Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033528
Article

Efficient Numerical Scheme for Solving Large System of Nonlinear Equations

Mudassir Shams[1], Nasreen Kausar[2,*], Shams Forruque Ahmed[3], Irfan Anjum Badruddin[4] and Syed Javed[4]

[1]Department of Mathematics and Statistics, Riphah International University, I-14, Islamabad, 44000, Pakistan
[2]Department of Mathematics, Yildiz Technical University, Faculty of Arts and Science, Esenler, 34210, Istanbul, Turkey
[3]Science and Math Program, Asian University for Women, Chattogram, Bangladesh
[4]Mechanical Engineering Department, College of Engineering, King Khalid University, Abha, 61421, Saudi Arabia
*Corresponding Author: Nasreen Kausar. Email: kausar.nasreen57@gmail.com
Received: 19 June 2022; Accepted: 15 September 2022

Abstract: A fifth-order family of an iterative method for solving systems of nonlinear equations and highly nonlinear boundary value problems has been developed in this paper. Convergence analysis demonstrates that the local order of convergence of the numerical method is five. The computer algebra system CAS-Maple, Mathematica, or MATLAB was the primary tool for dealing with difficult problems since it allows for the handling and manipulation of complex mathematical equations and other mathematical objects. Several numerical examples are provided to demonstrate the properties of the proposed rapidly convergent algorithms. A dynamic evaluation of the presented methods is also presented utilizing basins of attraction to analyze their convergence behavior. Aside from visualizing iterative processes, this methodology provides useful information on iterations, such as the number of diverging-converging points and the average number of iterations as a function of initial points. Solving numerous highly nonlinear boundary value problems and large nonlinear systems of equations of higher dimensions demonstrate the performance, efficiency, precision, and applicability of a newly presented technique.

Keywords: Nonlinear equations; convergence order; boundary value problem; computational time; basins of attraction; converging points

1 Introduction

Determining the roots of polynomial equations is among the oldest problems in mathematics, whereas polynomial equations have a wide range of applications in science and engineering. For example, aerospace engineers may use polynomials to determine the acceleration of a rocket or jet, and mechanical engineers use polynomials to research and design engines and machines. The search for finding the roots of a system of polynomials and a system of linear or nonlinear equations is one of the primal and difficult problems with wide applications in science, engineering, finance and particular in differential equations. Iterative numerical schemes for solving nonlinear systems of equations associated with initial value problems or boundary value problems are very important because, in general, obtaining a closed form solution using the analytical or exact technique is quite difficult. Generally, nonlinear initial value problems or boundary value problems are solved in two main steps i.e., first to discretize the problem using the difference method, finite difference method, finite element method, Pseudo-Spectral collocation method to the obtained tridiagonal system of linear or nonlinear equations, and in the second step, some numerical iterative numerical schemes are used to solve the tridiagonal system of linear or nonlinear equations.

The first famous, effective and very simple scheme is Newton's method to solve a nonlinear system of equations is given as:

$$\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \mathbf{F}'(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{x}^{(k)}), \tag{1}$$

where $\mathbf{F}'(\mathbf{x})$ is the Jacobin matrix approximated at $\mathbf{x}^{(k)}$ i.e.,

$$\mathbf{F}'(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial \mathrm{x}_1} & \frac{\partial f_1}{\partial \mathrm{x}_2} & \cdots & \frac{\partial f_1}{\partial \mathrm{x}_n} \\ \frac{\partial f_2}{\partial \mathrm{x}_1} & \frac{\partial f_2}{\partial \mathrm{x}_2} & \cdots & \frac{\partial f_2}{\partial \mathrm{x}_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial \mathrm{x}_1} & \frac{\partial f_n}{\partial \mathrm{x}_2} & \cdots & \frac{\partial f_n}{\partial \mathrm{x}_n} \end{pmatrix} \text{ and } \mathbf{F}(\mathbf{x}) = \mathbf{F}(x_1, \ldots, x_n) = \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ f_2(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{pmatrix} = 0. \tag{2}$$

Method (2), has quadratic convergence locally. A lot of modifications have been made in classical Newton's Raphson method in order to reduce the number of function and Jacobin evaluations in each iteration step, and so accelerate the convergence order. The extension of the classical Newton method, as described by Weerakoon et al. [1], Özban [2], Gerlach [3] and Young et al. [4], to the function of serval variable has been developed in [5–7] and references therein.

An open closed quadrature-based iterative method was designed by Frontini et al. [8]. This method was improved by Darvishi et al. [9] to obtain a fourth-order scheme. A number of methods, such as the domain decomposition method [10,11], the weight function technique [12], and the replacement of the higher derivative by an approximation [13–15], were used to develop iterative methods to solve a system of nonlinear equation.

The fundamental goal of this study is to construct a higher-order iterative method for solving nonlinear system of equations and highly nonlinear boundary value problems. Basins of attraction are used to demonstrate the efficiency of our method in comparison to the literature's existing method.

This article is organized as follows: Following the introduction in Section 1, Section 2 provides a brief description of method construction and convergence analysis. The dynamical aspect of the proposed technique's attraction basins is discussed in Section 3. The numerical outcomes of the proposed method and comparisons to other higher-order existing methods from the literature are shown in Section 4. The paper concludes with Section 5.

2 Construction of Numerical Methods and Convergence Analysis

This section presents some well-known existing iterative schemes of fifth-order convergence.

In 2020, Singh [16] proposed the following fifth-order technique ($MS\alpha_1$):

$$\mathbf{x}^{(k+1)} = \mathbf{w}^{(k)} - \left(\frac{\mathbf{F}(\mathbf{w}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right), \tag{3}$$

where $\mathbf{w}^{(k)} = \mathbf{z}^{(k)} - \left(\frac{\mathbf{F}(\mathbf{z}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)$, $\mathbf{z}^{(k)} = \mathbf{x}^{(k)} - \left(\frac{\mathbf{F}(\mathbf{x}^{(k)}) + \mathbf{F}(y^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)$ and $\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \left(\frac{\mathbf{F}(x^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)$.

In 2013, Zhang et al. [17] presented the fifth-order iterative technique ($MS\alpha_2$) as below:

$$\mathbf{x}^{(k+1)} = \mathbf{z}^{(k)} - \left(\frac{\mathbf{F}(\mathbf{z}^{(k)})}{\mathbf{F}'(\mathbf{y}^{(k)})}\right), \tag{4}$$

where $\mathbf{z}^{(k)} = \mathbf{x}^{(k)} - \frac{1}{2}\left(\frac{\mathbf{8F}\left(\mathbf{x}^{(k)}\right)}{\mathbf{F}'\left(\mathbf{x}^{(k)}\right) + 3\left(\mathbf{F}'\left(\frac{\mathbf{2x}^{(k)} + \mathbf{y}^{(k)}}{3}\right) + \mathbf{F}'\left(\frac{\mathbf{x}^{(k)} + 2\mathbf{y}^{(k)}}{3}\right) + \mathbf{F}'\left(\mathbf{y}^{(k)}\right)\right)}\right)$ and $\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \left(\frac{\mathbf{F}(\mathbf{x}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)$.

Cordero et al. [18] developed the following fifth-order iterative scheme ($MS\alpha_3$) in 2007:

$$\mathbf{z}^{(k)} = \mathbf{x}^{(k)} - 3\left(\frac{\mathbf{F}\left(\mathbf{x}^{(k)}\right)}{2F'\left(\frac{3x^{(k)} + y^{(k)}}{4}\right) - \mathbf{F}'\left(\frac{\mathbf{x}^{(k)} + y^{(k)}}{2}\right) + 2F'\left(\frac{\mathbf{x}^{(k)} + 3y^{(k)}}{4}\right)}\right), \tag{5}$$

where $\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \left(\frac{\mathbf{F}(\mathbf{x}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)$.

Cordero et al. [18] also constructed the following fifth-order iterative scheme ($MS\alpha_4$):

$$\mathbf{z}^{(k)} = \mathbf{x}^{(k)} - 6\left(\frac{\mathbf{F}\left(\mathbf{x}^{(k)}\right)}{\mathbf{F}'\left(\mathbf{x}^{(k)}\right) + 4F'\left(\frac{\mathbf{x}^{(k)} + y^{(k)}}{2}\right) + \mathbf{F}'\left(y^{(k)}\right)}\right), \tag{6}$$

where $\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \left(\frac{\mathbf{F}(\mathbf{x}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)$.

The following scheme (abbreviated as $MS\alpha_*$) is proposed in the present study:

$$\mathbf{x}^{(k+1)} = \mathbf{y}^{(k)} - \left(\frac{8\mathbf{F}'(\mathbf{y}^{(k)}) - 6\mathbf{F}'(\mathbf{x}^{(k)})}{10\mathbf{F}'(\mathbf{y}^{(k)}) - 8\mathbf{F}'(\mathbf{x}^{(k)})} - \frac{15}{4}\left(\frac{\mathbf{F}'(\mathbf{y}^{(k)}) - \mathbf{F}'(\mathbf{x}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)^2\right)\left(\frac{\mathbf{F}(\mathbf{y}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right), \tag{7}$$

where $\mathbf{y}^{(k)} = \mathbf{x}^{(k)} - \left(\frac{\mathbf{F}(\mathbf{x}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)$.

Convergence analysis

For the iteration schemes (7), we have the following convergence theorem by using the computer algebra system CAS-Maple 18 and finding the error relation of the iterative schemes defined in (7).

Theorem *Let the function* $\mathbf{F} : E \subseteq \mathrm{R}^n \rightarrow \mathrm{R}^n$ *be sufficiently Fréchet differentiable on an open set* E *containing the root* $\boldsymbol{\zeta}$ *of* $\mathbf{F}(\mathbf{x}^{(k)}) = 0$. *If the initial estimation* $\mathbf{x}^{(0)}$ *is close to* $\boldsymbol{\zeta}$, *the method's* $\mathrm{MS}\alpha_*$ *convergence order is at least five and satisfies the following:*

$$\widehat{\mathbf{e}}^{(k)} = \left(-186\mathbf{C}_2^4 - \mathbf{C}_2^2\mathbf{C}_3\right)(\mathbf{e}^{(k)})^5 + \left\|\mathbf{O}(\mathbf{e}^{(k)})^6\right\|, \tag{8}$$

where $C_i = \dfrac{1}{i!}\frac{\mathbf{F}'(\zeta^{(k)})}{\mathbf{F}^{(i)}(\zeta^{(k)})}, \quad i = 2, 3, \ldots$

Proof: *Let* $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \boldsymbol{\zeta}$, $\mathbf{e}^{\sim(k)} = \mathbf{y}^{(k)} - \boldsymbol{\zeta}$ *and* $\widehat{\mathbf{e}}^{(k)} = \mathbf{x}^{(k+1)} - \boldsymbol{\zeta}$ *be the error in generating Taylor series* $\mathbf{F}(\mathbf{x}^{(k)})$ *in the region of* $\boldsymbol{\zeta}$ *assuming that* $\mathbf{F}'(\mathbf{r})^{-1}$ *exists, we write:*

$$\mathbf{F}(\mathbf{x}^{(k)}) = \mathbf{F}\boldsymbol{\zeta}^{(k)} + \mathbf{F}'(\boldsymbol{\zeta}^{(k)})(\mathbf{x}-\mathbf{x}^{(k)}) + \frac{1}{2!}\mathbf{F}''(\boldsymbol{\zeta}^{(k)})(\mathbf{x}-\mathbf{x}^{(k)})^2 + \frac{1}{3!}\mathbf{F}'''(\boldsymbol{\zeta}^{(k)})(\mathbf{x}-\mathbf{x}^{(k)})^3 + \ldots \tag{9}$$

and $\mathbf{F}(\mathbf{x}) = \mathbf{0}$,

$$\mathbf{F}(\mathbf{x}^{(k)}) = \mathbf{F}'(\zeta^{(k)})\left\{\mathbf{e}^{(k)} + \mathbf{C}_2(\mathbf{e}^{(k)})^2 + \mathbf{C}_3(\mathbf{e}^{(k)})^3 + \cdots + \mathbf{C}_6(\mathbf{e}^{(k)})^6\right\} + \left|\left|\mathbf{O}(\mathbf{e}^{(k)})^7\right|\right|, \tag{10}$$

Dividing Eq. (9) by $[\mathbf{F}'(\mathbf{x}^{(k)})]_{-1}$, we have:

$$[\mathbf{F}'(\mathbf{x}^{(k)})]^{-1}\mathbf{F}(\mathbf{x}^{(k)}) = \mathbf{e}^{(k)} - \mathbf{C}_2(\mathbf{e}^{(k)})^2 + (2(\mathbf{C}_2)^2 - 2\mathbf{C}_3)(\mathbf{e}^{(k)})^3 + \ldots \tag{11}$$

$$\mathbf{e}^{\sim(k)} = \mathbf{y}^{(k)} - \boldsymbol{\zeta} = \mathbf{C_2} + (-2\mathbf{C}_2^2 + 2\mathbf{C}_3)(\mathbf{e}^{(k)})^3 + \ldots \tag{12}$$

Expanding $\mathbf{F}'(\mathbf{y}^{(k)})$ about $\boldsymbol{\zeta}$ and using Eq. (12), we obtain:

$$\mathbf{F}'(\mathbf{y}^{(k)}) = 1 + 2\mathbf{C}_2^2(\mathbf{e}^{\sim(k)})^2 + 2(-2\mathbf{C}_2^2 + 2\mathbf{C}_3)(\mathbf{e}^{\sim(k)})^3 + \ldots \tag{13}$$

$$\begin{aligned}8\left(\mathbf{F}'(\mathbf{y}^{(k)}) - 6\mathbf{F}'(\mathbf{x}^{(k)})\right)] =& \mathbf{2}-\mathbf{12C}_2(\mathbf{e}^{\sim(k)}) + (16\mathbf{C}_2^2 - 18\mathbf{C}_3)(\mathbf{e}^{\sim(k)})^2 + (-32\mathbf{C}_2^3\\ &+ 32\mathbf{C}_2\mathbf{C}_3 - 24\mathbf{C}_4)(\mathbf{e}^{\sim(k)})^3 + \ldots\end{aligned} \tag{14}$$

$$\begin{aligned}10\left(\mathbf{F}'(\mathbf{y}^{(k)}) - 8\mathbf{F}'(\mathbf{x}^{(k)})\right) =& 2 - 16\mathbf{C}_2(\mathbf{e}^{\sim(k)}) + (20\mathbf{C}_2^2 - 24\mathbf{C}_3)(\mathbf{e}^{\sim(k)})^2 + (-40\mathbf{C}_2^3\\ &+ 40\mathbf{C}_2\mathbf{C}_3 - 32\mathbf{C}_4)(\mathbf{e}^{\sim(k)})^3 + \ldots\end{aligned} \tag{15}$$

$$\begin{aligned}15\left(\mathbf{F}'(\mathbf{y}^{(k)}) - \mathbf{F}'(\mathbf{x}^{(k)})\right)]^2 =& 60\mathbf{C}_2^2(\mathbf{e}^{\sim(k)})^2 + (-120\mathbf{C}_2^3 + 180\mathbf{C}_2\mathbf{C}_3)(\mathbf{e}^{\sim(k)})^3 + (300\mathbf{C}_2^4\\ &- 420\mathbf{C}_2^2\mathbf{C}_3 + 240\mathbf{C}_2\mathbf{C}_4 + 125\mathbf{C}_3^2)(\mathbf{e}^{\sim(k)})^4 + \left|\left|\mathbf{O}(\mathbf{e}^{\sim(k)})^5\right|\right|.\end{aligned} \tag{16}$$

$$\begin{aligned}4\left(\mathbf{F}'(\mathbf{x}^{(k)})\right)]^2 =& 4 + 16_2\mathbf{C}(\mathbf{e}^{\sim(k)}) + (16\mathbf{C}_2^2 + 24\mathbf{C}_3)(\mathbf{e}^{\sim(k)})^2 + (48\mathbf{C}_2\mathbf{C}_3 + 32\mathbf{C}_4)(\mathbf{e}^{\sim(k)})^3\\ &+ (64\mathbf{C}_2\mathbf{C}_4 + 36\mathbf{C}_3^2 + 40\mathbf{C}_5) + \left|\left|\mathbf{O}(\mathbf{e}^{\sim(k)})^4\right|\right|.\end{aligned} \tag{17}$$

Using Eq. (13) and Eq. (15) in the second-step of Eq. (7), we get:

$$\left(\frac{8\mathbf{F}'(\mathbf{y}^{(k)}) - 6\mathbf{F}'(\mathbf{x}^{(k)})}{10\mathbf{F}'(\mathbf{y}^{(k)}) - 8\mathbf{F}'(\mathbf{x}^{(k)})} - \frac{15}{4}\left(\frac{\mathbf{F}'(\mathbf{y}^{(k)}) - \mathbf{F}'(\mathbf{x}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right)^2\right)\left(\frac{\mathbf{F}(\mathbf{y}^{(k)})}{\mathbf{F}'(\mathbf{x}^{(k)})}\right) = 1 + 2\mathbf{C}(\mathbf{e}^{\sim(k)})$$

$$+ (-\mathbf{C}_2^2 + 3\mathbf{C}_3)(\mathbf{e}^{\sim(k)})^2 + (186\mathbf{C}_2^3 - \mathbf{C}_2\mathbf{C}_3 + 4\mathbf{C}_4)(\mathbf{e}^{\sim(k)})^3 + (5\mathbf{C}_5 - 285\mathbf{C}_2^4 + \frac{9}{4}\mathbf{C}_3^2 + 836\mathbf{C}_2^2\mathbf{C}_3$$

$$- 2\mathbf{C}_2\mathbf{C}_4)(\mathbf{e}^{\sim(k)})^4 + (5856\mathbf{C}_2^5 + 2267\mathbf{C}_2^3\mathbf{C}_3$$

$$+ 1118\mathbf{C}_2^2\mathbf{C}_4 + 1251\mathbf{C}_3^2\mathbf{C}_2 - 3\mathbf{C}_2\mathbf{C}_5 + 6\mathbf{C}_4\mathbf{C}_3 + 6\mathbf{C}_6)(\mathbf{e}^{\sim(k)})^5 + \ldots \tag{18}$$

$$\widehat{\mathbf{e}}^{(k)} = \mathbf{x}^{(k+1)} - \boldsymbol{\zeta} = \left(-186\mathbf{C}_2^4 - \mathbf{C}_2^2\mathbf{C}_3\right)(\mathbf{e}^{(k)})^5 + \left\|\mathbf{O}(\mathbf{e}^{(k)})^6\right\|. \tag{19}$$

Hence, it proves the theorem

3 Dynamical Planes

The basins of attraction [19–22] is a graphical representation of how root-finding algorithms respond to different initial estimate points. It is more than a graphical illustration of how a root-finding approach works; it also enables the comparsion of qualitative issues. Visual analysis of dynamical planes, i.e., basins of attraction, is another effective and profitable means of demonstrating the usefulness of iterative methods for solving nonlinear equations with these advantageous properties. A complex square $|-3, 3 \times -3, 3|^2 \in \mathbb{C}$ with its centre at the origin and a total of 490000 points is used to generate the dynamical planes. The region on which the first hypotheses are predicted is analyzed in order to locate the root of the nonlinear equation. The stopping criterion $\left|x^{k+1} - x^k\right| < 10^{-3}$ is utilised to terminate the computer program, and a maximum of 20 iterations are needed root to convergence of the root. Dark black points are assigned, if the orbit of the iterative methods does not converge to root after 20 iterations. Each root is assigned a unique color. In iterative techniques, distinct basins of attraction are illustrated by different colours. Figs. 1–15 illustrate basins of attraction generated by iterative methods for the following non-linear equations:

$$x^5 + x^3 - x - 1 = 0. \tag{20}$$

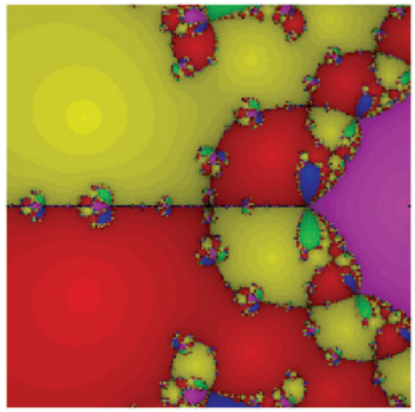

Figure 1: The program's outcome—the basins of attraction for $MS\alpha_*$ applied to Eq. (20)

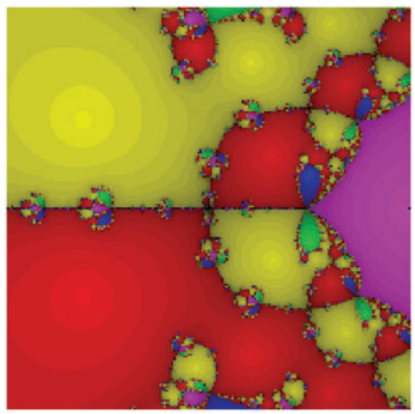

Figure 2: The program's outcome—the basins of attraction for $MS\alpha_1$ applied to Eq. (20)

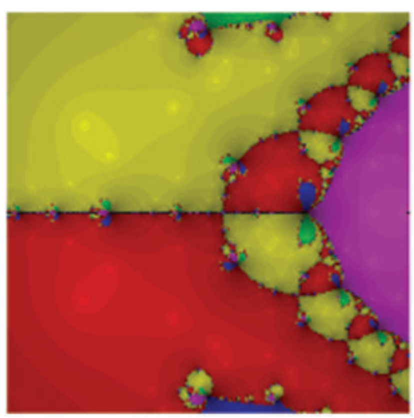

Figure 3: The program's outcome—the basins of attraction for $MS\alpha_2$ applied to Eq. (20)

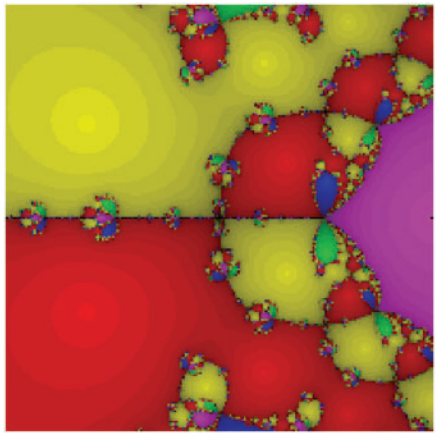

Figure 4: The program's outcome—the basins of attraction for $MS\alpha_3$ applied to Eq. (20)

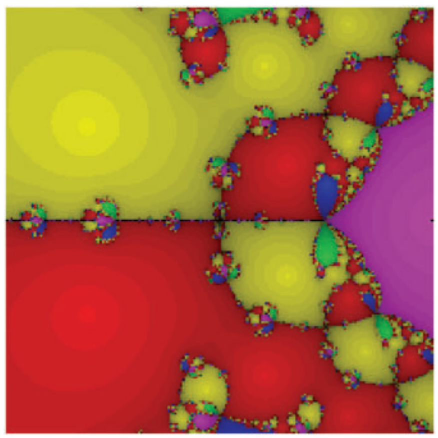

Figure 5: The program's outcome—the basins of attraction for $MS\alpha_4$ applied to Eq. (20)

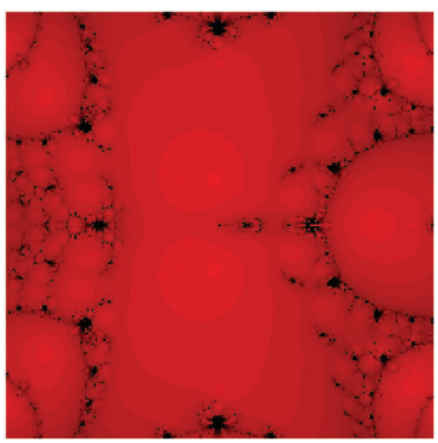

Figure 6: The program's outcome—the basins of attraction for $MS\alpha_*$ applied to Eq. (21)

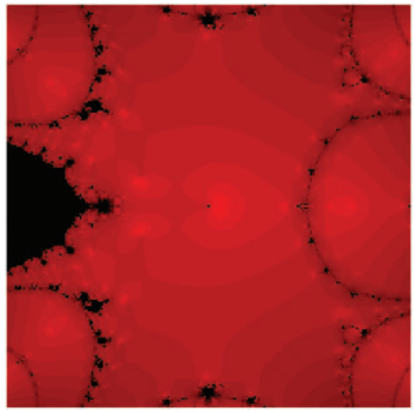

Figure 7: The program's outcome—the basins of attraction for MSα_1 applied to Eq. (21)

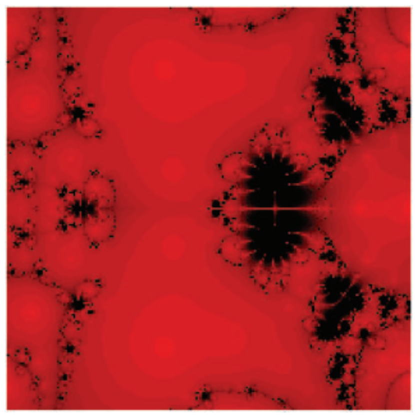

Figure 8: The program's outcome—the basins of attraction for MSα_2 applied to Eq. (21)

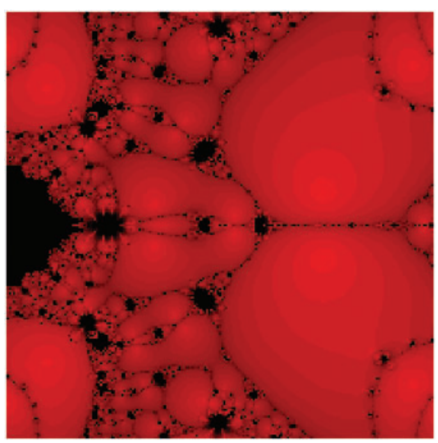

Figure 9: The program's outcome—the basins of attraction for MSα_3 Eq. (21)

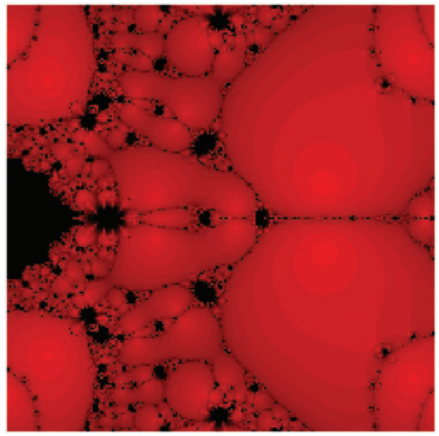

Figure 10: The program's outcome—the basins of attraction for $MS\alpha_4$ applied Eq. (21)

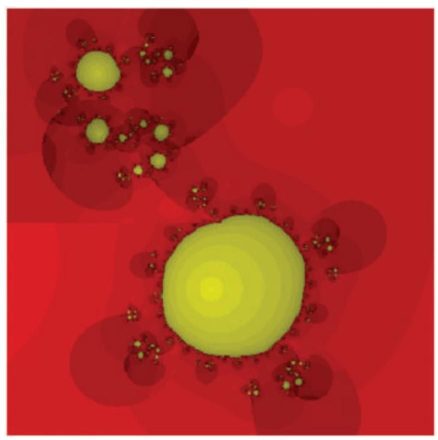

Figure 11: The program's outcome—the basins of attraction for $MS\alpha_*$ applied to Eq. (22)

Figure 12: The program's outcome—the basins of attraction for $MS\alpha_1$ applied to Eq. (22)

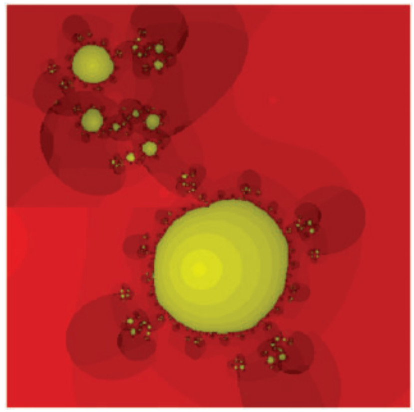

Figure 13: The program's outcome—the basins of attraction for $MS\alpha_2$ applied to Eq. (22)

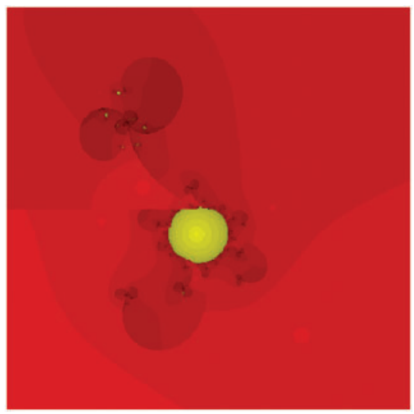

Figure 14: The program's outcome—the basins of attraction for $MS\alpha_3$ applied to Eq. (22)

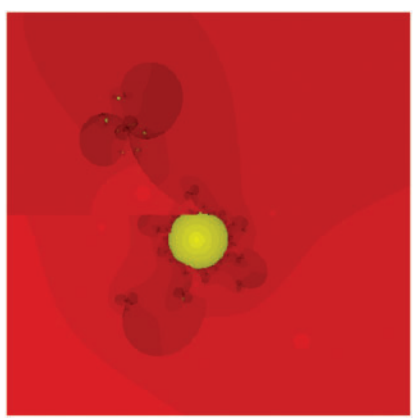

Figure 15: The program's outcome—the basins of attraction for $MS\alpha_4$ applied to Eq. (22)

Eq. (20) has the following exact roots $-0.62174-0.0440597i$, $-0.621744+0.440597i$, $0.121744-1.306621i$, $0.121744+1.30662i$, 1

$$\cos\left(x^5+x^3-x-1\right)-e^{x^{\frac{1}{3}}}=0. \tag{21}$$

Eq. (21) has one real root i.e., 1.135001329.

$$x^{\frac{3}{5}}-\frac{1}{x^{\frac{3}{4}}}+2i=0. \tag{22}$$

Eq. (22) has the following exact roots $-2.15564-0.356601i$, $-0.0918328-0.630339i$.

In Tables 1–3, CPU-Time refers to the elapsed time in seconds, Start-Points denote the number of starting points, i.e., 490,000 in a square, Con-Points represent the number of converging points, and Div-Points signify the number of divergent points for the creation of dynamical planes (Attractions' basins). In terms of CPU-Time, Average-It, Start-Points, Con-Points, and Div-Points, Tables 1–3 clearly show that our newly developed technique $\mathrm{MS}\alpha_*$ outperforms the existing iterative methods $\mathrm{MS}\alpha_1$, $\mathrm{MS}\alpha_2$, $\mathrm{MS}\alpha_3$, $\mathrm{MS}\alpha_4$.

Table 1: Computing time, average iterations, number of initial points, convergence points, and diverging points when applying iterative methods to generate dynamical planes for Eq. (20)

Eq. (20)	$\mathrm{MS}\alpha_*$	$\mathrm{MS}\alpha_1$	$\mathrm{MS}\alpha_2$	$\mathrm{MS}\alpha_3$	$\mathrm{MS}\alpha_4$
CPU-Time	0.01342	0.12345	0.15122	0.014714	0.15670
Average-It	4.50	5.73	5.91	5.01	6.78
Start-Points	490000	490000	490000	490000	490000
Con-Points	490000	488050	490000	490000	490000
Div-Points	0.00000	1950.00	0.00000	0.00000	0.00000

Table 2: Computing time, average iterations, number of initial points, convergence points, and diverging points when applying iterative methods to generate dynamical planes for Eq. (21)

Eq. (21)	$\mathrm{MS}\alpha_*$	$\mathrm{MS}\alpha_1$	$\mathrm{MS}\alpha_2$	$\mathrm{MS}\alpha_3$	$\mathrm{MS}\alpha_4$
CPU-Time	3.12341	6.14214	6.14215	7.12451	3.92451
Average-It	6.15	7.81	7.86	8.15	6.59
Start-Points	490000	490000	490000	490000	490000
Con-Points	471520	380150	410152	391542	376847
Div-Points	18480	109850	79848	98458	113153

Table 3: Computing time, average iterations, number of initial points, convergence points, and diverging points when applying iterative methods to generate dynamical planes for Eq. (22)

Eq. (22)	$MS\alpha_*$	$MS\alpha_1$	$MS\alpha_2$	$MS\alpha_3$	$MS\alpha_4$
CPU-Time	0.01251	0.12470	0.03451	0.01748	0.05602
Average-It	3.45	4.67	4.87	4.15	4.781
Start-Points	490000	490000	490000	490000	490000
Con-Points	490000	490000	485046	487501	490000
Div-Points	0.00000	0.00000	4954.00	2499.00	0.00000

4 Numerical Outcomes

The following iterative techniques are used to solve some extremely non-linear boundry value problem BVP and a large system of non-linear equations:

1. The newly constructed method $MS\alpha_*$ is of convergence order five
2. Singh et al.'s method $MS\alpha_1$ is of convergence order five
3. Zhang et al.'s method $MS\alpha_2$ is of convergence order five
4. Cordero et al.'s method $MS\alpha_3$ is of convergence order five
5. Cordero et al.'s method $MS\alpha_4$ is of convergence order five

All numerical computations are done using maple 18.0 with 75-digit floating point arithmetic in a laptop having Processor Intel® Core™ i3-3310 m CPU@2.4 GHz with a 64-bit operating system on Window 8. We terminate the computer program when the following stopping criterion is satisfied:

$$\mathbf{e} = \left|\left|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right|\right| < \epsilon = 10^{-15},$$

where **e** is the absolute error of the consecutive iterations. In Tables 4–8, D represents the dimension of the non-linear system of equations.

Example 1: N-Demission Problem [23]

Consider

$$\mathbf{F}_1 : f_i(x_i) = e^{x_i^2} - 1, i = 1, 2, 3, .., m \tag{23}$$

the exact solution of the system Eq. (23) is $\mathbf{X}^* = [0, 0, 0, \ldots, 0]^T$ taking $\mathbf{X}_0 = [0.5, 0.5, 0.5, \ldots, 0.5]$ as an initial estimate. Tables 4–5, indicates the numerical results of the system of non-linear equations Eq. (23) used.

Table 4: Iterations-number, and computational time in seconds to solve a large system of non-linear equations $\mathbf{F}_1$(x), $\mathbf{F}_2$(x) using $MS\alpha_*$, $MS\alpha_1$, $MS\alpha_2$, $MS\alpha_3$and $MS\alpha_4$ respectively

Number of iterations of iterative methods for the large system used in Example 1 and Example 2						
Example 1	D	$MS\alpha_*$	$MS\alpha_1$	$MS\alpha_2$	$MS\alpha_3$	$MS\alpha_4$
$\mathbf{F}_1$(x)	50	4	4	4	6	4
$\mathbf{F}_1$(x)	75	4	4	4	6	4

(Continued)

Table 4: Continued

Number of iterations of iterative methods for the large system used in Example 1 and Example 2						
Example 1	D	$MS\alpha_*$	$MS\alpha_1$	$MS\alpha_2$	$MS\alpha_3$	$MS\alpha_4$
$\mathbf{F}_1(x)$	100	4	4	4	6	4
$\mathbf{F}_2(x)$	50	4	4	4	6	4
$\mathbf{F}_2(x)$	75	4	4	4	6	4
$\mathbf{F}_2(x)$	100	4	4	4	6	4
Computational time in seconds of iterative methods for the large system used in Example 1 and Example 2						
$\mathbf{F}_1(x)$	50	0.141	0.329	0.593	0.437	0.312
$\mathbf{F}_1(x)$	75	0.203	0.656	0.891	0.969	0.344
$\mathbf{F}_1(x)$	100	1.113	1.984	1.313	1.484	0.543
$\mathbf{F}_2(x)$	50	0.132	0.671	0.651	1.001	0.751
$\mathbf{F}_2(x)$	75	0.320	0.761	0.766	1.047	0.961
$\mathbf{F}_2(x)$	100	1.008	1.561	1.078	1.574	1.675

Table 5: Accuracy and local computational order of convergence (LCOC) [24] of iterative methods to solve a large system of non-linear equations $\mathbf{F}_1(x)$ and $\mathbf{F}_2(x)$ respectively

Accuracy of iterative methods for the large system used in Example 1 and Example 2						
Example 2	D	$MS\alpha_*$	$MS\alpha_1$	$MS\alpha_2$	$MS\alpha_3$	$MS\alpha_4$
$\mathbf{F}_1(x)$	50	0.0	1.1511e-16	0.3512e-15	1.147e-16	1.1567e-15
$\mathbf{F}_1(x)$	75	0.0	1.120e-16	1.3125e-14	1.1130e-16	3.4417e-13
$\mathbf{F}_1(x)$	100	0.0	1.1102e-16	1.2751e-15	1.1102e-16	3.4417e-13
LCOC	100	5.431	5.014	4.916	4.967	5.015
$\mathbf{F}_2(x)$	50	0.0	0.0	0.0	0.0	0.0
$\mathbf{F}_2(x)$	75	0.0	0.0	0.0	0.0	6.7511e-16
$\mathbf{F}_2(x)$	100	0.0	0.0	1.5451e-12	1.9621e-17	6.6147e-16
LCOC	100	5.324	4.912	4.781	5.102	4.991

Example 2: N-Dimensional Problems [23]

Consider

$$\mathbf{F}_2 : f_i(x_i) = x_i^2 - \cos(x_i - 1),\ i = 1, 2, 3, .., m \tag{24}$$

the exact solution of the system Eq. (24) is $\mathbf{X}^* = [1, 1, 1, \ldots, 1]^T$ and taking $\mathbf{X}_0 = [2, 2, 2, \ldots, 2]^T$ as an initial estimate. Tables 4–5, indicates the numerical results of the system of non-linear equations Eq. (24) used.

Table 6: Number of iterations, to solve BVP-I using $MS\alpha_*$, $MS\alpha_1$, $MS\alpha_2$, $MS\alpha_3$, $MS\alpha_4$

Number of iterations of iterative methods for BVP-I used in Example 3-respectively						
BVP	D	$MS\alpha_*$	$MS\alpha_1$	$MS\alpha_2$	$MS\alpha_3$	$MS\alpha_4$
BVP-I	22	3	3	3	3	3

Table 7: Computational time in seconds to solve BVP-I using $MS\alpha_*$, $MS\alpha_1$, $MS\alpha_2$, $MS\alpha_3$, $MS\alpha_4$

Computational time in seconds of iterative methods for solving BVP-1 used in Example 3						
BVP	D	$MS\alpha_*$	$MS\alpha_1$	$MS\alpha_2$	$MS\alpha_3$	$MS\alpha_4$
BVP-I	22	0.8015	0.9134	0.8971	1.6151	0.8915

Table 8: Accuracy of iterative methods $MS\alpha_*$, $MS\alpha_1$, $MS\alpha_2$, $MS\alpha_3$and $MS\alpha_4$ to solve BVP-I

Accuracy of iterative methods for solving BVP-I used in Example 3						
BVP	D	$MS\alpha_*$	$MS\alpha_1$	$MS\alpha_2$	$MS\alpha_3$	$MS\alpha_4$
BVP-I	22	3.3141e-16	2.2661e-15	5.5961e-16	1.5712e-16	1.0081e-15

Application in Differential Equation

Here, we solve some highly non-linear BVPs using the newly constructed iterative method and existing methods in literature to show the dominance efficiency of our methods $MS\alpha_*$ with comparison to $MS\alpha_1$, $MS\alpha_2$, $MS\alpha_3$ and $MS\alpha_4$ respectively.

Example 3: [24,25]

Consider the non-linear boundary value problem (BVP-I):

$$y'' = -\beta(e^y), 0 \le x \le 1 \tag{25}$$

$y(0) = 0$; $y(1) = 0$. The exact solution to the non-linear boundary value problem does not exist therefore for graphical comparison we take the approximate solutions using the shooting method Fig. 16.

By dividing the interval [0,1] into n = 22 equal subinterval as:

$$x_0 = 0 < x_1 < \ldots < x_n = 1; x_{i+1} = x_i + h \text{ and } h = \frac{1}{n}.$$

Assuming $y_0 = y(x_0) = 0, y_1 = y(x_1), \ldots, y_n = y(x_n) = 1$. Using the procedure of finite-difference central approximations of the derivatives i.e.,

$$y''(x_i) = \frac{1}{h^2}\left(y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right) - \frac{h^2}{12}y^{(iv)}(\xi), \text{ for some } \xi \in (x_{i-1}, x_{i+1}) \tag{26}$$

$$y'(x_i) = \frac{1}{2h}\left(y(x_{i+1}) - y(x_{i-1})\right) - \frac{h^2}{6}y^{(iii)}(\xi), \text{ for some } \xi \in (x_{i-1}, x_{i+1}) \tag{27}$$

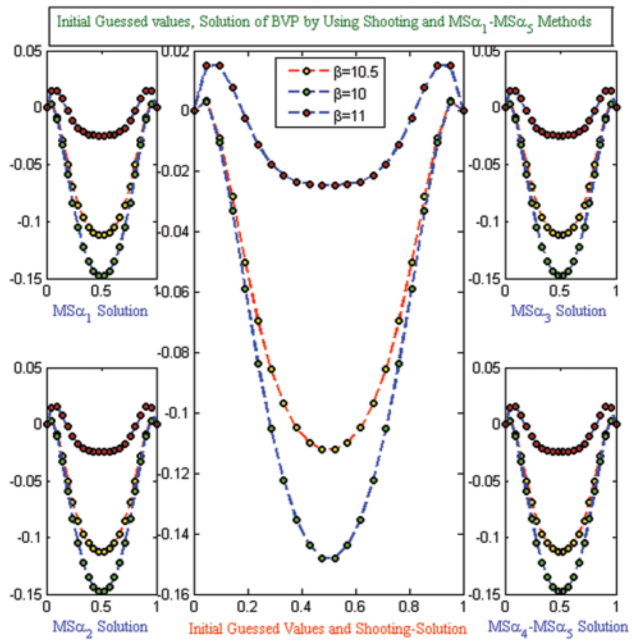

Figure 16: Numerical solution of BVP-1 using shooting methods, $MS\alpha_*$, $MS\alpha_1 - MS\alpha_4$.

In non-linear boundary value problem Eq. (25), we get the following non-linear system of equations:

$$484y_{i+2} - 968y_{i+1} + 848y_i + \beta e^{y_i} = 0, i = 1, 2, \ldots, 22 \tag{28}$$

We chose the following initial approximation

$\mathbf{X0} = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]^T$
The solution to this non-linear boundary value problem up to 5 decimal places is $\mathbf{X}^* =$

$$\begin{bmatrix} 0.00, 0.8363576950\text{e}{-1}, 0.1650251902, 0.2439777869, \\ 0.3202933671, 0.3937628071, 0.4641691297, 0.5312888986, \\ 0.5948939496, 0.6547534661, 0.7106363959, 0.7623141892, \\ 0.8095638206, 0.8521710394, 0.8899337709, 0.9226655761, \\ 0.9501990588, 0.9723891027, 0.9891158112, 1.000287032, \\ 1.005840356, 1.005744498, 1.000000000 \end{bmatrix}^T$$

We solve the nonlinear system of equations Eq. (28) by taking $\beta = 10.5, 10, 11$. Tables 6–8, indicates the numerical results of the BVP-1 used.

5 Conclusion

A precise approach was developed in this paper for constructing iterative schemes. Using Computer Algebra System CAS-symbolic computation with strong speeding iterative numerical schemes, we developed novel efficient numerical iterative methods for solving nonlinear systems of equations. We were prompted to use symbolic computation via multiple programs written in the computer algebra system CAS-Maple due to the fact that the newly derived technique required lengthy and complicated mathematical statements. Maple was used to perform numerical examples of higher-order nonlinear systems of equations as well as to solve some highly nonlinear BVPs. These examples revealed that the newly developed approaches' theoretical order of convergence corresponds to the computational outcomes. In addition to providing visual insight into the convergence behavior of iterative methods, the generation of basins of attraction could also generate qualitative concerns for comparison. It is evident from all Figs. 1–16 and Tables 1–8, that the iterative schemes $MS\alpha_*$ are more effective than $MS\alpha_1$, $MS\alpha_2$, $MS\alpha_3$, $MS\alpha_4$.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the Large Groups Project under grant number RGP. 2/235/43.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Weerakoon and T. G. I. Fernando, "A variant of newton's method with accelerated third-order convergence," *Applied Mathematics Letters*, vol. 13, no. 8, pp. 87–93, 2000.

[2] A. Y. Özban, "Some new variants of newton's method," *Applied Mathematics Letters*, vol. 17, no. 6, pp. 677–682, 2004.

[3] J. Gerlach, "Accelerated convergence in newton's method," *SIAM Review*, vol. 36, no. 2, pp. 272–276, 1994.

[4] J. Young and M. David, "AM ostrowski, solution of equations and systems of equations," *Bulletin of the American Mathematical Society*, vol. 68, no. 4, pp. 306–308, 1962.

[5] A. Cordero and J. R. Torregrosa, "Variants of newton's method for functions of several variables," *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 199–208, 2006.

[6] W. Haijun, "New third-order method for solving systems of nonlinear equations," *Numerical Algorithms*, vol. 50, no. 3, pp. 271–282, 2009.

[7] A. Cordero and J. R. Torregrosa, "On interpolation variants of newton's method for functions of several variables," *Journal of Computational and Applied Mathematics*, vol. 234, no. 1, pp. 34–43, 2010.

[8] M. Frontini and E. Sormani, "Third-order methods from quadrature formulae for solving systems of nonlinear equations," *Applied Mathematics and Computation*, vol. 149, no. 3, pp. 771–782, 2004.

[9] M. T. Darvishi and A. Barati, "A fourth-order method from quadrature formulae to solve systems of nonlinear equations," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 257–261, 2007.

[10] G. Adomian, "Solution of physical problemes by decomposition," *Computers & Mathematics with Applications*, vol. 27, no. 9, pp. 145–154, 1994.

[11] D. K. R. Babajee, M. Z. Dauhoo, M. T. Darvishi and A. Barati, "A note on the local convergence of iterative methods based on adomian decomposition method and 3-node quadrature rule," *Applied Mathematics and Computation*, vol. 200, no. 1, pp. 452–458, 2008.

[12] A. Cordero, E. Martínez and J. R. Torregrosa, "Iterative methods of order four and five for systems of nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 231, no. 2, pp. 541–551, 2009.

[13] J. F. Traub, "Computational complexity of iterative processes," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 167–179, 1972.

[14] S. A. Sariman and I. Hashim, "New optimal newton-householder methods for solving nonlinear equations and their dynamics," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 69–85, 2020.

[15] J. R. Sharma, R. K. Guha and R. Sharma, "An efficient fourth order weighted-newton method for systems of nonlinear equations," *Numerical Algorithms*, vol. 62, no. 2, pp. 307–323, 2013.

[16] A. Singh, "An efficient fifth-order iterative scheme for solving a system of nonlinear equations and PDE," *International Journal of Computing Science and Mathematics*, vol. 11, no. 4, pp. 316–326, 2020.

[17] X. Zhang and J. Tan, “The fifth order of three-step iterative methods for solving systems of nonlinear equations,” *Mathematica Numerica Sinica*, vol. 35, no. 3, pp. 297–304, 2013.
[18] A. Cordero and J. R. Torregrosa, “Variants of newton’s method using fifth order quadrature formulas,” *Applied Mathematics and Computation*, vol. 190, no. 1, pp. 686–698, 2007.
[19] A. Naseem, M. Rehman and T. Abdeljawad, “Computational methods for non-linear equations with some real-world applications and their graphical analysis,” *Intelligent Automation & Soft Computing*, vol. 30, no. 3, pp. 805–819, 2021.
[20] Y. M. Chu, N. Rafiq, M. Shams, S. Akram, N. A. Mir *et al.,* “Computer methodologies for the comparison of some efficient derivative free simultaneous iterative methods for finding roots of non-linear equations,” *Computers, Materials & Continua*, vol. 66, no. 1, pp. 275–290, 2021.
[21] O. S. Solaiman and I. Hashim, “Optimal eighth-order solver for nonlinear equations with applications in chemical engineering,” *Intelligent Automation & Soft Computing*, vol. 27, no. 2, pp. 379–390, 2021.
[22] M. Shams, N. Rafiq, N. A. Mir, B. Ahmad, S. Abbasi *et al.,* “On computer implementation for comparison of inverse numerical schemes for non-linear equations,” *CSSE-Computer Systems Science & Engineering*, vol. 36, no. 3, pp. 493–507, 2021.
[23] M. Q. Khirallah and M. A. Hafiz, “Solving system of nonlinear equations using family of jarratt methods,” *International Journal of Differential Equations and Applications*, vol. 12, no. 2, pp. 69–83, 2013.
[24] R. L. Burden and J. D. Faires, “Boundary-value problems for ordinary differential equations,” in *Numerical Analysis*, 9th ed., vol. 1. Boston, MA 02210, USA: Belmount Thomson Brooks/Cole Press, pp. 1–863, 2005.
[25] Y. Lin, “Enclosing all solutions of two point boundary value problems for ODEs,” *Computer and Chemical Engineering*, vol. 32, no. 8, pp. 1714–1725, 2008.