accuracy of the result for most practical problems is essential [3]. A problem becomes even more demanding if it requires solving a system of nonlinear equations after modeling.

Nonlinear equations is one of the most important problems in numerical computations, especially for a diverse range of engineering applications, including applications in many scientific fields [14]. Many problems can be reduce to solving nonlinear equations, which is one of the basic

problems in mathematics [1]. Great efforts have been made by researchers, and many constructive theories and algorithms are proposed to solve systems of nonlinear equations [13]. However, there still exist some problems in solving such systems. For most traditional numerical methods such as Newton's method, the convergence and performance characteristics can be highly sensitive to the initial guess of the solution. However, it is very difficult to select a reasonable initial guess of a solution for most nonlinear equations [11]. The algorithm may fail or the results may be improper if the initial guess of the solution is unreasonable. Many different combinations of the traditional numerical methods and intelligent algorithms are applied to solve systems of nonlinear equations [15, 17], which can overcome the problem of selecting a reasonable initial guess of the solution. But the algorithms are too complicated or

expensive to calculate with when there are a number of nonlinear equations to solve simultaneously.

Among the classes of numerical schemes for solving nonlinear systems of equations, the Newton–Raphson scheme remains popular. However, the Newton–Raphson method is confronted with some drawbacks, a major one of which is the need to compute the inverse Jacobian matrix iteration by iteration. This makes it inefficient for large sized problems especially [21], which provides the motivation for the current work.

The Broyden method, which is a quasi-Newton method, has seen significant modifications and improvements and these have motivated other researchers to develop new methods capable of solving efficiently nonlinear systems of equations [1]. Many authors continue to present different techniques which are Newton-like schemes [23, 8, 7], Mixed Free Secant methods, or Quadrature formulas.

One of such approaches which was used to improve the performance of the Broyden method, introduced the central finite difference to approximate the inverse Jacobian matrix which ended up proposing two improved classes of schemes [1]. In another research work, the Steepest Decent method was presented and used to obtain good and sufficient initial guess (starting values) before solving the problem with the Broyden method [16].

A leading trend of new methods developed for the computation of solutions of systems of nonlinear equations for the past few years has been to formulate iterative schemes using the quadrature rules. Some references in relation to developed methods using quadrature rules include [18, 21] [10].

$$\int_{x_k}^x F'(t)dt = \int_{x_{k,1}}^1 \int_{x_{k,2}}^2 \cdots \int_{x_{k,n}}^n F'(x_1, x_2, \dots, x_n) dx_n dx_{n-1} \dots dx_1 \quad (4)$$

The alternative approach is to treat the multiple integral as a nested sequence of one–dimensional integrals, and to use one-dimensional quadrature rule with respect to each argument in turn (Hafiz et al, 2012). Hence we can approximate  $\int_{x_k}^x F'(t)dt$  with the weighted combination of quadrature formulas. The authors [6-8, 21, 20] and the references therein have proposed various methods for approximating the indefinite integral in equation (4) using Newton Cotes formulae of order zero to one. In a related work, a variant of the Broyden-like method was proposed using the weighted combination of the Trapezoidal, Simpson and Midpoint quadrature rules and this resulted into the Broyden-like method named TSMM, [21], given as;

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

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

where:  $z_k = \frac{x_k + m_k}{2}, k = 0, 1, \dots$

The TSMM was compared with some existing methods such as the Classical Broyden (CB), Trapezoidal-Broyden (TB) and the Midpoint-Simpson-Broyden methods (MSB) and the TSMM method outperformed all other methods [21]. Another related work by the same author was carried out the following year which resulted into a robust Broyden-like

Newton Cotes quadrature rules are a group of formulas for numerical integration based on evaluating the integrand at equally spaced points. Named after Isaac Newton and Roger Cotes [7], they fit data to local order  $k$  polynomial approximants. The Newton–Cotes quadrature formulas approximate the integral of a function  $\int_a^b f(x)dx$  by evaluating the function at  $k$  nodes  $(x_1, x_2, \dots, x_n)$  and weighting those nodes with  $n$  weights  $w_1, w_2, \dots, w_n$ . The most common of Newton-Cotes quadrature formulas are the Mid-point, Trapezoidal and Simpson's rules. The general form of the Newton–Cotes formula is;

$$\int_a^b f(x)dx = \sum_{k=1}^n w_k f(x_k) \quad (1)$$

The Newton's method can be derived from the Taylor's series expansion of a function (of a single variable)  $f(x)$  about the point  $x_1$ :

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

where  $f$ , and its first and second derivatives  $f'$  and  $f''$  are evaluated at  $x_1$ . In the case of a multiple variable function  $F: R^n \rightarrow R^n$ , (2) can be shown [20], to equivalently give:

$$F(x) = F(x_k) + \int_{x_k}^x F'(t)dt \quad (3)$$

The matrix of partial derivatives  $F'(t)$  appearing in (3) is the Jacobian  $J$ , where  $\int_{x_k}^x F'(t)dt$  is multiple integrals as in (4):

method which was named the Midpoint-Trapezoidal (MT) method. The iterative scheme of the method is given as;

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

$$x_{k+1} = x_k - 4[B(x_k) + 2B(z_k) + B(m_k)]^{-1}F(x_k) \quad (6)$$

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

The MT method was also compared with other existing methods such as the Classical Broyden (CB), Trapezoidal-Broyden and Midpoint-Simpson-Broyden (MSB) methods and it came out that the MT method performed extremely better than all the others [20]. A notable fact in all the above mentioned Broyden-like methods is the continuous usage of the three common quadrature rules (i.e. Trapezoidal, Midpoint and Simpson rules). With the improved forms of some of these common quadrature rules, it was anticipated that further improved methods could be developed.

This study approximates the integral in Equation (4) by using the weighted combination of the quadrature rules Trapezoidal, Midpoint, Simpson, Simpson's 1/3 and Simpson's 3/8 quadrature rules.

The table below shows how the the quadrature rules were combined;

**Table 1.** Combination of Quadrature Rules to Yield New Broyden-like Methods.

Quadrature rules	Trapezoidal (T)	Midpoint (M)	Simpson (S)
Simpson-1/3 (S-1/3)	TS-1/3	MS-1/3	SS-1/3
Simpson-3/8 (S-3/8)	TS-3/8	MS-3/8	SS-3/8

A related work by the authors [10] constructed a Broyden-like method called the Trapezoidal–Simpson’s 3/8 method with the weighted combination of the Trapezoidal and Simpson’s 3/8 quadrature rules. Though this method performed better in some bench mark problems than the other Broyden-like methods, there were instances where some other methods had a better result.

In this study the following objectives are achieved: (i) Development of five new Broyden – like methods using combined weights of the quadrature rules; (ii) The new methods are analysed by comparing the number of iterations and the CPU time with the existing Broyden-like methods using selected systems of nonlinear equations as test problems. In the rest of the paper, section 2.0 describes the general formula of Simpson 1/3 and 3/8 quadrature rule while section 3.0 gives details on how the newly developed methods were derived, with numerical tests and results well illustrated in section 4.0 and section 5.0 gives a summary conclusion on findings from the research.

## 2. The General Formula of Simpson 1/3 and 3/8 Quadrature Rule

Most (if not all) of the developed formulas for integration are based on a simple concept of approximating a given function  $f(x)$  by a simpler function (usually a polynomial function)  $f_i(x)$ , where  $i$  represents the order of the polynomial function. Simpson’s 1/3 rule for integration was derived by approximating the integrand  $f(x)$  with a 2<sup>nd</sup> order (quadratic) polynomial function  $f_2(x)$  [2], given by:

$$f_2(x) = a_0 + a_1x + a_2x^2 \quad (7)$$

Choosing

$$\{x_0, f(x_0)\}, \{x_1, f(x_1)\} \text{ and } \{x_2, f(x_2)\}$$

$$f(x_0) = a_0 + a_1x_0 + a_2x_0^2$$

$$f(x_1) = a_0 + a_1x_1 + a_2x_1^2$$

$$f(x_2) = a_0 + a_1x_2 + a_2x_2^2$$

$$\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{bmatrix}$$

$$[A]_{3 \times 3} a_{3 \times 1} = f_{3 \times 1} \quad (8)$$

$$a_{3 \times 1} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = [A]^{-1} f \quad (9)$$

Substituting (9) into (8), we obtain

$$f_2(x) = [1, x, x^2][A]^{-1} f$$

Furthermore

$$x_0 = a$$

$$x_1 = a + h = a + \frac{b-a}{2} = \frac{a+b}{2}$$

$$x_2 = a + 2h = a + 2\left(\frac{b-a}{2}\right) = \frac{a+b}{2}$$

Substituting the form of  $f_2(x)$  into  $I = \int_a^b f(x)dx$ , we have  
 $I \approx \int_a^b f_2(x)dx$

$$= (b-a) \frac{(f(x_0) + 4f(x_1) + f(x_2))}{6}$$

Since  $h = \frac{b-a}{2} \Rightarrow b-a = 2h$  the above equation becomes

$$I \approx \frac{2h}{6} \times \{f(x_0) + 4f(x_1) + f(x_2)\}$$

$$\int_a^b f(x)dx \approx \left(\frac{b-a}{6}\right) \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) \quad (10)$$

and this is the Simpson 1/3 quadrature rule.

In a similar fashion, Simpson 3/8 rule for integration can be derived by approximating the given function  $f(x)$  with the 3<sup>rd</sup> order (cubic) polynomial  $f_3(x)$  given as

$$f_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3 = [1, x, x^2, x^3] \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

The unknown coefficients  $a_0, a_1, a_2$ , and  $a_3$  in (9) can be obtained by substituting four known coordinate data points  $\{x_0, f(x_0)\}, \{x_1, f(x_1)\}, \{x_2, f(x_2)\}$  and  $\{x_3, f(x_3)\}$  into it as follows:

$$f(x_0) = a_0 + a_1x_0 + a_2x_0^2 + a_3x_0^3$$

$$f(x_1) = a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3$$

$$f(x_2) = a_0 + a_1x_2 + a_2x_2^2 + a_3x_2^3$$

$$f(x_3) = a_0 + a_1x_3 + a_2x_3^2 + a_3x_3^3$$

The above can be put in matrix notation as:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ f(x_3) \end{bmatrix} \quad (11)$$

Expression (11), it can symbolically be represented as

$$[A]_{4 \times 4} a_{4 \times 1} = f_{4 \times 1} \quad (12)$$

Therefore:

$$a_{4 \times 1} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = [A]^{-1} f \quad (13)$$

Substituting (13) into (12), we obtain

$$f_3(x) = [1, x, x^2, x^3][A]^{-1} f \quad (14)$$

Furthermore

$$\begin{aligned} x_0 &= a \\ x_1 &= a + h = a + \frac{b-a}{3} = \frac{2a+b}{3} \\ x_2 &= a + 2h = a + \frac{2b-2a}{3} = \frac{a+2b}{3} \\ x_3 &= a + 3h = a + \frac{3b-3a}{3} = b \end{aligned}$$

Substituting the form of  $f_3(x)$  into  $I = \int_a^b f(x)dx$ , we have

$$I \approx \int_a^b f_3(x)dx = (b-a) \times \frac{\{f(x_0)+3f(x_1)+f(x_2)\}}{8} \quad (15)$$

Since  $h = \frac{b-a}{3} \Rightarrow b-a = 3h$  and Equation (15) becomes

$$I \approx \frac{3h}{8} \times \{f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\}$$

Substituting  $h = \frac{b-a}{3}$  into the above equation and rewriting the whole equation in terms of  $a$  and  $b$ , the general formula for Simpson's 3/8 quadrature rule is given by (16)

$$\int_a^b f(x)dx \approx \left(\frac{b-a}{8}\right) \left(f(a) + 3f\left(\frac{2a+b}{3}\right) + 3f\left(\frac{a+2b}{3}\right) + f(b)\right) \quad (16)$$

$$\int_{x_k}^x F'(t)dt = \frac{x-x_k}{4} [F'(x_k) + F'(x)] + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)\right] \quad (19)$$

Substituting equation (19) into (3), we have

$$F(x) = F(x_k) + \frac{x-x_k}{4} [F'(x_k) + F'(x)] + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)\right] \quad (20)$$

Since  $F(x) = 0$ , we get

$$0 = F(x_k) + \frac{x-x_k}{4} [F'(x_k) + F'(x)] + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)\right] \quad (21)$$

Multiplying through equation (21) by  $\frac{12}{x-x_k}$ , we get

$$0 = \frac{12}{x-x_k} F(x_k) + 3[F'(x_k) + F'(x)] + \left[F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)\right] \quad (22)$$

Expanding equation (22) gives

$$0 = \frac{12}{x-x_k} F(x_k) + 3F'(x_k) + 3F'(x) + F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)$$

### 3. Derivation of the New Broyden–Like Methods

#### Derivation of TS-1/3 Method

The Newton method is derived from the Taylor's series expansion of a function (of a single variable)  $f(x)$  about the point  $x$ ;

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

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 can be shown that

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

Suppose  $x^*$  is a simple root of the nonlinear equation  $F(x) = 0$ , where  $F$  is sufficiently differentiable. Let  $F: D \subset R^n \rightarrow R^n$  be a smooth mapping that has continuous second order partial derivatives on a convex open set  $D \subset R^n$  and that has a locally unique root  $x$  in  $D$ ,  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))'$ ,  $x = (x_1, x_2, \dots, x_n)$  and  $f_k: R^n \rightarrow R$  is a nonlinear function, then we have equation (2) above.

Taking into consideration the two quadrature rules that is; Trapezoidal quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{2}\right) (F(a) + F(b)) \quad (17)$$

Simpson 1/3 quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{6}\right) \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) \quad (18)$$

Approximating the integral in equation (3) by the average of Trapezoidal and Simpson 1/3 (TS-1/3) quadrature rules yields:

$$0 = \frac{12}{x-x_k} F(x_k) + 3F'(x_k) + 3F'(x) + F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x) \quad (23)$$

$$\Rightarrow \frac{-12}{x-x_k} F(x_k) = 4F'(x_k) + 4\left(\frac{x_k+x}{2}\right) + 4F'(x)$$

$$\Rightarrow x - x_k = \frac{-12F(x_k)}{4F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + 4F'(x)}$$

$$\Rightarrow x = x_k - \frac{-12F(x_k)}{4F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + 4F'(x)}$$

$$x = x_k - 12 \left[ 4F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + 4F'(x) \right]^{-1} F(x_k) \quad (24)$$

Given  $x_k$  and setting  $x = x_{k+1}$  in equation (24), we have

$$x_{k+1} = x_k - 12 \left[ 4F'(x_k) + 4F'\left(\frac{x_k+x_{k+1}}{2}\right) + 4F'(x_{k+1}) \right]^{-1} F(x_k) \quad (25)$$

which is an implicit equation because the presence of  $x_{k+1}$  on both sides of (24). To avoid this, we use the  $(k+1)^{th}$  iteration of the Broyden's method in the right hand side. Therefore:

$$x_{k+1} = x_k - 12[4F'(x_k) + 4F'(z_k) + 4F'(m_k)]^{-1} F(x_k) \quad (26)$$

with  $m_k = x_k - B_k^{-1} F(x_k)$  and  $z_k = \frac{x_k+m_k}{2}$

Now replacing  $F'(x_k)$ ,  $F'(m_k)$  and  $F'(z_k)$  by  $B(x_k)$ ,  $B(m_k)$  and  $B(z_k)$  respectively and use the same procedure as prescribed in [4, 5, 9], we get

$$x_{k+1} = x_k - 12[4B(x_k) + 4B(z_k) + 4B(m_k)]^{-1} F(x_k) \quad (27)$$

Let  $B_k = 4B(x_k) + 4B(z_k) + 4B(m_k)$

$$\Rightarrow x_{k+1} = x_k - 12B_k^{-1} F(x_k) \quad (28)$$

Hence for a given initial solution  $x_0$  and initial matrix  $B_0 = I$  ( $I$  an identity matrix), an approximate solution  $x_{k+1}$  can be computed by the iterative schemes as in [20], where:

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

$$x_{k+1} = x_k - 12[4B(x_k) + 4B(z_k) + 4B(m_k)]^{-1} F(x_k) \quad (29)$$

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

*Derivation of MS-3/8 Method*

Consider the two quadrature rules:

Midpoint quadrature rule

$$\int_a^b F(x) \approx (b-a)F\left(\frac{a+b}{2}\right) \quad (30)$$

Simpson 3/8 quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{8}\right) \left( f(a) + 3f\left(\frac{2a+b}{3}\right) + 3f\left(\frac{a+2b}{3}\right) + f(b) \right) \quad (31)$$

Approximating the integral in Equation 3 by the average of the Midpoint and Simpson 3/8 (MS-3/8) quadrature rules yields:

$$\int_{x_k}^x F'(t) dt = \frac{x-x_k}{2} \left[ F'\left(\frac{x_k+x}{2}\right) \right] + \left(\frac{x-x_k}{16}\right) \left[ F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x) \right] \quad (32)$$

Substituting Equation 32 into Equation 3, we have

$$F(x) = F(x_k) + \frac{x-x_k}{2} \left[ F'\left(\frac{x_k+x}{2}\right) \right] + \left(\frac{x-x_k}{16}\right) \left[ F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x) \right] \quad (33)$$

Since  $F(x) = 0$ , we get

$$0 = F(x_k) + \frac{x-x_k}{2} \left[ F' \left( \frac{x_k+x}{2} \right) \right] + \left( \frac{x-x_k}{16} \right) \left[ F'(x_k) + 3F' \left( \frac{2x_k+x}{3} \right) + 3F' \left( \frac{x_k+2x}{3} \right) + F'(x) \right] \quad (34)$$

Multiplying through Equation 34 by  $\frac{16}{x-x_k}$ , we get

$$0 = \frac{16}{x-x_k} F(x_k) + 8 \left[ F' \left( \frac{x_k+x}{2} \right) \right] + \left[ F'(x_k) + 3F' \left( \frac{2x_k+x}{3} \right) + 3F' \left( \frac{x_k+2x}{3} \right) + F'(x) \right] \quad (35)$$

Expanding Equation 35 gives

$$\begin{aligned} 0 &= \frac{16}{x-x_k} F(x_k) + 8F' \left( \frac{x_k+x}{2} \right) + F'(x_k) + 3F' \left( \frac{2x_k+x}{3} \right) + 3F' \left( \frac{x_k+2x}{3} \right) + F'(x) \\ \Rightarrow \frac{-16}{x-x_k} F(x_k) &= 8F' \left( \frac{x_k+x}{2} \right) + F'(x_k) + 3F' \left( \frac{2x_k+x}{3} \right) + 3F' \left( \frac{x_k+2x}{3} \right) + F'(x) \\ \Rightarrow x-x_k &= \frac{-16F(x_k)}{8F' \left( \frac{x_k+x}{2} \right) + F'(x_k) + 3F' \left( \frac{2x_k+x}{3} \right) + 3F' \left( \frac{x_k+2x}{3} \right) + F'(x)} \\ x &= x_k - 16 \left[ 8F' \left( \frac{x_k+x}{2} \right) + F'(x_k) + 3F' \left( \frac{2x_k+x}{3} \right) + 3F' \left( \frac{x_k+2x}{3} \right) + F'(x) \right]^{-1} F(x_k) \end{aligned} \quad (36)$$

Setting  $x = x_{k+1}$  in Equation 36, we have

$$x_{k+1} = x_k - 16 \left[ 8F' \left( \frac{x_k+x_{k+1}}{2} \right) + F'(x_k) + 3F' \left( \frac{2x_k+x_{k+1}}{3} \right) + 3F' \left( \frac{x_k+2x_{k+1}}{3} \right) + F'(x_{k+1}) \right]^{-1} F(x_k) \quad (37)$$

Setting  $F' \left( \frac{2x_k+x_{k+1}}{3} \right) \approx F' \left( \frac{x_k+2x_{k+1}}{3} \right) \approx F' \left( \frac{x_k+x_{k+1}}{2} \right)$ , Equation 37 becomes

$$x_{k+1} = x_k - 16 \left[ 8F' \left( \frac{x_k+x_{k+1}}{2} \right) + F'(x_k) + 3F' \left( \frac{x_k+x_{k+1}}{2} \right) + 3F' \left( \frac{x_k+x_{k+1}}{2} \right) + F'(x_{k+1}) \right]^{-1} F(x_k) \quad (38)$$

$$x_{k+1} = x_k - 16 \left[ 8F' \left( \frac{x_k+x_{k+1}}{2} \right) + F'(x_k) + 3F' \left( \frac{2x_k+x_{k+1}}{3} \right) + 3F' \left( \frac{x_k+2x_{k+1}}{3} \right) + F'(x_{k+1}) \right]^{-1} F(x_k) \quad (39)$$

$$x_{k+1} = x_k - 16 \left[ F'(x_k) + 14F' \left( \frac{x_k+x_{k+1}}{2} \right) + F'(x_{k+1}) \right]^{-1} F(x_k) \quad (40)$$

Which is an implicit equation because the presence of  $x_{k+1}$  at both sides of the equation, hence to avoid its implicit nature we use the  $(k+1)^{th}$  iteration of the Broyden's method in the right hand side. Thus we have;

$$x_{k+1} = x_k - 16[F'(x_k) + 14F'(z_k) + F'(m_k)]^{-1} F(x_k) \quad (41)$$

with  $m_k = x_k - B_k^{-1} F(x_k)$  and  $z_k = \frac{x_k+m_k}{2}$

Now replacing  $F'(x_k)$ ,  $F'(m_k)$  and  $F'(z_k)$  by  $B(x_k)$ ,  $B(m_k)$  and  $B(z_k)$  respectively and use the same procedure as prescribed in [4, 5, 9], we get

$$x_{k+1} = x_k - 16[B(x_k) + 14B(z_k) + B(m_k)]^{-1} F(x_k)$$

Let  $B_k = B(x_k) + 14B(z_k) + B(m_k)$

$$\Rightarrow x_{k+1} = x_k - 16B_k^{-1} F(x_k) \quad (42)$$

Hence we have the following method using initial matrix  $B_0 = I$  and an initial guess  $x_0$ . For a given  $x_0$  using initial matrix  $B_0 = I$ , an approximated solution for  $x_{k+1}$  can be computed by the iterative schemes as in [20];

$$\begin{aligned} m_k &= x_k - B_k^{-1} F(x_k) \\ x_{k+1} &= x_k - 16[B(x_k) + 14B(z_k) + B(m_k)]^{-1} F(x_k) \end{aligned} \quad (43)$$

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

In a similar way as in the above derivations, four other methods have been developed and their iterative schemes are shown below;

*Derivation of MS-1/3 Method*

Midpoint quadrature rule

$$\int_a^b F(x) \approx (b-a)F\left(\frac{a+b}{2}\right)$$

Simpson 1/3 quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{6}\right) \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

Approximating the integral in equation (3) by the average of Midpoint and Simpson 1/3 (MS – 1/3) quadrature rules yields:

$$\int_{x_k}^x F'(t)dt = \frac{x-x_k}{2} F'\left(\frac{x_k+x}{2}\right) + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)\right] \quad (44)$$

Substituting equation (44) into (3), we have

$$F(x) = F(x_k) + \frac{x-x_k}{2} F'\left(\frac{x_k+x}{2}\right) + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)\right] \quad (45)$$

Since  $F(x) = 0$ , we get

$$0 = F(x_k) + \frac{x-x_k}{2} F'\left(\frac{x_k+x}{2}\right) + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)\right] \quad (46)$$

Multiplying through equation (46) by  $\frac{12}{x-x_k}$ , we get

$$0 = \frac{12}{x-x_k} F(x_k) + 6F'\left(\frac{x_k+x}{2}\right) + \left[F'(x_k) + 4F'\left(\frac{x_k+x}{2}\right) + F'(x)\right] \quad (47)$$

Expanding equation (47) gives

$$\begin{aligned} 0 &= \frac{12}{x-x_k} F(x_k) + 10F'\left(\frac{x_k+x}{2}\right) + F'(x_k) + F'(x) \\ \Rightarrow \frac{-12}{x-x_k} F(x_k) &= 10F'\left(\frac{x_k+x}{2}\right) + F'(x_k) + F'(x) \\ \Rightarrow x-x_k &= \frac{-12F(x_k)}{10F'\left(\frac{x_k+x}{2}\right) + F'(x_k) + F'(x)} \\ x &= x_k - 12 \left[ 10F'\left(\frac{x_k+x}{2}\right) + F'(x_k) + F'(x) \right]^{-1} F(x_k) \end{aligned} \quad (48)$$

Setting  $x = x_{k+1}$  and  $x_k = x_k$  in equation (48), we have

$$x_{k+1} = x_k - 12 \left[ 10F'\left(\frac{x_k+x_{k+1}}{2}\right) + F'(x_k) + F'(x_{k+1}) \right]^{-1} F(x_k) \quad (49)$$

$$x_{k+1} = x_k - 12 \left[ F'(x_k) + 10F'\left(\frac{x_k+x_{k+1}}{2}\right) + F'(x_{k+1}) \right]^{-1} F(x_k) \quad (50)$$

Which is an implicit equation because the presence of  $x_{k+1}$  at both sides of the equation, hence to avoid its implicit nature we use the  $(k+1)^{th}$  iteration of the Broyden's method in the right hand side. Thus we have;

$$x_{k+1} = x_k - 12[F'(x_k) + 10F'(z_k) + F'(m_k)]^{-1} F(x_k)$$

with  $m_k = x_k - B_k^{-1}F(x_k)$  and  $z_k = \frac{x_k+m_k}{2}$

Now replacing  $F'(x_k)$ ,  $F'(m_k)$  and  $F'(z_k)$  by  $B(x_k)$ ,  $B(m_k)$  and  $B(z_k)$  respectively and use the same procedure as prescribed in [4] [5] [9], we get

$$x_{k+1} = x_k - 12[B(x_k) + 10B(z_k) + B(m_k)]^{-1} F(x_k)$$

Let  $B_k = B(x_k) + 10B(z_k) + B(m_k)$

$$\Rightarrow x_{k+1} = x_k - 12B_k^{-1}F(x_k) \quad (51)$$

Hence we have the following method using initial matrix  $B_0 = I$  and an initial guess  $x_0$ . For a given  $x_0$  using initial matrix  $B_0 = I$ , an approximated solution for  $x_{k+1}$  can be computed by the iterative schemes as in [20];

$$m_k = x_k - B_K^{-1}F(x_k) \quad (52)$$

$$x_{k+1} = x_k - 12[B(x_k) + 10B(z_k) + B(m_k)]^{-1}F(x_k) \quad (53)$$

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

*Derivation of SS-1/3 Method*

Simpson quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{2}\right) \left(f(a) + 3f\left(\frac{a+b}{2}\right) + f(b)\right) \quad (54)$$

Simpson 1/3 quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{6}\right) \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

Approximating the integral in equation (3) by the average of Simpson and Simpson 1/3 (SS – 1/3) quadrature rules yields:

$$\int_{x_k}^x F'(t)dt = \frac{x-x_k}{4} \left(F'(x_k) + 3F'\left(\frac{x+x_k}{2}\right) + F'(x)\right) + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x+x_k}{2}\right) + F'(x)\right] \quad (55)$$

Substituting equation (55) into (3), we have

$$F(x) = F(x_k) + \frac{x-x_k}{4} \left(F'(x_k) + 3F'\left(\frac{x+x_k}{2}\right) + F'(x)\right) + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x+x_k}{2}\right) + F'(x)\right] \quad (56)$$

$F(x) = 0$ , hence

$$0 = F(x_k) + \frac{x-x_k}{4} \left(F'(x_k) + 3F'\left(\frac{x+x_k}{2}\right) + F'(x)\right) + \left(\frac{x-x_k}{12}\right) \left[F'(x_k) + 4F'\left(\frac{x+x_k}{2}\right) + F'(x)\right] \quad (57)$$

Multiplying through (57) by  $\frac{12}{x-x_k}$

$$0 = \frac{12}{x-x_k} F(x_k) + 3F'(x_k) + 9F'\left(\frac{x+x_k}{2}\right) + 3F'(x) + \left[F'(x_k) + 4F'\left(\frac{x+x_k}{2}\right) + F'(x)\right] \quad (58)$$

$$0 = \frac{12}{x-x_k} F(x_k) + 4F'(x_k) + 13F'\left(\frac{x+x_k}{2}\right) + 4F'(x) \quad (59)$$

$$0 = \frac{12}{x-x_k} F(x_k) + 4F'(x_k) + 13F'\left(\frac{x+x_k}{2}\right) + 4F'(x) \quad (60)$$

$$\frac{-12F(x_k)}{x-x_k} = 4F'(x_k) + 13F'\left(\frac{x+x_k}{2}\right) + 4F'(x) \quad (61)$$

$$x - x_k = \frac{-12F(x_k)}{4F'(x_k) + 13F'\left(\frac{x+x_k}{2}\right) + 4F'(x)} \quad (62)$$

$$x = x_k - 12 \left[4F'(x_k) + 13F'\left(\frac{x+x_k}{2}\right) + 4F'(x)\right]^{-1} F(x_k) \quad (63)$$

Setting  $x = x_{k+1}$  and  $x_k = x_k$  in equation (63), we have

$$x_{k+1} = x_k - 12 \left[4F'(x_k) + 13F'\left(\frac{x_k + x_{k+1}}{2}\right) + 4F'(x_{k+1})\right]^{-1} F(x_k) \quad (64)$$

Which is an implicit equation because the presence of  $x_{k+1}$  at both sides of the equation, hence to avoid its implicit nature we use the  $(k+1)^{th}$  iteration of the Broyden's method in the right hand side. Thus we have;

$$x_{k+1} = x_k - 12[4F'(x_k) + 13F'(z_k) + 4F'(m_k)]^{-1}F(x_k) \quad (65)$$

Let  $B_k = 4B(x_k) + 13B(z_k) + 4B(m_k)$

$$\Rightarrow x_{k+1} = x_k - 12B_k^{-1}F(x_k) \quad (66)$$

*Derivation of SS-3/8 Method*



Simpson quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{2}\right) \left(f(a) + 3f\left(\frac{a+b}{2}\right) + f(b)\right)$$

Simpson 3/8 quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{8}\right) \left(f(a) + 3f\left(\frac{2a+b}{3}\right) + 3f\left(\frac{a+2b}{3}\right) + f(b)\right)$$

Approximating the integral in equation (3) by the average of Simpson and Simpson 1/3 (SS – 1/3) quadrature rules yields:

$$\int_{x_k}^x F'(t)dt = \frac{x-x_k}{4} \left(F'(x_k) + 3F'\left(\frac{x+x_k}{2}\right) + F'(x)\right) + \left(\frac{x-x_k}{16}\right) \left[F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x)\right] \quad (67)$$

Substituting equation (67) into (3), we have

$$F(x) = \frac{x-x_k}{4} \left(F'(x_k) + 3F'\left(\frac{x+x_k}{2}\right) + F'(x)\right) + \left(\frac{x-x_k}{16}\right) \left[F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x)\right] \quad (68)$$

$F(x) = 0$ , hence

$$0 = \frac{x-x_k}{4} \left(F'(x_k) + 3F'\left(\frac{x+x_k}{2}\right) + F'(x)\right) + \left(\frac{x-x_k}{16}\right) \left[F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x)\right] \quad (69)$$

Multiplying through (69) by  $\frac{16}{x-x_k}$

$$0 = \frac{16}{x-x_k} F(x_k) + 4 \left(F'(x_k) + 3F'\left(\frac{x+x_k}{2}\right) + F'(x)\right) + \left[F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x)\right] \quad (70)$$

$$0 = \frac{16}{x-x_k} F(x_k) + 4F'(x_k) + 12F'\left(\frac{x+x_k}{2}\right) + 4F'(x) + F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x) \quad (71)$$

$$\frac{-16}{x-x_k} = 4F'(x_k) + 12F'\left(\frac{x+x_k}{2}\right) + 4F'(x) + F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x) \quad (72)$$

$$\frac{-16}{x-x_k} = 5F'(x_k) + 12F'\left(\frac{x+x_k}{2}\right) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + 5F'(x) \quad (73)$$

$$x - x_k = \frac{-16F(x_k)}{5F'(x_k) + 12F'\left(\frac{x+x_k}{2}\right) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + 5F'(x)} \quad (74)$$

$$x = x_k - 16 \left[5F'(x_k) + 12F'\left(\frac{x+x_k}{2}\right) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + 5F'(x)\right]^{-1} F(x_k) \quad (75)$$

Setting  $x = x_{k+1}$  and  $x_k = x_k$  in equation (75), we have

$$x_{k+1} = x_k - 16 \left[5F'(x_k) + 12F'\left(\frac{x_{k+1}+x_k}{2}\right) + 3F'\left(\frac{2x_k+x_{k+1}}{3}\right) + 3F'\left(\frac{x_k+2x_{k+1}}{3}\right) + 5F'(x_{k+1})\right]^{-1} F(x_k) \quad (76)$$

Setting  $F'\left(\frac{2x_k+x_{k+1}}{3}\right) \approx F'\left(\frac{x_k+2x_{k+1}}{3}\right) \approx F'\left(\frac{x_k+x_{k+1}}{2}\right)$ , equation (76) becomes

$$x_{k+1} = x_k - 16 \left[5F'(x_k) + 18F'\left(\frac{x_k+x_{k+1}}{2}\right) + 5F'(x_{k+1})\right]^{-1} F(x_k) \quad (77)$$

Which is an implicit equation because the presence of  $x_{k+1}$  at both sides of the equation, hence to avoid its implicit nature we use the  $(k+1)^{th}$  iteration of the Broyden's method in the right hand side. Thus we have;

$$x_{k+1} = x_k - 16[5F'(x_k) + 18F'(z_k) + 5F'(m_k)]^{-1} F(x_k) \quad (78)$$

with  $m_k = x_k - B_k^{-1} F(x_k)$  and  $z_k = \frac{x_k+m_k}{2}$

Now replacing  $F'(x_k)$ ,  $F'(m_k)$  and  $F'(z_k)$  by  $B(x_k)$ ,  $B(m_k)$  and  $B(z_k)$  respectively and use the same procedure as prescribed in [4, 5-9], we get

$$x_{k+1} = x_k - 16[5B(x_k) + 18B(z_k) + 5B(m_k)]^{-1} F(x_k) \quad (79)$$

Let  $B_k = 5B(x_k) + 18B(z_k) + 5B(m_k)$

$$\Rightarrow x_{k+1} = x_k - 16B_K^{-1}F(x_k) \quad (80)$$

#### Derivation of TS-3/8 Method

The proposal of the TS-3/8 method by the authors of an earlier publication [10], gave an idea which led to the development of the five new Broyden-like methods which have been described above.

Newton's method may be derived from the Taylor's series expansion of the function (of a single variable)  $f(x)$  about the point  $x$  given by

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

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 [20], it is obvious (as in (4)) that

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

Suppose  $x^*$  is a simple root of the nonlinear equation  $F(x) = 0$ , where  $F$  is sufficiently differentiable. Let  $F: D \subset R^n \rightarrow R^n$  be a smooth mapping that has continuous second order partial derivatives on a convex open set  $D \subset R^n$  and that has a locally unique root  $x$  in  $D$ ,  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))'$ ,  $x = (x_1, x_2, \dots, x_n)$  and  $f_k: R^n \rightarrow R$  is a nonlinear function, then we have equation (2) above.

Taking into consideration the two quadrature rules that is:

Trapezoidal quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{2}\right)(F(a) + F(b))$$

Simpson 3/8 quadrature rule

$$\int_a^b F(x) \approx \left(\frac{b-a}{8}\right)\left(f(a) + 3f\left(\frac{2a+b}{3}\right) + 3f\left(\frac{a+2b}{3}\right) + f(b)\right)$$

Approximating the integral in Equation (3) by the average of Trapezoidal and Simpson 3/8 (TS – 3/8) quadrature rules yields:

$$\int_{x_k}^x F'(t)dt = \frac{x-x_k}{4}[F'(x_k) + F'(x)] + \left(\frac{x-x_k}{16}\right)\left[F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right)\right] + F'(x) \quad (81)$$

Substituting Equation (80) into (3), we have

$$F(x) = F(x_k) + \frac{x-x_k}{4}[F'(x_k) + F'(x)] + \left(\frac{x-x_k}{16}\right)\left[F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right)\right] + F'(x)$$

Since  $F(x) = 0$ , we get

$$0 = F(x_k) + \frac{x-x_k}{4}[F'(x_k) + F'(x)] + \left(\frac{x-x_k}{16}\right)\left[F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right)\right] + F'(x) \quad (82)$$

Multiplying through equation (81) by  $\frac{16}{x-x_k}$ , we get

$$0 = \frac{16}{x-x_k}F(x_k) + 4[F'(x_k) + F'(x)] + \left[F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right)\right] + F'(x) \quad (83)$$

Expanding equation (82) gives

$$\begin{aligned} 0 &= \frac{16}{x-x_k}F(x_k) + 4F'(x_k) + 4F'(x) + F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + F'(x) \\ 0 &= \frac{16}{x-x_k}F(x_k) + 5F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) \\ &\Rightarrow \frac{-16}{x-x_k}F(x_k) = 5F'(x_k) + 3F'\left(\frac{2x_k+x}{3}\right) + 3F'\left(\frac{x_k+2x}{3}\right) + 5F'(x) \end{aligned} \quad (84)$$

$$\begin{aligned} \Rightarrow x - x_k &= \frac{-16F(x_k)}{5F'(x_k) + 3F'\left(\frac{2x_k + x}{3}\right) + 3F'\left(\frac{x_k + 2x}{3}\right) + 5F'(x)} \\ \Rightarrow x &= x_k - \frac{16F(x_k)}{5F'(x_k) + 3F'\left(\frac{2x_k + x}{3}\right) + 3F'\left(\frac{x_k + 2x}{3}\right) + 5F'(x)} \\ x &= x_k - 16 \left[ 5F'(x_k) + 3F'\left(\frac{2x_k + x}{3}\right) + 3F'\left(\frac{x_k + 2x}{3}\right) + 5F'(x) \right]^{-1} F(x_k) \end{aligned} \quad (85)$$

Setting  $x = x_{k+1}$  and  $x_k = x_k$  in equation (84), we have

$$x_{k+1} = x_k - 16 \left[ 5F'(x_k) + 3F'\left(\frac{2x_k + x_{k+1}}{3}\right) + 3F'\left(\frac{x_k + 2x_{k+1}}{3}\right) + 5F'(x_{k+1}) \right]^{-1} F(x_k) \quad (86)$$

Setting  $F'\left(\frac{2x_k + x_{k+1}}{3}\right) \approx F'\left(\frac{x_k + 2x_{k+1}}{3}\right) \approx F'\left(\frac{x_k + x_{k+1}}{2}\right)$ , equation (85) becomes

$$x_{k+1} = x_k - 16 \left[ 5F'(x_k) + 3F'\left(\frac{x_k + x_{k+1}}{2}\right) + 3F'\left(\frac{x_k + x_{k+1}}{2}\right) + 5F'(x_{k+1}) \right]^{-1} F(x_k) \quad (87)$$

$$x_{k+1} = x_k - 16 \left[ 5F'(x_k) + 6F'\left(\frac{x_k + x_{k+1}}{2}\right) + 5F'(x_{k+1}) \right]^{-1} F(x_k) \quad (88)$$

In (86) we have an implicit equation because of the presence of  $x_{k+1}$  on both sides of it. To avoid its implicit nature we use the  $(k+1)^{th}$  iteration of the Broyden's method on the right hand side of (87). Thus we have:

$$x_{k+1} = x_k - 16[5F'(x_k) + 6F'(z_k) + 5F'(m_k)]^{-1} F(x_k) \quad (89)$$

with  $m_k = x_k - B_k^{-1}F(x_k)$  and  $z_k = \frac{x_k + m_k}{2}$

Now replacing  $F'(x_k)$ ,  $F'(m_k)$  and  $F'(z_k)$  by  $B(x_k)$ ,  $B(m_k)$  and  $B(z_k)$  respectively and using the same procedure as prescribed in [4, 5-9], we get

$$x_{k+1} = x_k - 16[5B(x_k) + 6B(z_k) + 5B(m_k)]^{-1} F(x_k)$$

Let  $B_k = 5B(x_k) + 6B(z_k) + 5B(m_k)$

$$\Rightarrow x_{k+1} = x_k - 16B_k^{-1}F(x_k) \quad (90)$$

Hence we have the following method using initial matrix  $B_0 = I$  and an initial guess  $x_0$ . For a given  $x_0$  using initial matrix  $B_0 = I$ , an approximated solution for  $x_{k+1}$  can be computed by the iterative schemes as in [20];

$$\begin{aligned} m_k &= x_k - B_k^{-1}F(x_k) \\ x_{k+1} &= x_k - 16[5B(x_k) + 6B(z_k) + 5B(m_k)]^{-1} F(x_k) \end{aligned} \quad (91)$$

where:  $z_k = \frac{x_k + m_k}{2}$ ,  $k = 0, 1, \dots$

## 4. Numerical Tests

In order to evaluate the performance of the new methods, they were tested, together with four other existing methods (i.e. Classical Broyden Method (CB), Trapezoidal-Simpson Method (TS), Midpoint-Trapezoidal (MT), Trapezoidal Simpson Midpoint Method (TSM) method, on four benchmark problems (Osinuga et al, 2018), using a set of seven dimensions ranging from 5 to 1065 variables. The results were then compared on the basis of two main characteristic features namely, the number of iterations (NI) and the Central Processing Unit (CPU) time in seconds. The computation was done in MATLAB R2020b with a double precision arithmetic on a computer with specification as follows; processor: AMD EI-2100APU with Radeon™

Graphics 1.00GHz, Installed memory (RAM): 4.00GB and the system type is 64 – bit Operating System, x 64 – based processor. The programme was designed to terminate whenever the number of iterations reached 500; any failure of a method to satisfy these convergence criteria is denoted by a dash (i.e. '-') as in Table 1.

The problems that were used for the test are:

Problem One

$$F_i(x) = x_i x_{i+1} - 1, F_n(x) = x_n x_1 - 1, i = 1, 2, \dots, n - 1 \text{ and } x_0 = (0.8, 0.8, \dots, 0.8)^T$$

Problem Two

$$F_i(x) = x_i x_{i+1} - 1, F_n(x) = x_n x_1 - 1, i = 1, 2, \dots, n - 1 \text{ and } x_0 = (0.5, 0.5, \dots, 0.5)^T$$

Problem Three

$$F_i(x) = x_i^2 - \cos(x_1 - 1), i = 1, 2, \dots, n \text{ and } x_0 = (0.5, 0.5, \dots, 0.5)^T$$

Problem Four

$$F_i(x) = \exp(x_i^2 - 1) - \cos(1 - x_i^2), i = 1, 2, \dots, n \text{ and } x_0 = (0.5, 0.5, \dots, 0.5)^T$$

## 5. Results and Discussion

This section present results of numerical computations which was carried out with the aim of comparing the performances of the Boryden-like methods. Table 2 gives detailed results of four existing Broyden-like methods while Table 3 contains results of the newly developed methods in addition to the TS-3/8 method proposed in an earlier publication [10]. All numerical computations were carried out using the same bench mark problems in order to compare the performance of the methods.

Results from Table 2 showed that the Classical Broyden method was the only one which could not satisfy the convergence criteria for the bench mark problem 3, all the other existing Broyden-like methods converged for all the problems adopted for this study.

The results in Table 2 also showed that the methods performed differently with all the four bench mark problems considered in this study. In some cases such as in bench mark problems 1 and 2, the TS method performed extremely better in terms of recording the least CPU time while for problems 3 and 4, the TSMM performed better than the other existing method and this can be seen in Table 3.

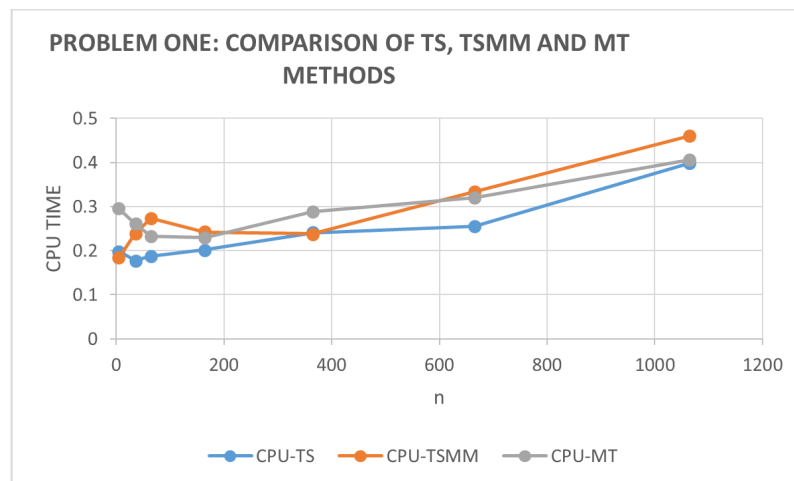
In order to give clarity to how the methods were compared, they were all applied to solve the bench mark problems selected for this study, with the hope of getting the method which would solve a problem with the least number of iterations and CPU time. In a situation where methods had the same number of iterations, more attention was paid to the CPU time needed for the computation and hence a graph was constructed to give a clear picture of which method performed better in terms of the

CPU time. Figures 1, 2, 3, 4, 5 and 6 gives an indication that the methods compared in the graph had the same number of iterations hence the need to have a clearer view of how well they performed in terms of their CPU time.

**Table 2.** Comparison of Existing Broyden-like Methods.

Problem n		1		2		4		5	
		CB		TS		TSM		MT	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU
1	5	6	0.051	4	0.198	4	0.185	4	0.295
	35	6	0.155	4	0.178	4	0.239	4	0.262
	65	5	0.215	4	0.188	4	0.273	4	0.232
	165	5	0.515	4	0.202	4	0.242	4	0.229
	365	5	0.515	4	0.241	4	0.238	4	0.289
	665	5	2.794	4	0.255	4	0.334	4	0.320
	1065	6	5.131	4	0.399	4	0.461	4	0.406
2	5	6	0.047	4	0.185	4	0.220	4	0.224
	35	6	0.164	4	0.189	4	0.235	4	0.226
	65	6	0.257	4	0.192	4	0.212	4	0.305
	165	6	1.178	4	0.197	4	0.245	4	0.314
	365	6	2.971	4	0.221	4	0.264	4	0.266
	665	6	2.934	4	0.286	4	0.314	4	0.354
	1065	6	6.884	4	0.451	4	0.396	4	0.459
3	5	-	-	4	0.224	4	1.399	5	0.266
	35	-	-	7	0.183	5	0.238	5	0.245
	65	-	-	7	0.197	5	0.263	5	0.250
	165	-	-	7	0.236	5	0.273	5	0.241
	365	-	-	7	0.260	5	0.292	5	0.283
	665	-	-	7	0.289	5	0.358	5	0.429
	1065	-	-	7	0.418	5	0.484	5	0.530
4	5	6	0.042	4	0.259	9	0.211	6	0.183
	35	6	0.138	4	2.522	9	0.254	6	0.236
	65	6	0.388	4	8.343	9	0.226	6	0.274
	165	6	0.767	4	47.078	9	0.286	6	0.244
	365	6	2.554	4	232.026	9	0.290	6	0.268
	665	6	6.492	4	773.373	9	0.384	6	0.326
	1065	6	8.312	4	1926.21	9	0.861	6	0.582

Table 3 gives a summary of how each of the existing methods performed when subjected to the bench mark problems. The best method is denoted first (1<sup>st</sup>) while the least performed method is denoted fourth (4<sup>th</sup>).



**Figure 1.** Comparison of TS, TSM and MT results for problem one.

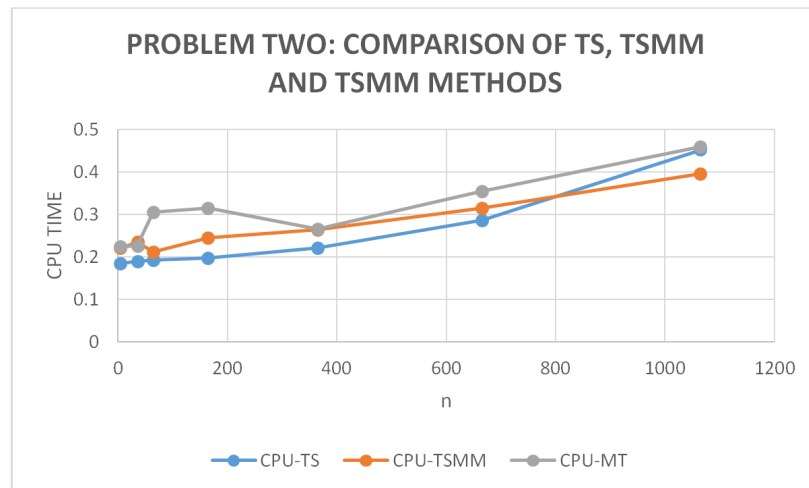


Figure 2. Comparison of TS, TSMM and MT RESULTS for problem two.

Table 3. Ranking of Performance of Existing Broyden-like Methods.

Problem	Ranking of Methods			
	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
1	TS	MT	TSMM	CB
2	TS	TSMM	MT	CB
3	TSMM	MT	TS	CB
4	TSMM	MT	CB	TS

Table 4 presents result of the newly developed Broyden-

like methods which indicates the number of iterations and CPU time needed to solve the given bench mark problems. Similar to the results presented in Table 2, it was observed that the MS-1/3 was the only method which could not satisfy the convergence criteria for one of the bench mark problems (i.e. Problem 4). Again, the methods performed differently with the different bench mark problems they were applied on, giving an indication that the different variant of Broyden-like methods may perform better on different problems./

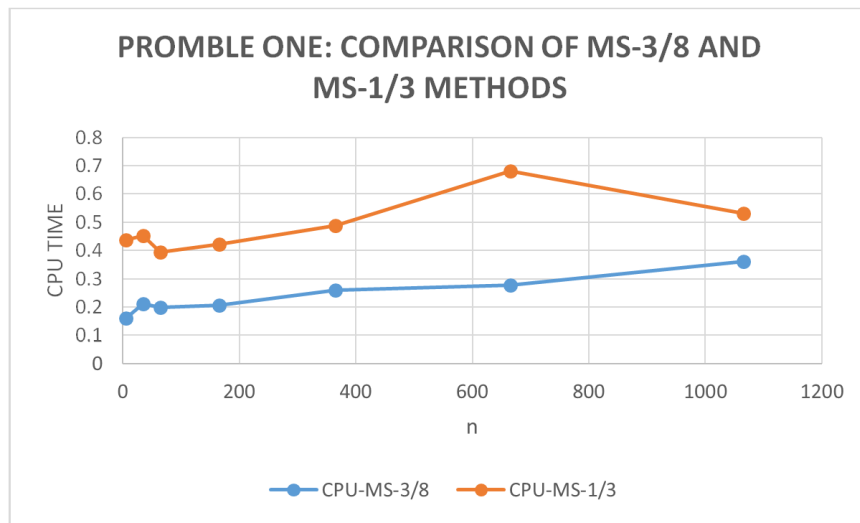
Table 4. Comparison of Newly Developed Broyden-like Methods.

Problem	n	1		2		3		4		5		6	
		TS-3/8		MS-3/8		TS-1/3		MS-1/3		SS-3/8		SS-1/3	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU	NI	CPU
1	5	4	0.186	3	0.160	4	0.305	3	0.436	47	0.206	19	0.151
	35	4	0.201	3	0.211	4	0.615	3	0.452	47	0.208	19	0.273
	65	4	0.188	3	0.199	4	0.452	3	0.393	47	0.191	19	0.461
	165	4	0.209	3	0.207	4	0.335	3	0.423	47	0.366	19	0.482
	365	4	0.213	3	0.259	4	0.438	3	0.487	47	0.705	19	0.492
	665	4	0.314	3	0.277	4	0.511	3	0.681	47	1.068	19	0.878
	1065	4	0.398	3	0.361	4	0.574	3	0.531	47	2.325	19	1.624
2	5	4	0.220	4	0.203	5	0.319	4	0.441	48	0.222	19	0.318
	35	4	0.210	4	0.208	4	0.345	4	0.484	49	0.214	19	0.418
	65	4	0.187	4	0.121	4	0.475	4	0.484	49	0.218	19	0.432
	165	4	0.239	4	0.266	4	0.499	4	0.427	49	0.270	19	0.371
	365	4	0.244	4	0.282	4	0.345	4	0.464	49	0.389	19	0.546
	665	4	0.278	4	0.273	4	0.579	4	0.656	49	0.659	19	0.887
	1065	4	0.859	4	0.370	4	0.659	4	0.642	49	1.409	19	1.572
3	5	7	0.206	4	0.155	5	0.394	4	0.362	48	0.168	19	0.316
	35	7	0.208	4	0.194	5	0.401	4	0.524	48	0.237	19	0.428
	65	7	0.202	4	0.201	5	0.402	4	0.448	48	0.283	19	0.203
	165	7	0.220	4	0.218	5	0.426	4	0.337	48	0.989	19	0.448
	365	7	0.280	4	0.229	5	0.688	4	0.365	48	0.494	19	0.554
	665	7	1.238	4	0.276	5	0.562	4	0.516	48	0.851	19	1.048
	1065	7	0.659	4	0.374	5	0.855	4	0.645	48	1.800	19	1.685
4	5	4	0.202	1	0.203	7	1.627	-	-	52	0.228	24	0.391
	35	4	2.635	1	0.193	7	0.311	-	-	52	0.214	25	0.355
	65	4	8.343	1	0.228	7	0.415	-	-	52	0.224	25	0.418
	165	4	51.077	1	0.198	7	0.343	-	-	52	0.223	25	0.427
	365	4	227.191	1	0.231	7	0.476	-	-	52	0.271	25	0.475
	665	4	759.877	1	0.247	7	0.614	-	-	52	0.348	25	0.917
	1065	4	1979.338	1	0.243	7	0.725	-	-	52	0.682	25	1.511

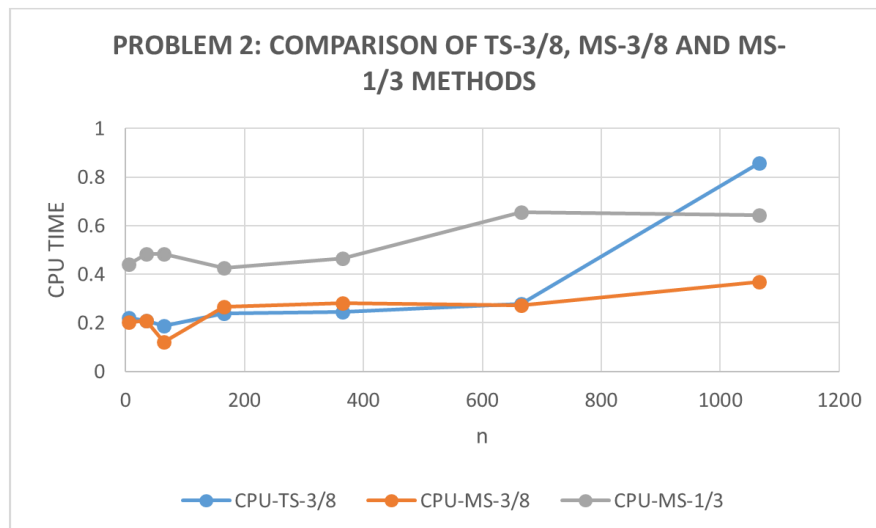
Results from Table 4 have shown clearly that the MS-3/8 method is outstanding in terms of its performance in all the

bench mark problems considered in this study. It is followed closely by the TS-3/8 method which performed well. Table 5

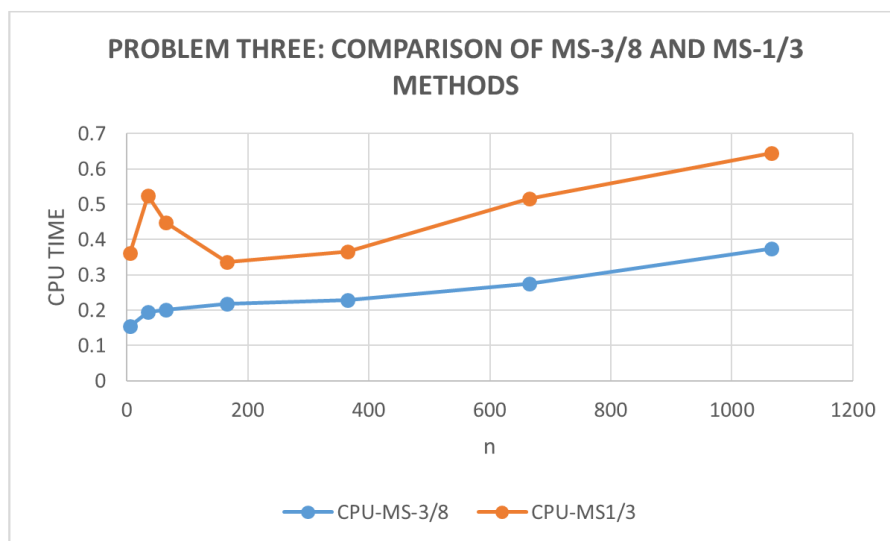
summaries the performance of the newly developed methods as they were applied on the bench mark problems.



*Figure 3. Comparison of MS-3/8 and MS-1/3 results for problem one.*



*Figure 4. Comparison of TS-3/8, MS-3/8 and MS-1/3 results for problem two.*

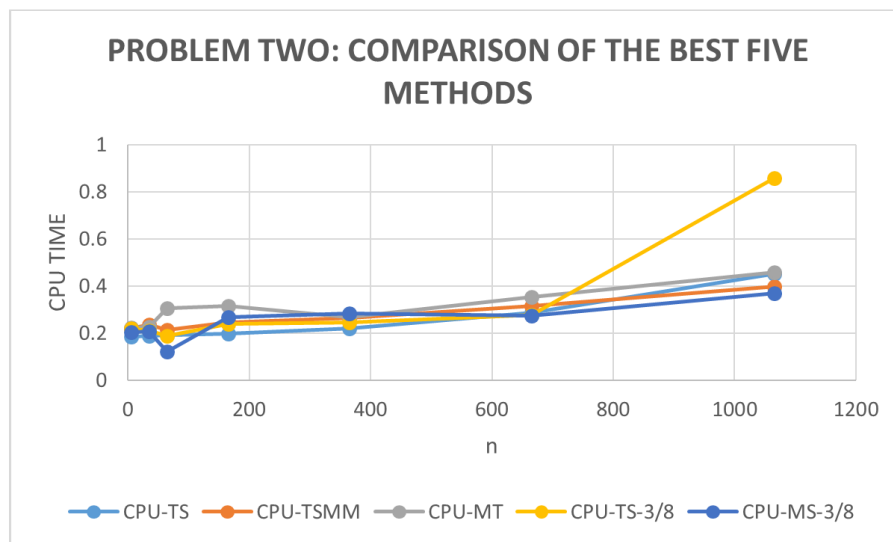


*Figure 5. Comparison of MS-3/8 and MS-1/3 results for problem three.*

**Table 5.** Ranking of Performance of Newly Developed Broyden-like Methods.

Problem	Ranking of Methods					
	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	5 <sup>th</sup>	6 <sup>th</sup>
1	MS-3/8	MS-1/3	TS-3/8	TS-1/3	SS-1/3	MS-3/8
2	MS-3/8	TS-3/8	MS-1/3	TS-1/3	SS-1/3	MS-3/8
3	MS-3/8	MS-1/3	TS-1/3	TS-3/8	SS-1/3	MS-3/8
4	MS-3/8	TS-3/8	TS-1/3	SS-1/3	SS-3/8	MS-1/3

Based on the results obtained in this study summarized in Tables 2, 3, 4 and 5 using the same bench mark problems, it can be concluded by inspection that the five best methods from this study can be ranked as in Table 6.

**Figure 6.** Comparison of the best five methods results for problem two.**Table 6.** Ranking of the Best Five Broyden-Like Methods.

Indicators	Ranking According to the Least Number of Iterations and CPU Time				
	1 <sup>st</sup> Method	2 <sup>nd</sup> Method	3 <sup>rd</sup> Method	4 <sup>th</sup> Method	5 <sup>th</sup> Method
Number of Iterations	MS-3/8	TS-3/8	TS	MT	TSMM
CPU Time	MS-3/8	TS-3/8	TS	MT	TSMM

## 6. Convergence Analysis

This section presents the convergence analysis for the best ranked Broyden-like method.

### Convergence of the MS-3/8 Method

The properties of local convergence of the proposed methods are presented here with the following standard assumptions on the nonlinear function  $F$ :

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

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

Definition 1.0 (q-super-linear convergence) [12]

Let  $x_k$  and  $x^* \in R^n$ . Then  $x_k \rightarrow x^*$  is q-superlinear if

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

### Lemma 1.1 [22]

Let  $F: R^n \rightarrow R^n$  be continuous and differentiable on an open convex set  $D \subset R^n, x \in D$ . If  $F'(x)$  is Lipschitz continuous with Lipschitz constant  $\lambda$ , 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  $\epsilon$  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

$$\lambda \max\{\|u - x\|, \|v - x\|\} \leq \epsilon.$$

### Lemma 1.2 [22]

Let  $x_k \in R^n, k \geq 0$ . If  $x_k$  converges q-super-linearly to  $x^* \in R^n$ , then

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

Here, we present the main result which is a modified result [18], to prove that the local order of convergence analysis is super-linear.

### Theorem 1.0

Let  $F: R^n \rightarrow R^n$  satisfy the hypothesis of Lemma 1.1 on

the set  $D$ . Let  $B_k$  be a sequence of non-singular matrices in the linear space  $L(R^n)$  of real matrices of order  $n$ . Suppose for some  $x_0$  the sequence  $x_k$  generated by (42) remains in  $D$  and  $\lim_{k \rightarrow \infty} x_k = x^*$  for each  $x_k \neq x^*$ . Then the sequence  $\{x_k\}$  converges  $q$ -super-linearly to  $x^*$  and  $F(x^*) = 0$  if and only if

$$\lim_{k \rightarrow \infty} \frac{\left\| \frac{1}{16} B_k F'(x^*) s_k \right\|}{\|s_k\|} = 0 \quad (92)$$

Where  $s_k = x_{k+1} - x_k$  and  $B_k = 5B(x_k) + 3B(z_k) + 5B(m_k)$ .

*Proof*

Given that (92) holds; it implies that (42) becomes

$$0 = \frac{1}{16} B_k s_k + F(x_k)$$

$$\| -F(x_{k+1}) \| = \left\| \left( \frac{1}{16} B_k - F'(x^*) \right) s_k + (-F(x_{k+1}) + F(x_k) + F'(x^*) s_k) \right\|$$

Using vector norm properties, it implies that;

$$\| -F(x_{k+1}) \| \leq \left\| \left( \frac{1}{16} B_k - F'(x^*) \right) s_k \right\| + \| (-F(x_{k+1}) + F(x_k) + F'(x^*) s_k) \|$$

Dividing through by  $\|s_k\|$ , we have;

$$\frac{\| -F(x_{k+1}) \|}{\|s_k\|} \leq \frac{\left\| \left( \frac{1}{16} B_k - F'(x^*) \right) s_k \right\|}{\|s_k\|} + \frac{\| (-F(x_{k+1}) + F(x_k) + F'(x^*) s_k) \|}{\|s_k\|}$$

Using Lemma 1.1

$$\| -F(x_{k+1}) \| \leq \frac{\left\| \left( \frac{1}{16} B_k - F'(x^*) \right) s_k \right\|}{\|s_k\|} + \lambda \max\{\|x_{k+1} - x^*\|, \|x_k - x^*\|\}$$

Since  $x_{k+1} \rightarrow x^* \forall k$ , then from (91), we have

$$\lim_{k \rightarrow \infty} \frac{\|F(x_{k+1})\|}{\|s_k\|} \leq \frac{\left\| \left( \frac{1}{16} B_k - F'(x^*) \right) s_k \right\|}{\|s_k\|} + \lambda \max\{\|x_{k+1} - x^*\|, \|x_k - x^*\|\}$$

$$F(x^*) = F(\lim_{k \rightarrow \infty} x_k) = \lim_{k \rightarrow \infty} F(x_k) = 0$$

But  $F'(x^*)$  is non-singular, thus by Lemma 1.1  $\exists \rho > 0, k_0 \geq 0$  such that we have;

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

For all  $k \geq k_0$ , we have

$$\begin{aligned} 0 &= \lim_{k \rightarrow \infty} \frac{\|F(x_{k+1})\|}{\|s_k\|} \\ &\geq \lim_{k \rightarrow \infty} \frac{1}{\rho} \frac{\|x_{k+1} - x^*\|}{\|s_k\|} \\ &\geq \lim_{k \rightarrow \infty} \frac{1}{\rho} \frac{\|x_{k+1} - x^*\|}{\|x_{k+1} - x^*\| + \|x_{k+1} - x^*\|} \end{aligned}$$

$$0 = \frac{1}{16} B_k s_k + F(x_k) - F'(x^*) s_k + F'(x^*) s_k$$

$$0 = \frac{1}{16} B_k s_k - F'(x^*) s_k + F(x_k) + F'(x^*) s_k$$

$$-F(x_{k+1}) + F(x_{k+1}) = \left( \frac{1}{16} B_k - F'(x^*) \right) s_k + F(x_k) + F'(x^*) s_k$$

$$\begin{aligned} -F(x_{k+1}) &= \left( \frac{1}{16} B_k - F'(x^*) \right) s_k + (-F(x_{k+1}) + F(x_k) \\ &\quad + F'(x^*) s_k) \end{aligned}$$

Taking norm of both sides, we have:



$$= \lim_{k \rightarrow \infty} \frac{\frac{1}{\rho} t_k}{1 + t_k}$$

where

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

It implies that

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

Therefore  $x_k$  converges q-super-linearly to  $x^*$ . Conversely, supposed that  $x_k$  converges q-super-linearly to  $x^*$  and  $F(x^*) = 0$ . Then by Lemma 1.1, there exist a  $\rho > 0$  such that we have

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

Then

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

Using Lemma 1.2, we have

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

It implies that

$$\frac{\left\| \left( \frac{1}{16} B_k - F'(x^*) \right) s_k \right\|}{\|s_k\|} \leq \lim_{k \rightarrow \infty} \frac{\|F(x_{k+1})\|}{\|s_k\|} + \lim_{k \rightarrow \infty} \frac{\|(-F(x_{k+1}) + F(x_k) + F'(x^*)s_k)\|}{\|s_k\|} \leq 0 + \lambda \max\{\|x_{k+1} - x^*\|, \|x_k - x^*\|\}$$

Since  $x_k$  converges to  $x^*$ , then

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

which proves that

$$\frac{\left\| \left( \frac{1}{16} B_k - F'(x^*) \right) s_k \right\|}{\|s_k\|} = 0$$

## 7. Conclusion

The study has shown the processes leading to the development of five new Broyden-like methods namely; TS – 1/3, MS – 1/3, MS – 3/8, SS – 1/3 and SS – 3/8. All the newly proposed methods in this study, after being subjected to solving some bench mark problems together with selected existing methods, produced satisfactory results and in some cases, did extremely better than the existing methods.

The data analysed in this study confirms that the MS – 3/8 has shown a convincing prove that it is efficient and robust when applied to solve nonlinear system of equations, as compared with all the other methods considered in the study.

Next to the MS – 3/8 method is TS – 3/8 and TS methods. The case where the MS – 3/8 method had the same number of iterations with these methods, its CPU time was lesser than them, making it the preferred choice of method.

Based on the results from the study, the MS – 3/8 method is seen to be the most efficient among all the methods considered in the study. It can therefore be concluded that the study has efficiently developed and presented new Broyden-like methods and have compared these new methods with existing ones and these two findings make up for the first two specific objectives of the entire research.

Subsequent report would now employ the newly developed method to solving a modelled photovoltaic system

problem with the aim of estimating its parameters.

## 8. Recommendations

An area to consider for the application of this method would be apply it on a real life problem model to know if it would show a similar results as in the bench mark problems considered in this study. This might be considered in future studies.

Gathering from the idea of using weighted combination of quadrature rules in the development of these new Broyden – like methods, there exist the possibility that applying this same procedure on other quadrature rules may yield other new methods.

## References

- [1] Al-Towaiq, M. H., & Abu Hour, Y. S. (2017). Two improved classes of Broyden's methods for solving nonlinear systems of equations. *JOURNAL OF MATHEMATICS AND COMPUTER SCIENCE-JMCS*, 17 (1), 22-31.
- [2] Autar, K., Egbu, E. K., Duc, N. (2017). *Numerical Methods with Applications*. <http://numericalmethods.eng.usf.edu>.
- [3] Azure, I., Aloliga, G., & Doabil, L. (2020). Comparative Study of Numerical Methods for Solving Non-linear Equations Using Manual Computation. *Mathematics Letters*, 5 (4), 41.
- [4] Cordero, A., Torregrosa, J. R., & Vassileva, M. P. (2012). Pseudocomposition: a technique to design predictor–corrector methods for systems of nonlinear equations. *Applied Mathematics and Computation*, 218 (23), 11496-11504.
- [5] Cordero, A., & Torregrosa, J. R. (2006). Variants of Newton's method for functions of several variables. *Applied Mathematics and Computation*, 183 (1), 199-208.
- [6] Darvishi, M. T., & Shin, B. C. (2011). High-order Newton-Krylov methods to solve systems of nonlinear equations. *Journal of the Korean Society for Industrial and Applied Mathematics*, 15 (1), 19-30.
- [7] Dhamacharoen, A. (2014). An efficient hybrid method for solving systems of nonlinear equations. *Journal of Computational and Applied Mathematics*, 263, 59-68.
- [8] Frontini, M. A. R. C. O., & Sormani, E. (2003). Some variant of Newton's method with third-order convergence. *Applied Mathematics and Computation*, 140 (2-3), 419-426.
- [9] Hafiz, M. A., & Bahgat, M. S. (2012). An efficient two-step iterative method for solving system of nonlinear equations. *Journal of Mathematics Research*, 4 (4), 28.
- [10] Isaac, A., Stephen, T. B., Seidu, B. (2021). A New Trapezoidal-Simpson 3/8 Method for Solving Systems of Nonlinear Equations. *American Journal of Mathematical and Computer Modelling*, 6 (1), 1-8.
- [11] Jain, M. K. (2003). *Numerical methods for scientific and engineering computation*. New Age International.
- [12] Kelley, C. T. (1995). *Iterative methods for linear and nonlinear equations*. Society for Industrial and Applied Mathematics.
- [13] Kou, J., Li, Y., & Wang, X. (2007). A composite fourth-order iterative method for solving non-linear equations. *Applied Mathematics and Computation*, 184 (2), 471-475. (11)
- [14] Li, Y., Wei, Y., & Chu, Y. (2015). Research on solving systems of nonlinear equations based on improved PSO. *Mathematical Problems in Engineering*, 2015. (1).
- [15] Luo, Y. Z., Tang, G. J., & Zhou, L. N. (2008). Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method. *Applied Soft Computing*, 8 (2), 1068-1073. (12).
- [16] Mahwash, K. N., & Gyang, G. D. (2018). Numerical Solution of Nonlinear Systems of Algebraic Equations.
- [17] Mo, Y., Liu, H., & Wang, Q. (2009). Conjugate direction particle swarm optimization solving systems of nonlinear equations. *Computers & Mathematics with Applications*, 57 (11-12), 1877-1882. (13).
- [18] Mohammad, H., & Waziri, M. Y. (2015). On Broyden-like update via some quadratures for solving nonlinear systems of equations. *Turkish Journal of Mathematics*, 39 (3), 335-345.
- [19] Muhammad, K., Mamat, M., & Waziri, M. Y. (2013). A Broyden's-like Method for solving systems of Nonlinear Equations. *World Appl Sc J*, 21, 168-173.
- [20] Osinuga, I. A., & Yusuff, S. O. (2018). Quadrature based Broyden-like method for systems of nonlinear equations. *Statistics, Optimization & Information Computing*, 6 (1), 130-138.
- [21] Osinuga, I. A., & Yusuff, S. O. (2017). Construction of a Broyden-like method for Nonlinear systems of equations. *Annals. Computer Science Series*, 15 (2), 128-135.
- [22] Van de Rotten, B., & Lunel, S. V. (2005). A limited memory Broyden method to solve high-dimensional systems of nonlinear equations. In *EQUADIFF 2003* (pp. 196-201).
- [23] Weerakoon, S., & Fernando, T. G. I. (2000). A variant of Newton's method with accelerated third-order convergence. *Applied Mathematics Letters*, 13 (8), 87-93.