

Article

# Efficient Two-Step Fifth-Order and Its Higher-Order Algorithms for Solving Nonlinear Systems with Applications

Parimala Sivakumar <sup>†</sup> and Jayakumar Jayaraman <sup>\*,†</sup> 

Department of Mathematics, Pondicherry Engineering College, Pondicherry 605014, India; parimalas@pec.edu

\* Correspondence: jjayakumar@pec.edu; Tel.: +91-996-526-9220

† Both authors have read and approved the final manuscript.

Received: 20 February 2019; Accepted: 26 March 2019; Published: 1 April 2019



**Abstract:** This manuscript presents a new two-step weighted Newton's algorithm with convergence order five for approximating solutions of system of nonlinear equations. This algorithm needs evaluation of two vector functions and two Frechet derivatives per iteration. Furthermore, it is improved into a general multi-step algorithm with one more vector function evaluation per step, with convergence order  $3k + 5$ ,  $k \geq 1$ . Error analysis providing order of convergence of the algorithms and their computational efficiency are discussed based on the computational cost. Numerical implementation through some test problems are included, and comparison with well-known equivalent algorithms are presented. To verify the applicability of the proposed algorithms, we have implemented them on 1-D and 2-D Bratu problems. The presented algorithms perform better than many existing algorithms and are equivalent to a few available algorithms.

**Keywords:** Newton's method; system of nonlinear equations; higher-order method; multi-step method; Frechet derivative; computational efficiency

## 1. Introduction

The design of iterative algorithms for solving a nonlinear system of equations is a most needed and challenging task in the domain of numerical analysis. Nonlinearity is a phenomenon that occurs in physical and natural events. Phenomena which have an inherent quality of nonlinearity occur frequently in fluid and plasma mechanics, gas dynamics, combustion, ecology, biomechanics, elasticity, relativity, chemical reactions, economics modeling problems, transportation theory, etc. Due to this frequency, many works related to the current mathematical research provide importance for the development and analysis of methods for nonlinear systems. Hence, finding a solution  $\alpha$  of the nonlinear system  $\Phi(x) = 0$  is a classical and difficult problem that unlocks the behavior pattern of many application problems in science and engineering, wherein  $\Phi : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a sufficiently Frechet differentiable function in an open convex set  $D$ .

In the last decade, many iterative techniques have been proposed for solving a system of nonlinear equations. See, for instance, [1–7] and the references therein. The popular method for getting a solution  $\alpha \in D$  is the Newton's method ( $2^{nd}$  NM)

$$x^{(r+1)} = G_{2^{nd}NM}(x^{(r)}) = x^{(r)} - [\Phi'(x^{(r)})]^{-1}\Phi(x^{(r)}), \quad r = 0, 1, 2, \dots \quad (1)$$

where  $G_{method}$  denotes any iterative scheme and  $\Phi'(x^{(r)})^{-1}$  denotes the inverse of first Frechet derivative  $\Phi'(x^{(r)})$ . This  $2^{nd}$  NM method needs evaluation of one function, same number of derivative and matrix inversion per iteration which produces second-order convergence.

Another classical scheme for solving system of nonlinear equations is a two-step third-order variant of Newton’s method ( $3^{rd}TM$ ) proposed by Traub [8] is given by

$$x^{(r+1)} = G_{3^{rd}TM}(x^{(r)}) = G_{2^{nd}NM}(x^{(r)}) - [\Phi'(x^{(r)})]^{-1}\Phi(G_{2^{nd}NM}(x^{(r)})).$$

The fourth-order Newton’s method ( $4^{th}NR$ ) formed using two steps is given by

$$x^{(r+1)} = G_{4^{th}NR}(x^{(r)}) = G_{2^{nd}NM}(x^{(r)}) - \Phi'[G_{2^{nd}NM}(x^{(r)})]^{-1}\Phi(G_{2^{nd}NM}(x^{(r)})), \tag{2}$$

was proposed by Noor et al. [5] based on the variational iteration technique. Abad et al. [1] combined the Newton and Traub method to obtain a fifth-order method ( $5^{th}ACT$ ) with three steps, given as below:

$$x^{(r+1)} = G_{5^{th}ACT}(x^{(r)}) = G_{3^{rd}TM}(x^{(r)}) - [\Phi'(G_{2^{nd}NM}(x^{(r)}))]^{-1}\Phi(G_{3^{rd}TM}(x^{(r)})). \tag{3}$$

Sharma and Arora [6] proposed an eighth-order three-step method ( $8^{th}SA$ ) given by

$$x^{(r+1)} = G_{8^{th}SA}(x^{(r)}) = z(x^{(r)}) - \left(\frac{7}{2}I - G(x^{(r)})\left(4I - \frac{3}{2}G(x^{(r)})\right)\right)[\Phi'(x^{(r)})]^{-1}\Phi(z(x^{(r)})),$$

where  $z(x^{(r)}) = G_{2^{nd}NM}(x^{(r)}) - \left(\frac{13}{4}I - G(x^{(r)})\left(\frac{7}{2}I - \frac{5}{4}G(x^{(r)})\right)\right)[\Phi'(x^{(r)})]^{-1}\Phi(G_{2^{nd}NM}(x^{(r)})),$   $G(x^{(r)}) = [\Phi'(x^{(r)})]^{-1}\Phi'(G_{2^{nd}NM}(x^{(r)})).$   $\tag{4}$

Recently, Madhu et al. [9] ( $5^{th}MBJ$ ) proposed a fifth-order two-step method given by

$$x^{(r+1)} = G_{5^{th}MBJ}(x^{(r)}) = G_{2^{nd}NM}(x^{(r)}) - H_1(x^{(r)})[\Phi'(x^{(r)})]^{-1}\Phi(G_{2^{nd}NM}(x^{(r)})),$$

$$H_1(x^{(r)}) = 2I - \tau(x^{(r)}) + \frac{5}{4}(\tau(x^{(r)}) - I)^2, \quad \tau(x^{(r)}) = [\Phi'(x^{(r)})]^{-1}\Phi'(G_{2^{nd}NM}(x^{(r)})). \tag{5}$$

Furthermore, this method was extended to produce  $3k + 5$  ( $k \geq 1$ ) order of convergence known as  $(3k + 5)^{th}MBJ$  method using an additional function evaluation and it is given below

$$x^{(r+1)} = G_{(3k+2)^{th}MBJ}(x^{(r)}) = \mu_k(x^{(r)}),$$

$$\mu_j(x^{(r)}) = \mu_{j-1}(x^{(r)}) - H_2(x^{(r)})[\Phi'(x^{(r)})]^{-1}\Phi(\mu_{j-1}(x^{(r)})), \quad H_2(x^{(r)}) = 2I - \tau(x^{(r)}) + \frac{3}{2}(\tau(x^{(r)}) - I)^2, \tag{6}$$

$$\mu_0(x^{(r)}) = G_{5^{th}MBJ}(x^{(r)}), \quad j = 1, 2, \dots, k, k \geq 1.$$

To derive the order of convergence for the iterative methods, higher-order derivatives are used although such derivatives are not present in the formulas. The solutions of the equation  $\Phi(x) = 0$  are rarely found in closed form and hence most of the methods for solving such equations are usually iterative. Convergence analysis of these iterative methods is an important part in establishing the order of any iterative method. As the convergence domain is narrow, one needs additional hypotheses to enlarge it. Furthermore, knowledge of initial approximations requires the convergence radius. Therefore, the applicability of the methods depends upon the assumptions on the higher-order derivatives of the function. Many research papers which are concerned with the local and semi-local convergence of iterative solutions are available in [10–13]. One of the most attractive features of each numerical algorithm is how the procedure deals with large-scale systems of nonlinear equations. Recently, Gonglin Yuan et al. [14,15] proposed conjugate gradient algorithms with global convergence under suitable conditions that possess some good properties for solving unconstrained optimization problems.

The primary goal of this manuscript is to propose higher-order iterative techniques which consumes less computational cost for very large nonlinear systems. Motivated by the different methods available in literature, a new efficient two-step method is proposed with convergence order five for solving systems of nonlinear equations. This new method consists of two functional evaluations,

two Frechet derivative evaluations and two inverse evaluations per iteration. The order of this method can be increased by three units whenever an additional step is included. This idea is generalized for obtaining new multi-step methods of order  $3k + 5, k \geq 1$ , increasing the convergence order by three units for every step. Also, this method requires only one new function evaluation for every new step.

The remaining part of this manuscript is organized as follows. Section 2 presents two new algorithms, one with convergence order five and the other is its multi-step version with order  $3k + 5, k \geq 1$ . In Section 3, the analysis of convergence of the new algorithms are presented. Computational efficiencies of the proposed algorithms are calculated based on the computational cost and a comparison with other methods in terms of ratio is reported in Section 4. Numerical results for some test problems and their comparison with few available equivalent methods are given in Section 5. Two application problems known as 1-D and 2-D Bratu problems have been solved using the presented methods in Section 6. Section 7 includes a short conclusion.

### 2. Development of Algorithms

#### Efficient fifth-order algorithm:

The method proposed below is a two-step weighted iterative algorithm (known as  $5^{th}PJ$ ), which is based on algorithm (2):

$$\begin{aligned}
 G_{2^{nd}NM}(x^{(r)}) &= x^{(r)} - [\Phi'(x^{(r)})]^{-1}\Phi(x^{(r)}), \\
 x^{(r+1)} &= G_{5^{th}PJ}(x^{(r)}) = G_{2^{nd}NM}(x^{(r)}) - H_1(x^{(r)})[\Phi'(G_{2^{nd}NM}(x^{(r)}))]^{-1}\Phi(G_{2^{nd}NM}(x^{(r)})), \quad (7)
 \end{aligned}$$

where  $H_1(x^{(r)}) = I + \frac{1}{4}(\tau(x^{(r)}) - I)^2, \quad \tau(x^{(r)}) = [\Phi'(x^{(r)})]^{-1}\Phi'(G_{2^{nd}NM})$ ,

where  $I$  denotes  $n \times n$  identity matrix. In the above algorithm, a weight function  $H_1(x^{(r)})$  is suitably chosen to produce fifth-order convergence keeping same number of function evaluations as in method (2).

#### Efficient $(3k + 5)^{th}$ -order algorithm:

With the  $5^{th}PJ$  algorithm as the base, we introduce new steps to obtain the following new algorithm (known as  $(3k + 5)^{th}PJ$ ) by evaluating a new function whenever a new step is included

$$\begin{aligned}
 x^{(r+1)} &= G_{(3k+5)^{th}PJ}(x^{(r)}) = \mu_k(x^{(r)}), \\
 \mu_j(x^{(r)}) &= \mu_{j-1}(x^{(r)}) - H_2(x^{(r)})[\Phi'(G_{2^{nd}NM}(x^{(r)}))]^{-1}\Phi(\mu_{j-1}(x^{(r)})), \quad (8) \\
 \text{where } H_2(x^{(r)}) &= I + \frac{1}{2}(\tau(x^{(r)}) - I)^2, \\
 \mu_0(x^{(r)}) &= G_{5^{th}PJ}(x^{(r)}), \quad j = 1, 2, \dots, k, k \geq 1.
 \end{aligned}$$

The above algorithm has convergence order  $3k + 5$ . The case  $k = 0$  produces the  $5^{th}PJ$  algorithm.

### 3. Convergence Analysis

To prove the theoretical convergence, the usual Taylor’s expansion ( $n$ -dimensional) has been applied. Hence, we recall some important results from [3]:

Let  $\Phi : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$  be sufficiently Fréchet differentiable in  $D$ . Assume that  $i$ th derivative of  $\Phi$  at  $u \in \mathbb{R}^n, i \geq 1$ , is the  $i$ -linear function  $\Phi^{(i)}(u) : \mathbb{R}^n \times \dots \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  such that  $\Phi^{(i)}(u)(v_1, \dots, v_i) \in \mathbb{R}^n$ . Given  $\alpha + h \in \mathbb{R}^n$ , which lies near the solution  $\alpha$  of the system of nonlinear equations  $\Phi(x) = 0$ , one may apply Taylor’s expansion (whenever Jacobian matrix  $\Phi'(\alpha)$  is nonsingular) to get

$$\Phi(\alpha + h) = \Phi'(\alpha) \left[ h + \sum_{i=2}^{p-1} C_i h^i \right] + O(h^p), \quad (9)$$

where  $C_i = (1/i!)[\Phi'(\alpha)]^{-1}\Phi^{(i)}(\alpha)$ ,  $i \geq 2$ . It is noted that  $C_i h^i \in \mathbb{R}^n$  since  $\Phi^{(i)}(\alpha) \in \mathcal{L}(\mathbb{R}^n \times \dots \times \mathbb{R}^n, \mathbb{R}^n)$  and  $[\Phi'(\alpha)]^{-1} \in \mathcal{L}(\mathbb{R}^n)$ . Expanding  $\Phi'(\alpha + h)$  using Taylor's series, we get

$$\Phi'(\alpha + h) = \Phi'(\alpha) \left[ I + \sum_{i=2}^{p-1} iC_i h^{i-1} \right] + O(h^p), \tag{10}$$

where  $I$  denotes the identity matrix. We remark that  $iC_i h^{i-1} \in \mathcal{L}(\mathbb{R}^n)$ . The error is denoted as  $E^{(r)} = x^{(r)} - \alpha$  for the  $r$ th iteration. The equation  $E^{(r+1)} = LE^{(r)p} + O(E^{(r)p+1})$  is called the *error equation*, where  $L$  is a  $p$ -linear function  $L \in \mathcal{L}(\mathbb{R}^n \times \dots \times \mathbb{R}^n, \mathbb{R}^n)$  and  $p$  denotes *order of convergence*. Also,  $E^{(r)p} = (E_1^{(r)}, E_2^{(r)}, \dots, E_n^{(r)})^p$ .

By using the above concepts, the following theorems are established.

**Theorem 1.** Let  $\Phi : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$  be sufficiently Fréchet differentiable at each point of an open convex neighborhood  $D$  of  $\alpha \in \mathbb{R}^n$ . Choose  $x^{(0)}$  close enough to  $\alpha$ , where  $\alpha$  is a solution of the system  $\Phi(x) = 0$  and  $\Phi'(x)$  is continuous and nonsingular at  $\alpha$ . The iterative formula (7) producing the sequence  $\{x^{(r)}\}_{r \geq 0}$  converges to  $\alpha$  locally with order five. The error expression is found to be

$$E^{(r+1)} = G_{5th Pf}(x^{(r)}) - \alpha = L_1 E^{(r)5} + O(E^{(r)6}), \quad L_1 = 4C_2^4 - C_2 C_3 C_2.$$

**Proof.** Applying Taylor's formula around  $\alpha$  we get

$$\Phi(x^{(r)}) = \Phi'(\alpha) \left[ E^{(r)} + C_2 E^{(r)2} + C_3 E^{(r)3} + C_4 E^{(r)4} + C_5 E^{(r)5} + C_6 E^{(r)6} \right] + O(E^{(r)7})$$

and we express the differential of first order of  $\Phi(x^{(r)})$  as

$$\Phi'(x^{(r)}) = \Phi'(\alpha) \left[ I + 2C_2 E^{(r)} + 3C_3 E^{(r)2} + 4C_4 E^{(r)3} + 5C_5 E^{(r)4} + 6C_6 E^{(r)5} \right] + O(E^{(r)6}),$$

where  $C_i = (1/i!)[\Phi'(\alpha)]^{-1}\Phi^{(i)}(\alpha)$ ,  $i = 2, 3, \dots$  and  $E^{(r)} = x^{(r)} - \alpha$ . Inverting the above series, we get

$$[\Phi'(x^{(r)})]^{-1} = \left[ I + X_1 E^{(r)} + X_2 E^{(r)2} + X_3 E^{(r)3} + X_4 E^{(r)4} + X_5 E^{(r)5} \right] [\Phi'(\alpha)]^{-1}, \tag{11}$$

where

$$X_1 = -2C_2, \quad X_2 = 4C_2^2 - 3C_3,$$

$$X_3 = -8C_2^3 + 6C_2 C_3 + 6C_3 C_2 - 4C_4,$$

$$X_4 = -5C_5 + 9C_3^2 + 8C_2 C_4 + 8C_4 C_2 + 16C_2^4 - 12C_2^2 C_3 - 12C_3 C_2^2 - 12C_2 C_3 C_2$$

and

$$X_5 = -32C_2^5 + 24C_3 C_2^3 + 24C_2 C_3 C_2^2 - 16C_4 C_2^2 + 24C_2^2 C_3 C_2 - 18C_3^2 C_2 - 16C_2 C_4 C_2 + 10C_5 C_2 + 24C_2^3 C_3 - 18C_3 C_2 C_3 - 18C_2 C_3^2 + 12C_4 C_3 - 16C_2^2 C_4 + 12C_3 C_4 + 10C_2 C_4 - 6C_6.$$

Then

$$\begin{aligned} [\Phi'(x^{(r)})]^{-1}\Phi(x^{(r)}) &= E^{(r)} - C_2 E^{(r)2} + 2(C_2^2 - C_3)E^{(r)3} + (-3C_4 - 4C_2^3 + 4C_2 C_3 + 3C_3 C_2)E^{(r)4} \\ &+ (6C_3^2 + 8C_2^4 - 8C_2^2 C_3 - 6C_2 C_3 C_2 - 6C_3 C_2^2 + 6C_2 C_4 + 4C_4 C_2 - 4C_5)E^{(r)5} + (-5C_6 - 2C_2 C_5 - 14C_2^2 C_4 \\ &+ 9C_3 C_4 + 16C_2^3 C_3 - 12C_3 C_2 C_3 - 12C_2 C_3^2 + 8C_4 C_3 - 16C_2^5 + 12C_3 C_2^3 + 12C_2 C_3 C_2^2 - 8C_4 C_2^2 \\ &+ 12C_2^2 C_3 C_2 - 9C_3^2 C_2 - 8C_2 C_4 C_2 + 5C_5 C_2 + 10C_2 C_4)E^{(r)6} + O(E^{(r)7}). \end{aligned}$$

Furthermore, we obtain

$$\begin{aligned}
 G_{2^{nd}NM}(x^{(r)}) &= \alpha + C_2 E^{(r)2} + 2(-C_2^2 + C_3) E^{(r)3} + (3C_4 + 4C_2^3 - 4C_2C_3 - 3C_3C_2) E^{(r)4} + (-6C_3^2 - 8C_2^4 \\
 &+ 8C_2^2C_3 + 6C_2C_3C_2 + 6C_3C_2^2 - 6C_2C_4 - 4C_4C_2 + 4C_5) E^{(r)5} + (5C_6 + 2C_2C_5 + 14C_2^2C_4 - 9C_3C_4 - 16C_2^3C_3 \\
 &+ 12C_3C_2C_3 + 12C_2C_3^2 - 8C_4C_3 + 16C_2^5 - 12C_3C_2^3 - 12C_2C_3C_2^2 + 8C_4C_2^2 - 12C_2^2C_3C_2 + 9C_3^2C_2 \\
 &+ 8C_2C_4C_2 - 5C_5C_2 - 10C_2C_4) E^{(r)6} + O(E^{(r)7}).
 \end{aligned} \tag{12}$$

Equation (12) leads to

$$\begin{aligned}
 \Phi(G_{2^{nd}NM}(x^{(r)})) &= \Phi'(\alpha) \left[ C_2 E^{(r)2} + 2(-C_2^2 + C_3) E^{(r)3} + (3C_4 + 5C_2^3 - 4C_2C_3 - 3C_3C_2) E^{(r)4} \right. \\
 &+ (-6C_3^2 - 12C_2^4 + 12C_2^2C_3 + 6C_2C_3C_2 + 6C_3C_2^2 - 6C_2C_4 - 4C_4C_2 + 4C_5) E^{(r)5} \\
 &+ (5C_6 + 2C_2C_5 + 14C_2^2C_4 - 9C_3C_4 - 16C_2^3C_3 + 12C_3C_2C_3 + 12C_2C_3^2 - 8C_4C_3 + 16C_2^5 \\
 &\left. - 12C_3C_2^3 - 12C_2C_3C_2^2 + 8C_4C_2^2 - 12C_2^2C_3C_2 + 9C_3^2C_2 + 8C_2C_4C_2 - 5C_5C_2 - 10C_2C_4) E^{(r)6} \right] + O(E^{(r)7}).
 \end{aligned} \tag{13}$$

Also,

$$\Phi'(G_{2^{nd}NM}(x^{(r)})) = [\Phi'(\alpha)] \left[ I + P_1 E^{(r)2} + P_2 E^{(r)3} + P_3 E^{(r)4} \right] + O(E^{(r)5}), \tag{14}$$

where  $P_1 = 2C_2^2$ ,  $P_2 = 4C_2C_3 - 4C_2^3$  and  $P_3 = 4C_2^3 - 4C_2C_3 - 3C_3C_2 + 3C_4 + 3C_3C_2^2$ .

Combining (11) and (14), one gets

$$\begin{aligned}
 \tau(x^{(r)}) &= [\Phi'(x^{(r)})]^{-1} \Phi'(G_{2^{nd}NM}(x^{(r)})) = I - 2C_2 E^{(r)} + (6C_2^2 - 3C_3) E^{(r)2} + (10C_2C_3 \\
 &+ 6C_3C_2 - 16C_2^3 - 4C_4) E^{(r)3} + (4C_2^3 - 4C_2C_3 - 3C_3C_2 + 3C_4 - 15C_3C_2^2 - 20C_2^2C_3 + 32C_2^4 - 5C_5 \\
 &+ 9C_3^2 + 8C_2C_4 + 8C_4C_2 - 12C_2C_3C_2) E^{(r)4} + O(E^{(r)5}).
 \end{aligned} \tag{15}$$

Then we get

$$H_1(x^{(r)}) = I + C_2^2 E^{(r)2} + (3C_2C_3 - 6C_2^3) E^{(r)3} + (25C_2^4 - 19C_2^2C_3 + \frac{9}{4}C_3^2 - 6C_2C_3C_2 + 4C_2C_4) E^{(r)4} + O(E^{(r)5}). \tag{16}$$

After calculating inverse of  $\Phi'(G_{2^{nd}NM}(x^{(r)}))$  and using (13), we get

$$\begin{aligned}
 [\Phi'(G_{2^{nd}NM}(x^{(r)}))]^{-1} \Phi(G_{2^{nd}NM}(x^{(r)})) &= (C_2 E^{(r)2} + 2(C_3 - C_2^2) E^{(r)3} + (3C_4 + 3C_2^3 - 3C_3C_2 - 4C_2C_3) E^{(r)4} \\
 &+ (4C_5 - 6C_2C_4 - 4C_4C_2 + 6C_2^2C_3 + 4C_2C_3C_2 + 6C_3C_2^2 - 6C_2^3 - 4C_2^4)) + O(E^{(r)5}).
 \end{aligned} \tag{17}$$

We get the required error estimate by using (12), (16) and (17) in (7)

$$E^{(r+1)} = G_{5^{th}PJ}(x^{(r)}) - \alpha = (4C_2^4 - C_2C_3C_2) E^{(r)5} + O(E^{(r)6}).$$

The above result agrees with fifth-order convergence.  $\square$

**Theorem 2.** Let  $\Phi : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$  be sufficiently Fréchet differentiable at each point of an open convex neighborhood  $D$  of  $\alpha \in \mathbb{R}^n$ . Choose  $x^{(0)}$  close enough to  $\alpha$ , where  $\alpha$  is a solution of the system  $\Phi(x) = 0$  and  $\Phi'(x)$  is continuous and nonsingular at  $\alpha$ . The iterative formula (8) producing the sequence  $\{x^{(r)}\}$  converges to  $\alpha$  locally with order  $3k + 5$ , where  $k$  is a positive integer and  $k \geq 1$ .

**Proof.** Taylor’s expansion of  $\Phi(\mu_{j-1}(x^{(r)}))$  around  $\alpha$  gives

$$\Phi(\mu_{j-1}(x^{(r)})) = \Phi'(\alpha) \left[ (\mu_{j-1}(x^{(r)}) - \alpha) + C_2(\mu_{j-1}(x^{(r)}) - \alpha)^2 + \dots \right] \tag{18}$$

By expanding

$$H_2(x^{(r)}) = I + \frac{1}{2}(\eta(x^{(r)}) - I)^2 = I + 2C_2^2 E^{(r)2} + (6C_2C_3 - 12C_2^3)E^{(r)3} + \dots \tag{19}$$

After evaluating  $[\Phi'(G_{2ndNM}(x^{(r)}))]^{-1}$  and combining with (19) yields

$$H_2(x^{(r)})[\Phi'(G_{2ndNM}(x^{(r)}))]^{-1} = [I + L_2 E^{(r)3} + \dots] [\Phi'(\alpha)]^{-1}, \quad L_2 = 2C_2C_3 - 8C_2^3. \tag{20}$$

Equations (18) and (20) leads to

$$\begin{aligned} H_2(x^{(r)})[\Phi'(G_{2ndNM}(x^{(r)}))]^{-1}\Phi(\mu_{j-1}(x^{(r)})) &= (I + L_2 E^{(r)3} + \dots) ((\mu_{j-1}(x^{(r)}) - \alpha) + C_2(\mu_{j-1}(x^{(r)}) - \alpha)^2 + \dots) \\ &= [\mu_{j-1}(x^{(r)}) - \alpha + L_2 E^{(r)3}(\mu_{j-1}(x^{(r)}) - \alpha) + C_2(\mu_{j-1}(x^{(r)}) - \alpha)^2 + \dots] \end{aligned} \tag{21}$$

Substituting (21) in (8) and calculating the error term, we get

$$\begin{aligned} \mu_j(x^{(r)}) - \alpha &= (\mu_{j-1}(x^{(r)}) - \alpha) - ((\mu_{j-1}(x^{(r)}) - \alpha) + L_2 E^{(r)3}(\mu_{j-1}(x^{(r)}) - \alpha) \\ &\quad + C_2(\mu_{j-1}(x^{(r)}) - \alpha)^2 + \dots) \\ &= -L_2 E^{(r)3}(\mu_{j-1}(x^{(r)}) - \alpha) + \dots \end{aligned} \tag{22}$$

It is proved that  $\mu_0(x^{(r)}) - \alpha = L_1 E^{(r)5} + O(E^{(r)6})$ . Therefore, for  $j = 1, 2, \dots$  in (22), we get

$$\begin{aligned} \mu_1(x^{(r)}) - \alpha &= -L_2(E^{(r)3}) (\mu_0(x^{(r)}) - \alpha) + \dots \\ &= -L_2L_1E^{(r)8} + \dots \\ \mu_2(x^{(r)}) - \alpha &= -L_2(E^{(r)3}) (\mu_1(x^{(r)}) - \alpha) + \dots \\ &= -L_2(-L_2L_1)E^{(r)11} + \dots \\ &= L_2^2L_1E^{(r)11} + \dots \end{aligned}$$

Proceeding by induction, we get

$$\mu_k(x^{(r)}) - \alpha = (-L_2)^k L_1 (E^{(r)(3k+5)}) + O(E^{(r)(3k+6)}), \quad k \geq 1.$$

The above result shows that the method has  $(3k + 5)$  order convergence.  $\square$

#### 4. Computational Efficiency of the Algorithms

The efficiency index of any iterative method is measured using the Ostrowski’s definition [16],  $EI = p^{\frac{1}{d}}$ , where  $p$  denotes the order of convergence and  $d$  denotes the number of functional evaluations per iteration. Different algorithms considered here are compared with the proposed algorithms in terms of computational cost. To evaluate Jacobian  $\Phi'$  and  $\Phi$ , one needs  $n^2$  function evaluations (all the elements of matrix  $\Phi'$ ) and  $n$  scalar functional evaluations (the coordinate terms in  $\Phi$ ) respectively. Whereas, any iterative method which solves system of nonlinear equations needs one or more matrix inversion. This indicates few linear systems must be solved. Hence, the number of operations required for solving a linear system of equations dominates when the computational cost of an iterative method is measured. Due to this, Cordero et al. [3] introduced the concept of computational efficiency index (CE), where the efficiency index given by Ostrowski is combined with the number of products-quotients required per iteration. Computational efficiency index is defined as  $CE = p^{1/(d+op)}$ ,

where  $op$  is the number of products-quotients per iteration and the details of its calculation is given in the next paragraph.

The total cost of computation for a nonlinear system of  $n$  equations with  $n$  unknowns is calculated as follows: For an iterative function,  $n$  scalar functions  $\phi_i$  ( $1 \leq i \leq n$ ) are evaluated and for the computation of divided difference  $n(n - 1)$  scalar functions are evaluated, where  $\Phi(x)$  and  $\Phi(y)$  are computed separately. Also,  $n^2$  quotients are added for divided difference. For calculating inverse linear operator,  $\frac{n^3-n}{3}$  products and divisions in the LU decomposition and  $n^2$  products and divisions for solving two triangular linear systems are taken. Moreover,  $n^2$  products for multiplying a matrix with a vector or a scalar and  $n$  products for multiplying a vector by a scalar are required.

The computational cost and efficiency of the proposed algorithms are compared with methods given by (1) to (6) and the algorithms presented recently by Sharma et al. [17] which are given below: A fifth-order three-step method (5<sup>th</sup>SD) is given by

$$\begin{aligned}
 x^{(r+1)} &= G_{5^{th}SD}(x^{(r)}) = z(x^{(r)}) - \psi(x_n, y_n)\Phi(z^{(r)}), \\
 \text{where } z(x^{(r)}) &= G_{2^{nd}NM}(x^{(r)}) - \Phi'(x^{(r)})^{-1}\Phi(G_{2^{nd}NM}(x^{(r)})), \\
 \psi(x_n, y_n) &= \left(2I - \Phi'(x^{(r)})^{-1}[z_r, y_r; \Phi]\right)\Phi'(x^{(r)})^{-1}.
 \end{aligned}$$

An eighth-order four-step method (8<sup>th</sup>SD) is given by

$$\begin{aligned}
 x^{(r+1)} &= G_{8^{th}SD}(x^{(r)}) = w(x^{(r)}) - \psi(x_n, y_n)\Phi(w^{(r)}), \\
 \text{where } w(x^{(r)}) &= z(x^{(r)}) - \psi(x_n, y_n)\Phi(z^{(r)}), \\
 z(x^{(r)}) &= G_{2^{nd}NM}(x^{(r)}) - \left(3I - 2\Phi'(x^{(r)})^{-1}[y_r, x_r; \Phi]\right), \\
 \psi(x_n, y_n) &= \left(2I - \Phi'(x^{(r)})^{-1}[z_r, y_r; \Phi]\right)\Phi'(x^{(r)})^{-1}.
 \end{aligned}$$

Table 1 displays the computational cost and computational efficiency (CE) of various methods.

**Table 1.** Table of Computational Cost and Computational Efficiency.

Method	Computational Cost ( $C_{method}$ )	Computational Efficiency
2 <sup>nd</sup> NM	$\frac{1}{3}n^3 + 2n^2 + \frac{2}{3}n$	$2 \frac{1}{C_{2^{nd}NM}}$
4 <sup>th</sup> NR	$\frac{2}{3}n^3 + 4n^2 + \frac{4}{3}n$	$4 \frac{1}{C_{4^{th}NR}}$
5 <sup>th</sup> ACT	$\frac{2}{3}n^3 + 5n^2 + \frac{7}{3}n$	$5 \frac{1}{C_{5^{th}ACT}}$
5 <sup>th</sup> MBJ	$\frac{1}{3}n^3 + 7n^2 + \frac{11}{3}n$	$5 \frac{1}{C_{5^{th}MBJ}}$
5 <sup>th</sup> SD	$\frac{1}{3}n^3 + 8n^2 + \frac{8}{3}n$	$5 \frac{1}{C_{5^{th}SD}}$
5 <sup>th</sup> PJ	$\frac{2}{3}n^3 + 8n^2 + \frac{7}{3}n$	$5 \frac{1}{C_{5^{th}PJ}}$
8 <sup>th</sup> SA	$\frac{1}{3}n^3 + 10n^2 + \frac{23}{3}n$	$8 \frac{1}{C_{8^{th}SA}}$
8 <sup>th</sup> MBJ	$\frac{1}{3}n^3 + 9n^2 + \frac{17}{3}n$	$8 \frac{1}{C_{8^{th}MBJ}}$
8 <sup>th</sup> SD	$\frac{1}{3}n^3 + 15n^2 + \frac{17}{3}n$	$8 \frac{1}{C_{8^{th}SD}}$
8 <sup>th</sup> PJ	$\frac{2}{3}n^3 + 10n^2 + \frac{13}{3}n$	$8 \frac{1}{C_{8^{th}PJ}}$

To compare the CE of considered iterative methods, we calculate the following ratio [17], where  $C_{method}$  and  $CE_{method}$  denote respectively the computational cost and computational efficiency of the method.

$$R_{method1;method2} = \frac{\log(CE_{method1})}{\log(CE_{method2})} = \frac{C_{method2} \log(\text{order of method1})}{C_{method1} \log(\text{order of method2})}. \tag{23}$$

It is clear that when  $R_{method1,method2} > 1$ , the iterative method 1 is more efficient than method 2.  $5^{th}PJ$  versus  $2^{nd}NM$ : The ratio (23) is given by

$$R_{5^{th}PJ;2^{nd}NM} = \frac{(\frac{1}{3}n^3 + 2n^2 + \frac{2}{3}n) \log(5)}{(\frac{2}{3}n^3 + 8n^2 + \frac{7}{3}n) \log(2)}.$$

Based on the computation, we have  $R_{5^{th}PJ;2^{nd}NM} > 1$  for  $n \geq 32$ . Thus, we conclude that  $CE_{5^{th}PJ} > CE_{2^{nd}NM}$  for  $n \geq 32$ .

$5^{th}PJ$  versus  $4^{th}NR$ : In this case, the ratio (23) is given by

$$R_{5^{th}PJ;4^{th}NR} = \frac{(\frac{2}{3}n^3 + 4n^2 + \frac{4}{3}n) \log(5)}{(\frac{2}{3}n^3 + 8n^2 + \frac{7}{3}n) \log(4)}.$$

It has been verified that  $R_{5^{th}PJ;4^{th}NR} > 1$  for  $n \geq 32$ . Hence, we have  $CE_{5^{th}PJ} > CE_{4^{th}NR}$  for  $n \geq 32$ .

$5^{th}PJ$  versus  $8^{th}SA$ : Here the ratio (23) is given by

$$R_{5^{th}PJ;8^{th}SA} = \frac{(\frac{1}{3}n^3 + 10n^2 + \frac{23}{3}n) \log(5)}{(\frac{2}{3}n^3 + 8n^2 + \frac{7}{3}n) \log(8)}.$$

It can be checked that  $R_{5^{th}PJ;8^{th}SA} > 1$  for  $n = 2$  which implies that  $CE_{5^{th}PJ} > CE_{8^{th}SA}$  for  $n = 2$ .

$5^{th}PJ$  versus  $8^{th}SD$ : Here the ratio (23) is given by

$$R_{5^{th}PJ;8^{th}SD} = \frac{(\frac{1}{3}n^3 + 15n^2 + \frac{17}{3}n) \log(5)}{(\frac{2}{3}n^3 + 8n^2 + \frac{7}{3}n) \log(8)}.$$

It has been verified that  $R_{5^{th}PJ;8^{th}SD} > 1$  for  $2 \leq n \leq 15$ . Hence, we have  $CE_{5^{th}PJ} > CE_{8^{th}SD}$  for  $2 \leq n \leq 15$ .

$8^{th}PJ$  versus  $2^{nd}NM$ : The ratio (23) is given by

$$R_{8^{th}PJ;2^{nd}NM} = \frac{(\frac{1}{3}n^3 + 2n^2 + \frac{2}{3}n) \log(8)}{(\frac{2}{3}n^3 + 10n^2 + \frac{13}{3}n) \log(2)}.$$

It can be checked that  $R_{8^{th}PJ;2^{nd}NM} > 1$  for  $n \geq 13$ . Thus, we conclude that  $CE_{8^{th}PJ} > CE_{2^{nd}NM}$  for  $n \geq 13$ .

$8^{th}PJ$  versus  $4^{th}NR$ : The ratio (23) is given by

$$R_{8^{th}PJ;4^{th}NR} = \frac{(\frac{2}{3}n^3 + 4n^2 + \frac{4}{3}n) \log(8)}{(\frac{2}{3}n^3 + 10n^2 + \frac{13}{3}n) \log(4)}.$$

It has been verified that  $R_{8^{th}PJ;4^{th}NR} > 1$  for  $n \geq 13$ . Hence, we conclude that  $CE_{8^{th}PJ} > CE_{4^{th}NR}$  for  $n \geq 13$ .

$8^{th}PJ$  versus  $5^{th}ACT$ : The ratio (23) is given by

$$R_{8^{th}PJ;5^{th}ACT} = \frac{(\frac{2}{3}n^3 + 5n^2 + \frac{7}{3}n) \log(8)}{(\frac{2}{3}n^3 + 10n^2 + \frac{13}{3}n) \log(5)}.$$

Based on the computation, we have  $R_{8^{th}PJ;5^{th}ACT} > 1$  for  $n \geq 19$ . Thus, we conclude that  $CE_{8^{th}PJ} > CE_{5^{th}ACT}$  for  $n \geq 19$ .

$8^{th}PJ$  versus  $5^{th}PJ$ : The ratio (23) is given by

$$R_{8^{th}PJ;5^{th}PJ} = \frac{(\frac{2}{3}n^3 + 8n^2 + \frac{7}{3}n) \log(8)}{(\frac{2}{3}n^3 + 10n^2 + \frac{13}{3}n) \log(5)}.$$

It can be checked that  $R_{8^{th}PJ;5^{th}PJ} > 1$  for  $n \geq 2$ , which implies that  $E_{8^{th}PJ} > E_{5^{th}PJ}$  for  $n \geq 2$ .

8<sup>th</sup> PJ versus 8<sup>th</sup> SA: The ratio (23) is given by

$$R_{8^{th}PJ;8^{th}SA} = \frac{(\frac{1}{3}n^3 + 10n^2 + \frac{23}{3}n) \log(8)}{(\frac{2}{3}n^3 + 10n^2 + \frac{13}{3}n) \log(8)}.$$

It has been verified that  $R_{8^{th}PJ;8^{th}SA} > 1$  for  $n \geq 2$  and 3. Hence, we conclude that  $CE_{8^{th}PJ} > CE_{8^{th}SA}$  for  $n \geq 2$  and 3.

8<sup>th</sup> PJ versus 8<sup>th</sup> SD: The ratio (23) is given by

$$R_{8^{th}PJ;8^{th}SD} = \frac{(\frac{1}{3}n^3 + 15n^2 + \frac{17}{3}n) \log(8)}{(\frac{2}{3}n^3 + 10n^2 + \frac{13}{3}n) \log(8)}.$$

Based on the computation, we have  $R_{8^{th}PJ;8^{th}SD} > 1$  for  $2 \leq n \leq 15$ , which implies that  $CE_{8^{th}PJ} > CE_{8^{th}SD}$  for  $2 \leq n \leq 15$ .

Consolidating the above ratios, the following theorem is stated to show the superiority of the proposed methods.

**Theorem 3.** Computational efficiency of 5<sup>th</sup> PJ and 8<sup>th</sup> PJ methods satisfy: (a)  $CE_{5^{th}PJ} > CE_{2^{nd}NM}, CE_{4^{th}NR}, CE_{8^{th}SA}$  and  $CE_{8^{th}SD}$  for  $n \geq 32, n \geq 32, n = 2$  and  $2 \leq n \leq 9$  respectively. (b)  $CE_{8^{th}PJ} > CE_{2^{nd}NM}, CE_{4^{th}NR}, CE_{5^{th}ACT}, CE_{5^{th}PJ}, CE_{8^{th}SA}$  and  $CE_{8^{th}SD}$  for  $n \geq 13, n \geq 13, n \geq 19, n \geq 2, n \geq 2$  and 3, and  $2 \leq n \leq 15$  respectively.

**Remark 1.** The ratio between the proposed method 5<sup>th</sup> PJ respectively with 5<sup>th</sup> ACT, 5<sup>th</sup> MBJ, 8<sup>th</sup> MBJ, 5<sup>th</sup> SD and the proposed method 8<sup>th</sup> PJ respectively with 5<sup>th</sup> MBJ, 8<sup>th</sup> MBJ, 5<sup>th</sup> SD do not satisfy the required condition  $R_{method1;method2} > 1$ .

### 5. Numerical Results

Numerical performance of the presented methods is compared with Newton’s method and some known methods given in the beginning of this paper. All the numerical calculations are done using MATLAB package for the test examples given below. We have taken 500 digits accuracy while calculating the approximate numerical solutions. The following error residual is adopted to stop the iteration:

$$err_{min} = \|x^{(r+1)} - x^{(r)}\|_2 < 10^{-100}.$$

The following formula is used to find approximated computational order of convergence  $p_c$ :

$$p_c \approx \frac{\log(\|x^{(r+1)} - x^{(r)}\|_2 / \|x^{(r)} - x^{(r-1)}\|_2)}{\log(\|x^{(r)} - x^{(r-1)}\|_2 / \|x^{(r-1)} - x^{(r-2)}\|_2)}.$$

Here N denotes the number of iterations needed for obtaining the minimum residual ( $err_{min}$ ),  $n_{inv}$  represents the total number of Frechet derivative’s inverse and  $n_{total}$  represents the total number of function evaluations ( $\Phi$  and  $\Phi'$ ) to reach the minimum residual as in [7].

**Test Example 1 (TE1):**

$$\Phi(y_1, y_2) = (y_1 + exp(y_2) - cos(y_2), 3y_1 - y_2 - sin(y_2)).$$

The Jacobian matrix is given by  $\Phi'(y) = \begin{pmatrix} 1 & exp(y_2) + sin(y_2) \\ 3 & -1 - cos(y_2) \end{pmatrix}$ . Initial approximation is taken as  $y^{(0)} = (1.5, 2)^T$  and the analytic solution is given by  $\alpha = (0, 0)^T$ .

**Test Example 2 (TE2):**

$$\begin{cases} y_2 y_3 + y_4(y_2 + y_3) = 0, \\ y_1 y_3 + y_4(y_1 + y_3) = 0, \\ y_1 y_2 + y_4(y_1 + y_2) = 0, \\ y_1 y_2 + y_1 y_3 + y_2 y_3 = 1. \end{cases}$$

The above system is solved by taking the starting approximation  $y^{(0)} = (0.5, 0.5, 0.5, -0.2)^T$ .

The solution is given by  $\alpha \approx (0.577350, 0.577350, 0.577350, -0.288675)^T$ . The Jacobian matrix is given by

$$\Phi'(y) = \begin{pmatrix} 0 & y_3 + y_4 & y_2 + y_4 & y_2 + y_3 \\ y_3 + y_4 & 0 & y_1 + y_4 & y_1 + y_3 \\ y_2 + y_4 & y_1 + y_4 & 0 & y_1 + y_2 \\ y_2 + y_3 & y_1 + y_3 & y_1 + y_2 & 0 \end{pmatrix}.$$

**Test Example 3 (TE3):**

$$\begin{cases} \cos y_2 - \sin y_1 = 0, \\ y_3^{y_1} - \frac{1}{y_2} = 0, \\ \exp y_1 - y_3^2 = 0. \end{cases}$$

The solution for the above system is  $\alpha \approx (0.909569, 0.661227, 1.575834)^T$ . The initial vector for the iteration is taken as  $x^{(0)} = (1, 0.5, 1.5)^T$ . The Jacobian matrix produced thus is given by

$$\Phi'(y) = \begin{pmatrix} -\cos y_1 & -\sin y_2 & 0 \\ y_3^{y_1} \ln y_3 & 1/y_2^2 & y_3^{y_1} y_1/y_3 \\ \exp y_1 & 0 & -2y_3 \end{pmatrix}.$$

**Test Example 4 (TE4):** The following boundary value problem is considered

$$y'' + y^3 = 0, \quad y(0) = 0, \quad y(1) = 1,$$

where equal mesh is used for dividing the interval  $[0, 1]$  which is given below

$$u_0 = 0 < u_1 < u_2 < \dots < u_{m-1} < u_m = 1, \quad u_{j+1} = u_j + h, \quad h = 1/m.$$

Denote  $y_0 = y(u_0) = 0, y_1 = y(u_1), \dots, y_{m-1} = y(u_{m-1}), y_m = y(u_m) = 1$ .

Discretizing the second derivative by the following difference formula

$$y'' \approx \frac{y_{r-1} - 2y_r + y_{r+1}}{h^2}, \quad r = 1, 2, 3, \dots, m - 1,$$

we obtain  $m - 1$  nonlinear equations in  $m - 1$  variables as given below

$$y_{r-1} - 2y_r + y_{r+1} + h^2 y_r^3 = 0, \quad r = 1, 2, 3, \dots, m - 1.$$

The above equations are solved by taking  $m = 16$  (i.e.,  $n = 15$ ) and  $y^{(0)} = (1, 1, \dots, 1)^T$  as the initial approximation, where we get the Jacobian matrix with 43 non-zero elements as below.

$$\begin{pmatrix} 3h^2y_1^2 - 2 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 3h^2y_2^2 - 2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 3h^2y_3^2 - 2 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & & & & & & \\ \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & \dots & 1 & 3h^2y_{14}^2 - 2 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 3h^2y_{15}^2 - 2 \end{pmatrix}.$$

The following solution is obtained for the differential equation at the given mesh points.

$$\alpha = \{0.065997633200364677\dots, 0.131994143490292748\dots, 0.197981670725993839\dots, \\ 0.263938884538034848\dots, 0.329824274254574844\dots, 0.395569509201723100\dots, \\ 0.461072959646730428\dots, 0.526193524526372529\dots, 0.590744978992414345\dots, \\ 0.654491128910354268\dots, 0.717142134576548678\dots, 0.778352432953974123\dots, \\ 0.837720734425024994\dots, 0.894792581480763658\dots, 0.949065916629282713\dots\}^T.$$

**Test Example 5 (TE5):** The following huge nonlinear system is considered

$$\begin{cases} y_i y_{i+1} - 1 = 0, & i = 1, 2, 3, \dots, 15, \\ y_{15} y_1 - 1 = 0. \end{cases}$$

The solution is  $\alpha = (1, 1, 1, \dots, 1)^T$ . Choosing the initial vector as  $y^{(0)} = (1.5, 1.5, 1.5, \dots, 1.5)^T$ , we obtain the following Jacobian matrix.

$$\Phi'(y) = \begin{pmatrix} y_2 & y_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & y_3 & y_2 & 0 & \dots & 0 & 0 \\ \vdots & & & & & & \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \dots & y_{15} & y_{14} \\ y_{15} & 0 & 0 & 0 & \dots & 0 & y_1 \end{pmatrix}.$$

The numerical results are displayed in Table 2 for the test examples TE1-TE5 by which we arrive at 8<sup>th</sup> PJ algorithm is the most efficient one with least residual error and less CPU time. Also, it is observed from the results, the proposed algorithms require fewer iterations than the compared methods for the test example TE3. However, when comparing number of functional evaluations ( $n_{total}$ ) and inverse evaluations ( $n_{inv}$ ), 8<sup>th</sup> PJ algorithm is the best among the compared methods.

**Table 2.** Comparison of results for test examples.

TE	Methods	$N$	$p_c$	$CPU_T$	$err_{min}$	$\Phi$	$\Phi'$	$n_{inv}$	$n_{total}$
<b>TE1</b>	2 <sup>nd</sup> NM	10	1.99	2.504327	$1.038 \times 10^{-103}$	10	10	10	40
	4 <sup>th</sup> NR	6	3.99	1.957957	$5.384 \times 10^{-207}$	12	12	12	48
	5 <sup>th</sup> ACT	6	4.99	2.397614	$2.280 \times 10^{-289}$	18	12	12	60
	5 <sup>th</sup> MBJ	6	4.99	2.241982	$1.095 \times 10^{-315}$	12	12	6	48
	5 <sup>th</sup> PJ	6	4.99	2.237277	0	12	12	12	48
	8 <sup>th</sup> SA	5	7.79	2.218581	0	15	10	5	50
	8 <sup>th</sup> MBJ	5	7.79	2.092063	0	15	10	5	50
	8 <sup>th</sup> PJ	5	7.90	2.074542	0	15	10	10	50
	11 <sup>th</sup> PJ	4	10.95	2.060324	$4.362 \times 10^{-154}$	16	8	8	48
<b>TE2</b>	2 <sup>nd</sup> NM	8	2.00	3.023825	$3.928 \times 10^{-145}$	8	8	8	128
	4 <sup>th</sup> NR	5	4.03	2.589849	$2.988 \times 10^{-291}$	10	10	10	160
	5 <sup>th</sup> ACT	4	5.15	2.916807	$5.083 \times 10^{-102}$	12	8	8	144
	5 <sup>th</sup> MBJ	4	5.15	2.840291	$5.083 \times 10^{-102}$	8	8	4	128
	5 <sup>th</sup> PJ	4	5.12	2.790838	$5.714 \times 10^{-121}$	8	8	8	128
	8 <sup>th</sup> SA	4	8.01	3.331801	0	12	8	4	144
	8 <sup>th</sup> MBJ	4	8.80	3.082510	0	12	8	4	144
	8 <sup>th</sup> PJ	4	8.60	3.071780	0	12	8	8	144
	11 <sup>th</sup> PJ	3	11.78	3.089214	$9.138 \times 10^{-106}$	12	6	6	120
<b>TE3</b>	2 <sup>nd</sup> NM	9	2.00	2.907378	$1.010 \times 10^{-107}$	9	9	9	90
	4 <sup>th</sup> NR	5	4.00	2.401273	$1.010 \times 10^{-107}$	10	10	10	100
	5 <sup>th</sup> ACT	6	5.00	3.457730	0	18	12	12	138
	5 <sup>th</sup> MBJ	5	3.92	3.086405	$7.523 \times 10^{-106}$	10	10	5	100
	5 <sup>th</sup> PJ	5	3.92	3.017444	$2.109 \times 10^{-143}$	10	10	10	100
	8 <sup>th</sup> SA	5	6.02	3.776025	0	15	10	5	115
	8 <sup>th</sup> MBJ	5	6.02	3.327364	0	15	10	5	115
	8 <sup>th</sup> PJ	4	5.86	2.945875	$1.938 \times 10^{-104}$	12	8	8	92
	11 <sup>th</sup> PJ	4	8.09	3.231956	$4.484 \times 10^{-228}$	16	8	8	104
<b>TE4</b>	2 <sup>nd</sup> NM	8	2.00	7.892696	$4.963 \times 10^{-114}$	8	8	8	240
	4 <sup>th</sup> NR	5	3.99	7.841182	$1.410 \times 10^{-228}$	10	10	10	300
	5 <sup>th</sup> ACT	5	4.05	11.754446	$2.580 \times 10^{-195}$	15	10	10	375
	5 <sup>th</sup> MBJ	5	4.02	14.253392	$1.001 \times 10^{-215}$	10	10	5	300
	5 <sup>th</sup> PJ	5	4.02	12.536327	$1.030 \times 10^{-253}$	10	10	10	300
	8 <sup>th</sup> SA	4	5.90	15.208284	$4.511 \times 10^{-155}$	12	8	4	300
	8 <sup>th</sup> MBJ	4	5.90	14.127740	$4.511 \times 10^{-155}$	12	8	4	300
	8 <sup>th</sup> PJ	4	5.93	14.014342	$1.533 \times 10^{-193}$	12	8	8	300
	11 <sup>th</sup> PJ	4	8.65	15.638620	0	16	8	8	360
<b>TE5</b>	2 <sup>nd</sup> NM	9	1.99	6.273565	$8.969 \times 10^{-179}$	9	9	9	405
	4 <sup>th</sup> NR	5	4.00	5.562079	$8.969 \times 10^{-179}$	10	10	10	450
	5 <sup>th</sup> ACT	5	5.00	7.186739	$1.399 \times 10^{-304}$	15	10	10	525
	5 <sup>th</sup> MBJ	5	5.00	8.735878	$1.399 \times 10^{-304}$	10	10	5	450
	5 <sup>th</sup> PJ	5	4.99	7.749309	0	10	10	10	450
	8 <sup>th</sup> SA	4	7.99	8.651872	$3.680 \times 10^{-226}$	12	8	4	420
	8 <sup>th</sup> MBJ	4	7.99	8.002011	$3.680 \times 10^{-226}$	12	8	4	420
	8 <sup>th</sup> PJ	4	7.99	8.069185	$1.358 \times 10^{-272}$	12	8	8	420
	11 <sup>th</sup> PJ	4	10.68	8.213288	0	16	8	8	480

## 6. Applications

### 6.1. Bratu Problem in One Dimension

Consider the one-dimensional Bratu problem stated as [18]

$$\frac{d^2U}{dx^2} + \lambda \exp U(x) = 0, \lambda > 0, 0 < x < 1, \tag{24}$$

subject to the conditions  $U(0) = U(1) = 0$ . The two known bifurcated exact solutions of the Planar Bratu problem are obtained for the values of  $\lambda < \lambda_c$ , one solution for  $\lambda = \lambda_c$  and for  $\lambda > \lambda_c$  solution does not exist. Here  $\lambda_c$  is known as critical value and its value is  $8(\eta^2 - 1)$ , where  $\eta$  denotes the fixed point of  $\coth(x)$ . The analytic solution for Equation (24) is found to be

$$U(x) = -2 \log_e \left[ \frac{\cosh \left( x - \frac{1}{2} \right) \frac{\theta}{2}}{\cosh \left( \frac{\theta}{4} \right)} \right]. \tag{25}$$

The constant  $\theta$  has to be found such that it satisfies the boundary conditions. The critical value of  $\lambda$  is obtained by using a similar procedure found in [19]. Substitute Equation (25) in (24) and convert it by using collocation at  $x = \frac{1}{2}$  as it is the middle point of  $[0, 1]$ . Instead, one can choose a different value for  $x$ , although it may give higher-order approximation but collocating at  $x = \frac{1}{2}$  will give equal distribution of the interval throughout the region and hence we get

$$\theta^2 = 2\lambda \cosh^2 \left( \frac{\theta}{4} \right). \tag{26}$$

Differentiate the above Equation (26) with respect to  $\theta$  and by taking  $\frac{d\lambda}{d\theta} = 0$ , it is seen that  $\lambda_c$  satisfies

$$\theta = \frac{1}{2} \lambda_c \cosh \left( \frac{\theta}{4} \right) \sinh \left( \frac{\theta}{4} \right). \tag{27}$$

Substituting the value of  $\lambda_c$  from (27) in (26) one gets the value of  $\theta_c$ ,

$$\frac{\theta_c}{4} = \coth \left( \frac{\theta_c}{4} \right), \tag{28}$$

for which  $\theta_c = 4.798714560$  can be obtained using an iterative method. Then the value of  $\lambda_c = 3.513830720$  is obtained from Equation (26). Figure 1 display this critical value of  $\lambda_c$ . Equation (24) is discretized using the finite-difference scheme to obtain

$$F_j(U_j) = \frac{U_{j+1} - 2U_j + U_{j-1}}{h^2} + \lambda \exp U_j = 0, j = 1(1)M - 1, \tag{29}$$

subject to the conditions  $U_0 = U_M = 0$  with mesh length  $h = 1/M$ . There are  $M - 1$  difference equations. The Jacobian matrix obtained is sparse and it has three non-zero entries on the main and two sub-diagonals. The solution of the above finite-difference equations converges to the solution of 1-D Bratu problem with the help of initial vector  $U^{(0)} = (0, 0, \dots, 0)^T$ .

We take  $M = 100$  and experiment for 350  $\lambda$ 's in the interval  $\lambda \in (0, 3.5]$  (mesh length = 0.01). For every  $\lambda$ , take  $M_\lambda$  to be the minimum number of iterations so that  $\|U_j^{(r+1)} - U_j^{(r)}\|_2 < 10^{-13}$ , where, the computed value  $U_j^{(r)}$  is calculated for 13-decimal-place accuracy. Let  $\overline{M}_\lambda$  represents the average of iteration number for the 350  $\lambda$ 's. Table 3 presents the numerical results for one-dimensional Bratu problem, where  $N$  denotes the number of iterations for convergence. It is observed from all the methods

found in Table 3, whenever the methods have higher-order convergence, a greater number of points in  $\lambda$  converge with fewer iterations. For the proposed 5<sup>th</sup> PJ and 8<sup>th</sup> PJ methods, a greater number of points in  $\lambda$  converge within four iterations. However, the least average iteration number ( $\overline{M}_\lambda$ ) reduces as the order of method increases. Therefore, it is found that 5<sup>th</sup> PJ and 8<sup>th</sup> PJ methods are better among the methods taken for comparison since they have the least average iteration number ( $\overline{M}_\lambda$ ).

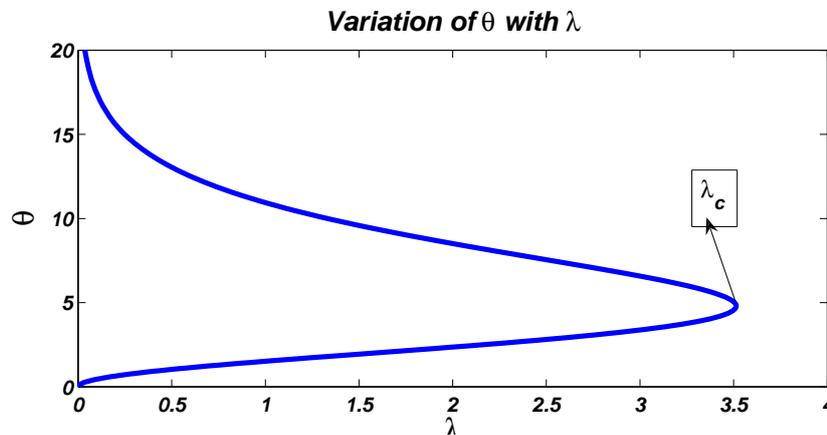


Figure 1. Variation of  $\theta$  for 1-D Bratu Problem

Table 3. Comparison of number of  $\lambda$ 's requiring N iterations to converge in 1-D Bratu problem.

Method	N = 2	N = 3	N = 4	N = 5	N > 5	$\overline{M}_\lambda$
2 <sup>nd</sup> NM	0	12	115	142	81	4.93
4 <sup>th</sup> NR	12	257	76	4	1	3.21
5 <sup>th</sup> ACT	1	136	171	37	5	3.71
5 <sup>th</sup> MBJ	23	263	60	3	1	3.14
5 <sup>th</sup> PJ	23	276	48	2	1	3.10
8 <sup>th</sup> SA	23	263	60	3	1	3.13
8 <sup>th</sup> MBJ	23	262	61	2	2	3.14
8 <sup>th</sup> PJ	23	276	48	1	2	3.09

### 6.2. Bratu Problem in Two Dimension

The following partial differential equations representing two-dimensional Bratu problem [19] is considered:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \lambda \exp(U) = 0, \quad x, y \in D = [0, 1] \times [0, 1] \tag{30}$$

satisfying the conditions

$$U(x, y) = 0, \quad x, y \in \Gamma, \tag{31}$$

where  $\Gamma$  represents the boundary of D. The two known bifurcated exact solutions of the Planar Bratu problem are obtained for the values of  $\lambda < \lambda_c$ , one solution for  $\lambda = \lambda_c$  and for  $\lambda > \lambda_c$  solution does not exist. The analytic solution for Equation (30) is found to be

$$U(x, y) = 2 \log_e \left[ \frac{\cosh\left(\frac{\theta}{4}\right) \cosh\left(\left(x - \frac{1}{2}\right)\left(y - \frac{1}{2}\right)\theta\right)}{\cosh\left(\left(x - \frac{1}{2}\right)\left(\frac{\theta}{2}\right)\right) \cosh\left(\left(y - \frac{1}{2}\right)\left(\frac{\theta}{2}\right)\right)} \right]. \tag{32}$$

The constant  $\theta$  must be found such that it satisfies the boundary conditions. Substitute Equation (32) in (30) and convert it by using collocation at  $(x, y) = \left(\frac{1}{2}, \frac{1}{2}\right)$  as it is the center of the domain D. Instead,

one can choose a different value for  $(x, y)$ , although it may give higher-order approximation but collocating at  $(x, y) = \left(\frac{1}{2}, \frac{1}{2}\right)$  will give equal distribution of the interval throughout the region and hence we get

$$\theta^2 = \lambda \cosh^2 \left(\frac{\theta}{4}\right). \tag{33}$$

Differentiate the above Equation (33) with respect to  $\theta$  and by taking  $\frac{d\lambda}{d\theta} = 0$ , it is seen that  $\lambda_c$  satisfies

$$\theta = \frac{1}{4}\lambda_c \cosh \left(\frac{\theta}{4}\right) \sinh \left(\frac{\theta}{4}\right). \tag{34}$$

Substituting the value of  $\lambda_c$  from (34) in (33) one gets the value of  $\theta_c$ ,

$$\frac{\theta_c}{4} = \coth \left(\frac{\theta_c}{4}\right), \tag{35}$$

for which  $\theta_c = 4.798714561$  can be obtained using an iterative method. Then the value of  $\lambda_c = 7.027661438$  is obtained from Equation (34). Figure 2 display this critical value of  $\lambda_c$ .

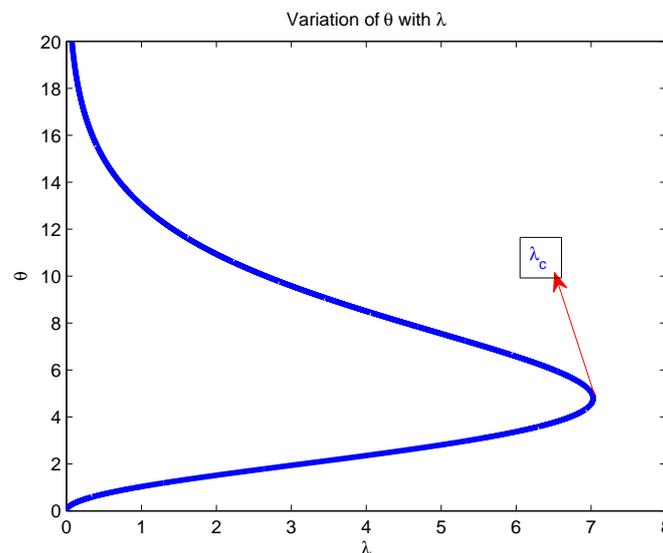


Figure 2. Variation of  $\theta$  for 2-D Bratu Problem

Equation (30) is discretized using the finite-difference five-point formula with mesh length  $h$  on both the axes to obtain the nonlinear equations

$$F(U_{i,j}) = -(4U_{i,j} - \lambda h^2 \exp(U_{i,j})) + U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1}, \tag{36}$$

where  $U_{i,j}$  is  $U$  at  $(x_i, y_j)$ ,  $x_i = ih$ ,  $y_j = jh$ ,  $i, j = 1(1)M$ , subject to the conditions  $U_{i,0} = U_{i,M+1} = U_{0,j} = U_{M+1,j} = 0$  for  $i, j = 0(1)M + 1$ . Equation (36) represents a set of  $M \times M$  nonlinear equations in  $U_{i,j}$  which are then solved by using iterative methods. By taking  $M = 10$  and  $M = 20$ , we experiment for 700  $\lambda$ 's, in the interval  $\lambda \in (0, 7]$  (mesh length = 0.01). For every  $\lambda$ , take  $M_\lambda$  to be the minimum number of iterations so that  $\|U_{i,j}^{(r+1)} - U_{i,j}^{(r)}\|_2 < 10^{-11}$ , where, the computed value  $U_{i,j}^{(r)}$  is calculated for 13-decimal-place accuracy. Let  $\overline{M}_\lambda$  represents the average of iteration number for the 700  $\lambda$ 's.

Tables 4 and 5 presents the numerical results for this problem, where  $N$  denotes the number of iterations for convergence of the solution of nonlinear Equation (36). It is observed from all the methods found in Table 5 that the presented 8<sup>th</sup> PJ algorithm converges for all the mesh points in two iterations. With respect to the lowest mean iteration number ( $\overline{M}_\lambda$ ), 8<sup>th</sup> PJ algorithm is better among the

algorithms taken for comparison since they have the least  $\overline{M}_\lambda$ , while performing equivalently with few same-order algorithms for  $M = 20$ .

**Table 4.** Comparison of number of  $\lambda$ 's requiring N iterations to converge in 2-D Bratu problem ( $M = 10$ ).

Method	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N > 5$	$\overline{M}_\lambda$
2 <sup>nd</sup> NM	0	101	520	79	0	3.96
4 <sup>th</sup> NR	101	599	0	0	0	2.85
5 <sup>th</sup> ACT	200	500	0	0	0	2.71
5 <sup>th</sup> MBJ	121	579	0	0	0	2.82
5 <sup>th</sup> PJ	120	580	0	0	0	2.82
8 <sup>th</sup> SA	514	186	0	0	0	2.26
8 <sup>th</sup> MBJ	514	186	0	0	0	2.26
8 <sup>th</sup> PJ	441	259	0	0	0	2.37

**Table 5.** Comparison of number of  $\lambda$ 's requiring N iterations to converge in 2-D Bratu problem ( $M = 20$ ).

Method	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$\overline{M}_\lambda$
2 <sup>nd</sup> NM	1	212	487	0	3.69
4 <sup>th</sup> NR	213	487	0	0	2.69
5 <sup>th</sup> ACT	419	281	0	0	2.40
5 <sup>th</sup> MBJ	217	483	0	0	2.69
5 <sup>th</sup> PJ	217	483	0	0	2.69
8 <sup>th</sup> SA	700	0	0	0	2
8 <sup>th</sup> MBJ	700	0	0	0	2
8 <sup>th</sup> PJ	700	0	0	0	2

### 7. Conclusions

Newton-type algorithms produce second and fourth-order convergence for single and two-step methods, respectively. By using appropriate weight function, we have improved the Newton's fourth-order method to fifth-order by keeping the same number of function evaluation. Furthermore, a multi-step version producing higher-order convergence has been proposed with fifth-order method as its base to solve nonlinear system of equations. The new schemes do not use second or higher-order Frechet derivatives to achieve higher-order convergence. Computational efficiencies of the proposed algorithms are calculated based on the computational cost and comparison with other methods in terms of ratio which shows that the presented algorithms are better than many other algorithms. By considering a few examples, we have illustrated numerically the new schemes are superior to that of Newton's method and some recently available fourth-, fifth-, and eighth-order algorithms. Also, we infer from the computational results, the proposed methods have robust and efficient convergence behavior. To check the application aspect of the new algorithms, we have implemented them on one-dimensional and two-dimensional Bratu problems. The results of these applications indicate that the presented methods perform better than some available algorithms.

**Author Contributions:** Conceptualization, validation, writing—review and editing, resources, supervision—J.J.; Methodology, software, formal analysis, data curation, writing—original draft preparation, visualization—P.S.

**Funding:** This research received no external funding.

**Acknowledgments:** The authors would like to thank the anonymous reviewers for their constructive comments and useful suggestions which greatly helped to improve this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Abad, M.F.; Cordero, A.; Torregrosa, J.R. Fourth and fifth-order methods for solving nonlinear systems of equations: An application to the global positioning system. *Abstr. Appl. Anal.* **2013**, *2013*, 586708. [[CrossRef](#)]
2. Babajee, D.K.R.; Madhu, K.; Jayaraman, J. On some improved harmonic mean newton-like methods for solving systems of nonlinear equations. *Algorithms* **2015**, *8*, 895–909. [[CrossRef](#)]
3. Cordero, A.; Hueso, J.L.; Martinez, E.; Torregrosa, J.R. A modified newton-jarratt's composition. *Numer. Algorithms* **2010**, *55*, 87–99. [[CrossRef](#)]
4. Madhu, K. Sixth order newton-type method for solving system of nonlinear equations and its applications. *Appl. Math. E-Notes* **2017**, *17*, 221–230.
5. Noor, M.A.; Waseem, M.; Noor, K.I.; Al-Said, E. Variational iteration technique for solving a system of nonlinear equations. *Optim. Lett.* **2013**, *7*, 991–1007. [[CrossRef](#)]
6. Sharma, J.R.; Arora, H. Improved newton-like methods for solving systems of nonlinear equations. *SeMA J.* **2017**, *74*, 147–163. [[CrossRef](#)]
7. Sharma, J.R.; Guha, R.K.; Sharma, R. An efficient fourth order weighted-newton method for systems of nonlinear equations. *Numer. Algorithms* **2013**, *62*, 307–323. [[CrossRef](#)]
8. Traub, J.F. *Iterative Methods for the Solution of Equations*; Prentice-Hall: Upper Saddle River, NJ, USA, 1964.
9. Madhu, K.; Babajee, D.K.R.; Jayaraman, J. An improvement to double-step newton method and its multi-step version for solving system of nonlinear equations and its applications. *Numer. Algorithms* **2017**, *74*, 593–607. [[CrossRef](#)]
10. Amat, S.; Busquier, S.; Gutierrez, J.M. Third-order iterative methods with applications to hammerstein equations: A unified approach. *J. Comput. Appl. Math.* **2011**, *235*, 2936–2943. [[CrossRef](#)]
11. Amat, S.; Busquier, S.; Gutierrez, J.M. Geometric constructions of iterative methods to solve nonlinear equations. *Comput. Appl. Math.* **2003**, *157*, 197–205. [[CrossRef](#)]
12. Argyros, I.K.; Magrenan, A.A. *Iterative Methods and Their Dynamics with Applications*; CRC Press: New York, NY, USA, 2017.
13. Sharma, J.R.; Argyros, I.K.; Kumar, S. Ball convergence of an efficient eighth order iterative method under weak conditions. *Mathematics* **2018**, *6*, 260. [[CrossRef](#)]
14. Yuan, G.; Duan, X.; Liu, W.; Wang, X.; Cui, Z.; Sheng, Z. Two new PRP Conjugate Gradient Algorithms for Minimization Optimization Models. *PLoS ONE* **2015**, *10*, 1–24. [[CrossRef](#)] [[PubMed](#)]
15. Yuan, G.; Zhang, M. A three-terms Polak-Ribire-Polyak conjugate gradient algorithm for large-scale nonlinear equations. *J. Comput. Appl. Math.* **2015**, *286*, 186–195. [[CrossRef](#)]
16. Ostrowski, A.M. *Solutions of Equations and System of Equations*; Academic Press: New York, NY, USA, 1960.
17. Sharma, J.R.; Kumar, D. On a class of efficient higher order newton-like methods. *Math. Model. Anal.* **2019**, *24*, 105–126. [[CrossRef](#)]
18. Buckmire, R. Investigations of nonstandard Mickens-type finite-difference schemes for singular boundary value problems in cylindrical or spherical coordinates. *Num. Methods Partial Differ. Equ.* **2003**, *19*, 380–398. [[CrossRef](#)]
19. Odejide, S.A.; Aregbesola, Y.A.S. A note on two dimensional Bratu problem. *Kragujev. J. Math.* **2006**, *29*, 49–56.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).