

Computer Methodologies for the Comparison of Some Efficient Derivative Free Simultaneous Iterative Methods for Finding Roots of Non-Linear Equations

Yuming Chu¹, Naila Rafiq², Mudassir Shams^{3,*}, Saima Akram⁴, Nazir Ahmad Mir³ and Humaira Kalsoom⁵

¹Department of Mathematics, Huzhou University, Huzhou, 313000, China

²Department of Mathematics, National University of Modern Languages, Islamabad, Pakistan

³Department of Mathematics and Statistics, Riphah International University I-14, Islamabad, 44000, Pakistan

⁴Center for Advanced Studies in Pure and Applied Mathematics Bahauddin Zakariya University, Multan, Pakistan

⁵School of Mathematical Sciences, Zhejiang University, Hangzhou, 310027, China

*Corresponding Author: Mudassir Shams. Email: mudassir4shams@gmail.com

Received: 04 June 2020; Accepted: 02 July 2020

Abstract: In this article, we construct the most powerful family of simultaneous iterative method with global convergence behavior among all the existing methods in literature for finding all roots of non-linear equations. Convergence analysis proved that the order of convergence of the family of derivative free simultaneous iterative method is nine. Our main aim is to check out the most regularly used simultaneous iterative methods for finding all roots of non-linear equations by studying their dynamical planes, numerical experiments and CPU time-methodology. Dynamical planes of iterative methods are drawn by using MATLAB for the comparison of global convergence properties of simultaneous iterative methods. Convergence behavior of the higher order simultaneous iterative methods are also illustrated by residual graph obtained from some numerical test examples. Numerical test examples, dynamical behavior and computational efficiency are provided to present the performance and dominant efficiency of the newly constructed derivative free family of simultaneous iterative method over existing higher order simultaneous methods in literature.

Keywords: Non-linear equation; iterative method; simultaneous method; basins of attractions; computational efficiency

1 Introduction

One of the ancient problems in mathematics is the estimations of roots of non-linear equation

$$f(\eta) = 0. \quad (1)$$

There are number of applications of non-linear equation in science and engineering. Newton's method is a numerical iterative scheme which finds a single root at a time. The simultaneous iterative method (SIM) such as, Weirstrass [1] method is used to find all the distinct roots of Eq. (1). The iterative methods for finding single root of non-linear polynomial equation have been studied by [2–4] and many others. On



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the other hand, there are lot of numerical iterative methods devoted to approximate all roots of Eq. (1) simultaneously (see, e.g., [1,5–8] and the references therein). The SIM are popular as compared to single root finding methods due to their wider range of convergence, reliability and their applications for parallel computing as well. Further details on SIM, their convergence analysis, efficiency and parallel implementations can be seen in [9,10–13] and references cited there in. The main objective of this article is to construct SIM which have more efficient and higher convergence order for approximating all distinct roots of Eq. (1). For the analysis and comparison of convergence behavior of simultaneous iterative methods, we use the techniques of dynamical plane with CAS MATLAB (R2011b).

2 Constructions of Simultaneous Method

Here, we construct a ninth order derivative free simultaneous method which is more efficient than the similar methods existing in literature.

2.1 Construction of Simultaneous Methods for Distinct Roots

Consider eighth order derivative free Kung–Traub’s [4] family of iterative method (abbreviated as KF):

$$\begin{aligned}\sigma^{(t)} &= \eta^{(t)} - \frac{\alpha f(\eta^{(t)})^2}{f(v^{(t)}) - f(\eta^{(t)})}, \\ u^{(t)} &= \sigma^{(t)} - \left(\frac{f(\sigma^{(t)})f(v^{(t)})}{(f(v^{(t)}) - f(\sigma^{(t)})) \left(\frac{f(\eta^{(t)}) - f(\sigma^{(t)})}{\eta^{(t)} - \sigma^{(t)}} \right)} \right), \\ z^{(t)} &= u^{(t)} - \left(\frac{f(\sigma^{(t)})f(v^{(t)}) \left(\sigma^{(t)} - \eta^{(t)} + \frac{f(\eta^{(t)})}{\frac{f(\eta^{(t)}) - f(u^{(t)})}{\eta^{(t)} - u^{(t)}}} \right)}{(f(\sigma^{(t)}) - f(u^{(t)})) (f(v^{(t)}) - f(u^{(t)}))} \right) + \left(\frac{f(\sigma^{(t)})}{\frac{f(\sigma^{(t)}) - f(u^{(t)})}{\sigma^{(t)} - u^{(t)}}} \right), \text{ where } v^{(t)} = \eta^{(t)} + \alpha f(\eta^{(t)}).\end{aligned}$$

Using well known Weierstrass [1] method, abbreviated as (WKD), we have:

$$\sigma_i^{(t)} = \eta_i^{(t)} - \frac{f(\eta_i^{(t)})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\eta_i^{(t)} - \eta_j^{(t)})} \quad (i, j = 1, \dots, n) \quad (2)$$

Replacing $\eta_j^{(t)}$ by $z_j^{(t)}$ in Eq. (2), we get new simultaneous iterative method (abbreviated as SIM1),

$$\sigma_i^{(t)} = \eta_i^{(t)} - \frac{f(\eta_i^{(t)})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\eta_i^{(t)} - z_j^{(t)})} \quad (i, j = 1, \dots, n) \quad (3)$$

where

$$z_j^{(t)} = u_j^{(t)} - \left(\frac{f(\sigma_j^{(t)})f(v_j^{(t)}) \left(\sigma_j^{(t)} - \eta_j^{(t)} + \frac{f(\eta_j^{(t)})}{f(\eta_j^{(t)}) - f(u_j^{(t)})} \right)}{(f(\sigma_j^{(t)}) - f(u_j^{(t)})) (f(v_j^{(t)}) - f(u_j^{(t)}))} \right) + \left(\frac{f(\sigma_j^{(t)})}{f(\sigma_j^{(t)}) - f(u_j^{(t)})} \right),$$

$$u_j^{(t)} = \sigma_j^{(t)} - \left(\frac{f(\sigma_j^{(t)})f(v_j^{(t)})}{(f(v_j^{(t)}) - f(\sigma_j^{(t)})) \left(\frac{f(\eta_j^{(t)}) - f(\sigma_j^{(t)})}{\eta_j^{(t)} - \sigma_j^{(t)}} \right)} \right), \quad \sigma_j^{(t)} = \eta_j^{(t)} - \frac{\alpha f(\eta_j^{(t)})^2}{f(v_j^{(t)}) - f(\eta_j^{(t)})}$$

and $v_j^{(t)} = \eta_j^{(t)} + \alpha f(\eta_j^{(t)})$.

Thus, we have a new derivative free family of simultaneous method Eq. (3), abbreviated as SIM1, for approximating all the distinct roots of Eq. (1).

2.2 Convergence Analysis

Here, we discuss the convergence of iterative method SIM1:

Theorem: Let $\zeta_1, \zeta_2, \dots, \zeta_n$ be n simple roots of Eq. (1). If $\eta_1^{(0)}, \eta_2^{(0)}, \dots, \eta_n^{(0)}$ be the sufficiently close initial approximations to actual roots, then the order of convergence of SIM1 is nine.

Proof: Let $\varepsilon_i = \eta_i - \zeta_i, \varepsilon'_i = \sigma_i - \zeta_i$, be the errors in η_i and σ_i approximations respectively. For simplification, we omit iteration index t . From SIM1, we have:

$$\sigma_i - \zeta_i = \eta_i - \zeta_i - w^*(\eta_i), \text{ where } w^*(\eta_i) = \frac{f(\eta_i^{(t)})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\eta_i^{(t)} - z_j^{(t)})}.$$

$$\varepsilon'_i = \varepsilon_i - \varepsilon_i \frac{w^*(\eta_i)}{\varepsilon_i} = \varepsilon_i(1 - S_i), \tag{4}$$

where

$$S_i = \frac{w^*(\eta_i)}{\varepsilon_i} = \frac{f(\eta_i)}{\varepsilon_i \prod_{\substack{j=1 \\ j \neq i}}^n (\eta_i - v_j)} = \frac{\prod_{j=1}^n (\eta_i - \zeta_j)}{\varepsilon_i \prod_{\substack{j=1 \\ j \neq i}}^n (\eta_i - z_j)} = \frac{(\eta_i - \zeta_i) \prod_{\substack{j=1 \\ j \neq i}}^n (\eta_i - \zeta_j)}{\varepsilon_i \prod_{\substack{j=1 \\ j \neq i}}^n (\eta_i - z_j)} = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{(\eta_i - \zeta_j)}{(\eta_i - z_j)} \tag{5}$$

Now, if ζ_i is a simple root, then for a small enough ε , $|\eta_i - v_j|$ is bounded away from zero, and so

$$\frac{\eta_i - \zeta_j}{\eta_i - z_j} = 1 + \frac{z_j - \zeta_j}{\eta_i - z_j} = 1 + O(\varepsilon^8), \tag{6}$$

where $z_j - \zeta_j = \tilde{O}(\varepsilon^8)$, see [4]:

$$\prod_{\substack{j=1 \\ j \neq i}}^n \frac{(\eta_i - \zeta_j)}{(\eta_i - v_j)} = (1 + \tilde{O}(\varepsilon^8))^{n-1} = 1 + (n-1)\tilde{O}(\varepsilon^8) = 1 + \tilde{O}(\varepsilon^8). \quad (7)$$

$$S_i - 1 = \tilde{O}(\varepsilon^8).$$

Thus, Eq. (4) gives:

$$\varepsilon'_i = \tilde{O}(\varepsilon)^9. \quad (8)$$

Hence, the theorem is proved.

3 Dynamical Studies of KF, SIM1 and SPJ1

In this section, we discuss the dynamical study of KF, SIM1 and [14] method (abbreviated as SPJ1). We have discussed the dynamical behavior of simultaneous methods to show global convergence as dynamical planes of single root finding methods may have divergence regions which do not exist in simultaneous methods. Let us recall some basic concepts of this study. For more details on the dynamical behavior of the iterative methods one can consult [2] and [15]. Taking a rational map $\mathfrak{R}_f: C \rightarrow C$, where C is a complex plane, the orbit $\eta_0 \in C$ defines a set such as, $orb(\eta) = \{\eta_0, \mathfrak{R}_f(\eta_0), \mathfrak{R}_f^2(\eta_0), \dots, \mathfrak{R}_f^m(\eta_0), \dots\}$. The convergence $orb(\eta) \rightarrow \eta^*$ is understood in the sense if $\lim_{k \rightarrow \infty} \mathfrak{R}(\eta) = \eta^*$ exist. A point $\eta_0 \in C$ known as attracting, if $|\mathfrak{R}'(\eta)| < 1$. An attracting point $\eta^* \in C$ defines basins of attraction $\mathfrak{R}(\eta^*)$ as the set of starting points whose orbit tends to η^* . To generate basins of attraction, we take grid 2000×2000 of square $[-2.5 \times 2.5]^2 \in C$. To each root of Eq. (1), we assign a color to which the corresponding orbit of the iterative methods starts and converges to a fixed point. Take color map as Jet. We take $|f(\eta_i)| < 10^{-5}$ and maximum numbers of iterations are chosen as 5 due to wider convergence region of simultaneous methods. Dark black points are assigned, if the orbit of the iterative methods does not converge to root after 5 iterations. We obtained basins of attractions for the following three test function $f_1(\eta) = \eta^4 + \eta^2 + \eta - 1$ and $f_2(\eta) = \eta^6 + \eta - 1$ and $f_3(\eta) = \sin(\frac{\eta-1}{2}) \sin(\frac{\eta-2}{2}) \sin(\frac{\eta-2.5}{2})$. The root of $f_1(\eta)$ are $0.2 + 1.3i, 0.2 - 1.3i, -1, 0.5$, roots of $f_2(\eta)$ are $-1.1, -0.4 + 1i, -0.4 - 1i, 0.6 + 0.7i, 0.6 - 0.7i, 0.7$ and root of $f_3(\eta)$ are $1, 2, 2.5$ correct up to 1-decimal place. Brightness in color in Figs. 1–9 means less number of iterations. Finally, in Fig. 10, we present Elapsed time of basins of attraction corresponding to iterative map KF, SIM1 and SPJ1 using tic-toc command in MATLAB (R2011b).

4 Computational Aspects

Here, we discuss the computational efficiency and convergence behavior of the [14] method (abbreviated as SPJ1) and the new method SIM1. As presented in [14], the efficiency index (\bar{E}) is used to estimate the efficiency of iterative method as:

$$\bar{E}(m) = \frac{\log r}{G}, \quad (9)$$

where G in [14], denotes the cost of computation and r , the order of convergence.

$$G = G(m) = w_{as}AS_m + w_mM_m + w_dD_m. \quad (10)$$

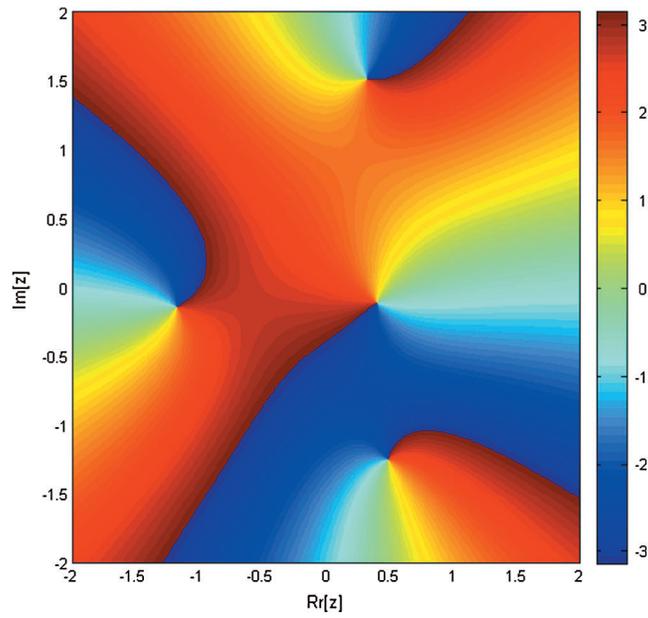


Figure 1: Basin of attraction of iterative method SIM1 for non-linear equation $f_1(\eta) = \eta^4 + \eta^2 + \eta - 1$

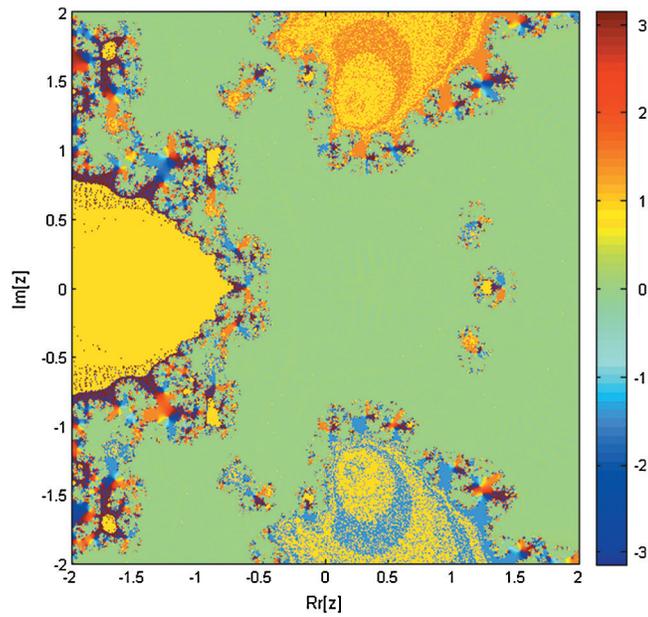


Figure 2: Basin of attraction of iterative method KF for non-linear equation $f_1(\eta) = \eta^4 + \eta^2 + \eta - 1$

Thus, Eq. (9) becomes:

$$\bar{E}(m) = \frac{\log r}{w_{as}AS_m + w_mM_m + w_dD_m}. \tag{11}$$

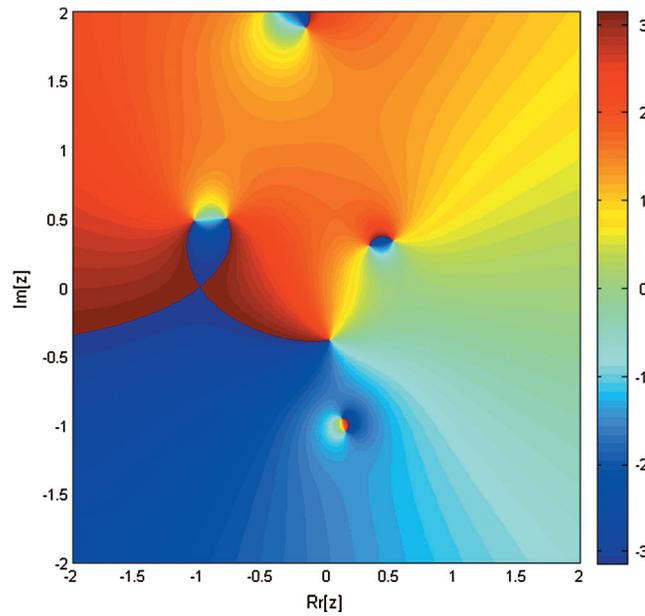


Figure 3: Basin of attraction of iterative method SPJ1 for non-linear equation $f_1(\eta) = \eta^4 + \eta^2 + \eta - 1$

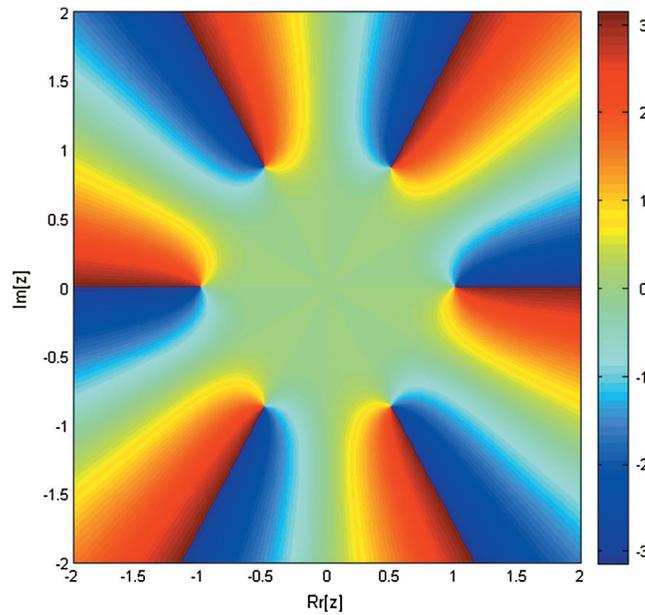


Figure 4: Basin of attraction of iterative method SIM1 for non-linear equation $f_2(\eta) = \eta^6 + \eta - 1$

Using Eq. (11) and data in Tab. 1, we find the percentage ratio $\Omega(SIM1, SPJ1)$ [14] as:

$$\Omega(SIM1, SPJ1) = \left(\frac{\bar{E}(SIM1)}{\bar{E}(SPJ1)} - 1 \right) \times 100 \quad (12)$$

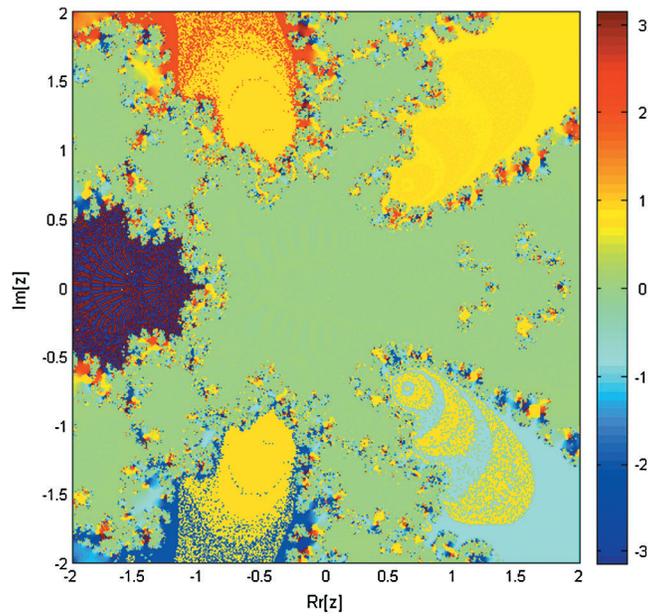


Figure 5: Basin of attraction of iterative method KF for non-linear equation $f_2(\eta) = \eta^6 + \eta - 1$

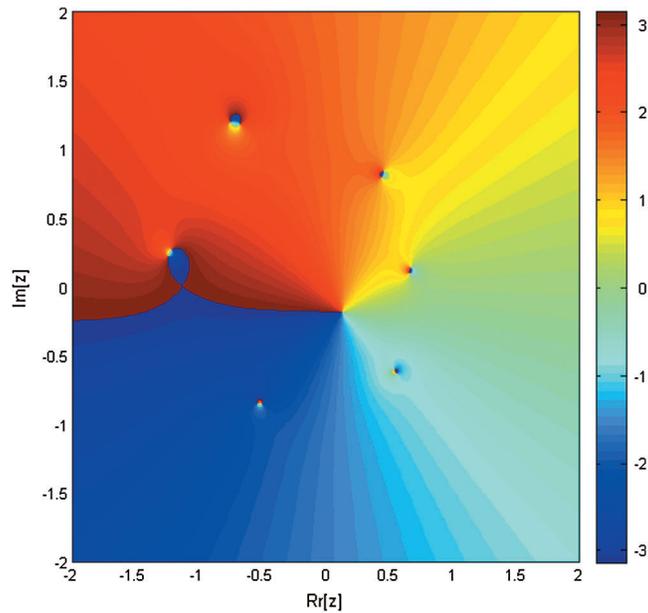


Figure 6: Basin of attraction of iterative method SPJ1 for non-linear equation $f_2(\eta) = \eta^6 + \eta - 1$

$$\Omega(SPJ1, SIM1) = \left(\frac{\bar{E}(SPJ1)}{\bar{E}(SIM1)} - 1 \right) \times 100 \tag{13}$$

Figs. 11–12, graphically illustrates these percentage ratios. Figs. 11–12, clearly show that the newly constructed simultaneous method SIM1 is more efficient as compared to Petkovic method (SPJ1).

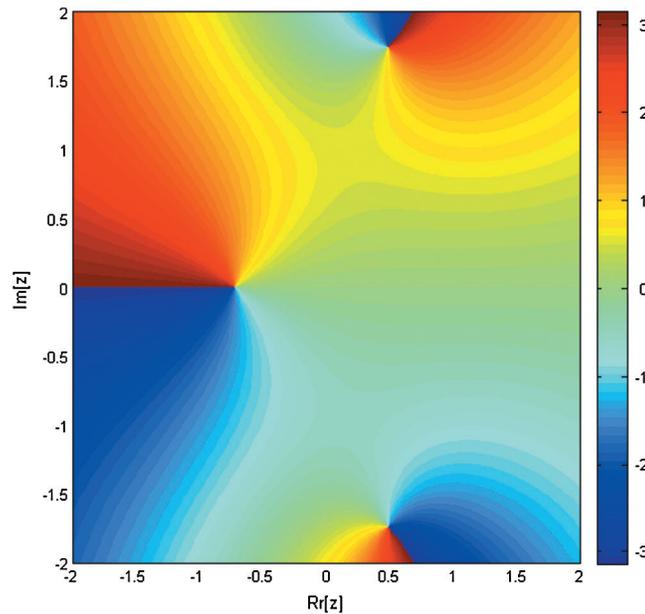


Figure 7: Basin of attraction of iterative method SIM1 for non-linear equation $f_3(\eta) = \sin\left(\frac{\eta-1}{2}\right) \sin\left(\frac{\eta-2}{2}\right) \sin\left(\frac{\eta-2.5}{2}\right)$

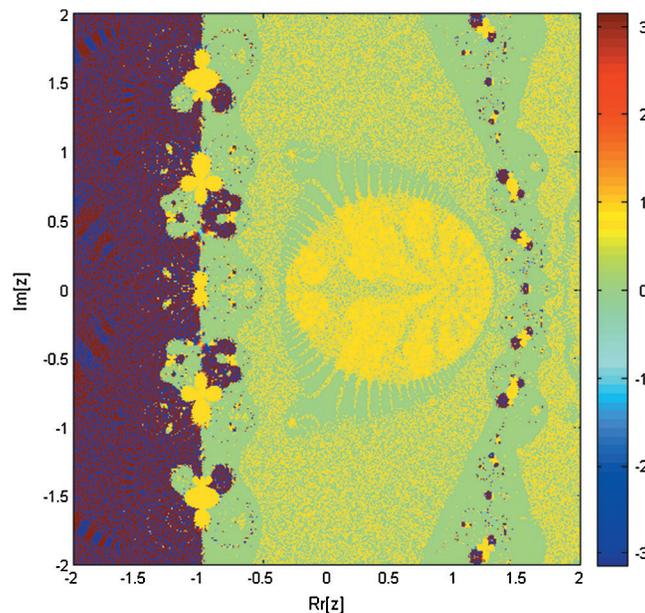


Figure 8: Basin of attraction of iterative method KF for non-linear equation $f_3(\eta) = \sin\left(\frac{\eta-1}{2}\right) \sin\left(\frac{\eta-2}{2}\right) \sin\left(\frac{\eta-2.5}{2}\right)$

5 Numerical Results

Here, some numerical test examples are considered in order to show the performance of simultaneous ninth order derivative free method SIM1. We compare our method with [14] method (SPJ1) of convergence order ten for distinct roots. All the numerical calculations are done by using Maple 18 with 64 digits floating point arithmetic. We take $\epsilon = 10^{-30}$ as tolerance and use as a stopping criteria.

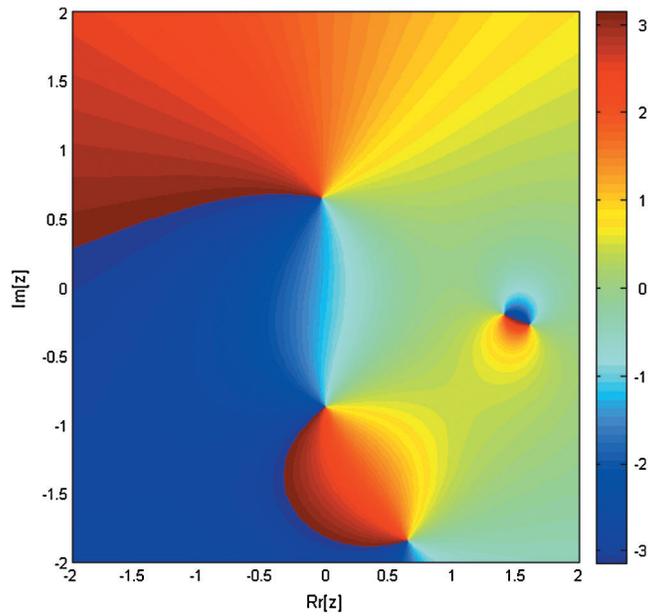


Figure 9: Basin of attraction of iterative method SPJ1 for non-linear equation $f_3(\eta) = \sin(\frac{\eta-1}{2}) \sin(\frac{\eta-2}{2}) \sin(\frac{\eta-2.5}{2})$

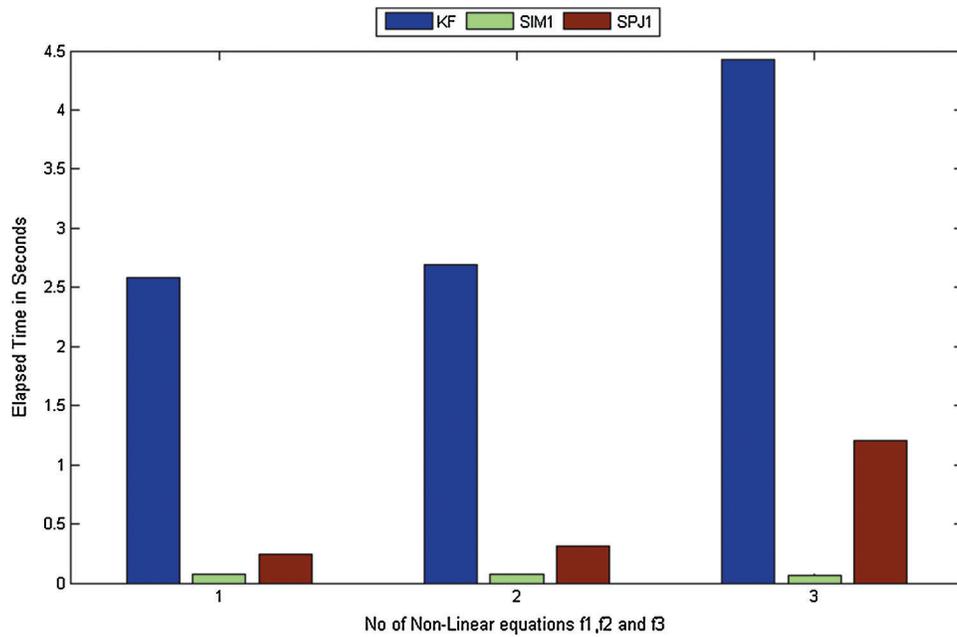


Figure 10: Elapsed time of iterative methods SIM1, KF, SPJ1 in seconds for non-linear function $f_1(\eta), f_2(\eta)$ and $f_3(\eta)$ respectively

Table 1: The number of basic arithmetic operations

Methods	AS_m	M_m	D_m
SIM1	$20 m^2 + O(m)$	$8 m^2 + O(m)$	$2 m^2 + O(m)$
SPJ1	$22 m^2 + O(m)$	$18 m^2 + O(m)$	$7 m^2 + O(m)$

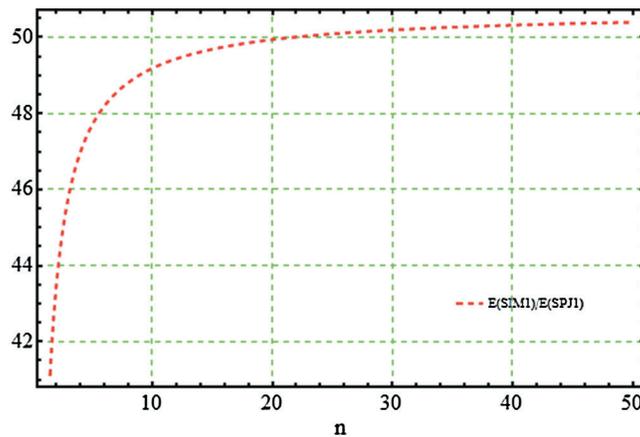


Figure 11: Computation efficiency of SIM1 w.r.t. SPJ1

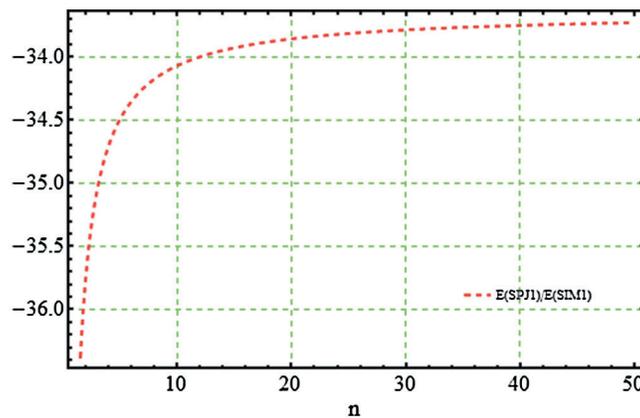


Figure 12: Computation efficiency of SPJ1 w.r.t. SIM1

$$\check{e}_i = \left\| \eta_i^{(t+1)} - \eta_i^{(t)} \right\|_2$$

Tests examples from [16–18] are provided in [Tabs. 2–3](#). In all Tables, CO denotes the order of convergence, α , parameter valued in SIM1, n , the number of iterations and *CPU*, execution time in seconds. [Figs. 13–16](#), show that residue fall of the methods *SIM1* and *SPJ1* for the numerical test examples 1 – 2, shows that method *SIM1* is more efficient as compared to *SPJ1*. We observe that numerical results of the method *SIM1* are comparable with *SPJ1* method on same number of iteration.

Table 2: Simultaneous determination of all roots $f_4(\eta)$

Method	CO	CPU	α	n	\check{e}_1	\check{e}_2	\check{e}_3	\check{e}_4
SPJ1	10	0.172	–	5	1.6e–10	3.0e–11	1.8e–12	1.5e–9
SIM1	9	0.094	–0.9212	5	5.6e–12	1.9e–11	7.4e–12	1.8e–11
SIM1	9	0.109	–0.9111	5	7.6e–12	3.4e–11	1.4e–13	1.2e–10

Table 3: Simultaneous determination of all roots $f_5(\eta)$

Method	CO	CPU	α	n	\check{e}_1	\check{e}_2	\check{e}_3
SPJ1	10	0.125	–	4	6.3e-10	2.4e-5	3.3e-9
SIM1	9	0.078	9/101	4	2.2e-11	1.8e-5	7.2e-10
SIM1	9	0.081	-12/100	4	8.3e-11	8.7e-5	3.6e-10

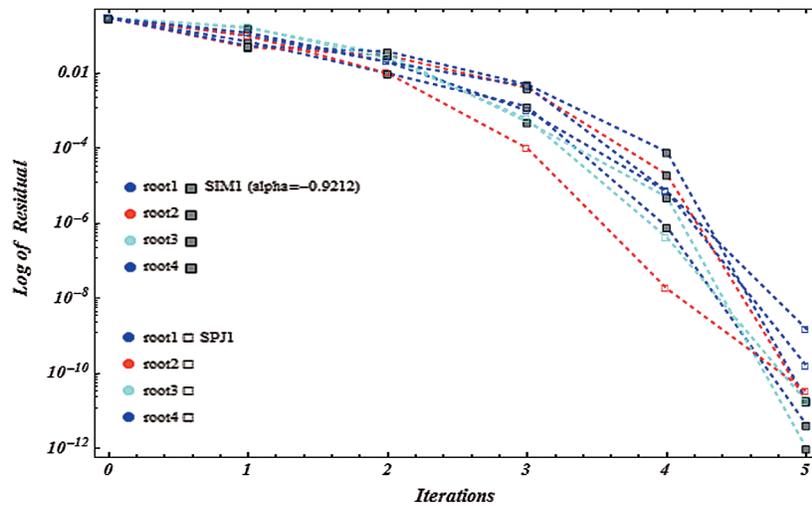


Figure 13: Shows residual graph of SIM1 ($\alpha = -0.9212$) and SPJ1 for non-linear function $f_4(\eta)$

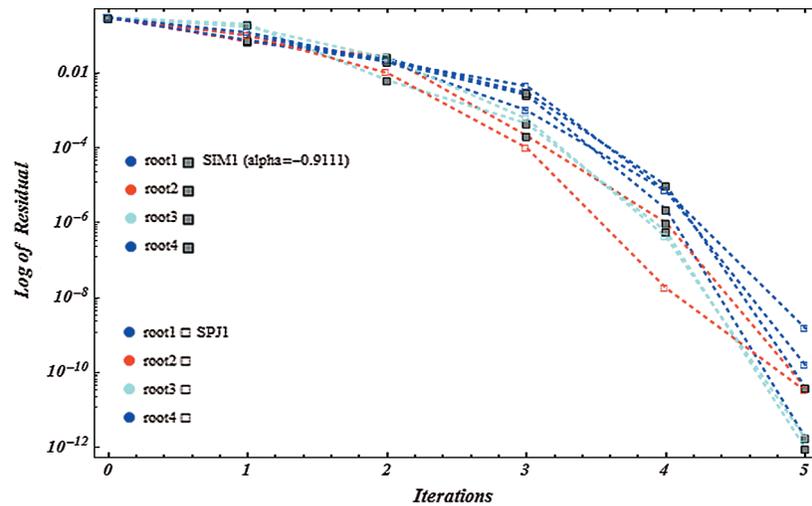


Figure 14: Shows residual graph of SIM1 ($\alpha = -0.9111$) and SPJ1 for non-linear function $f_4(\eta)$

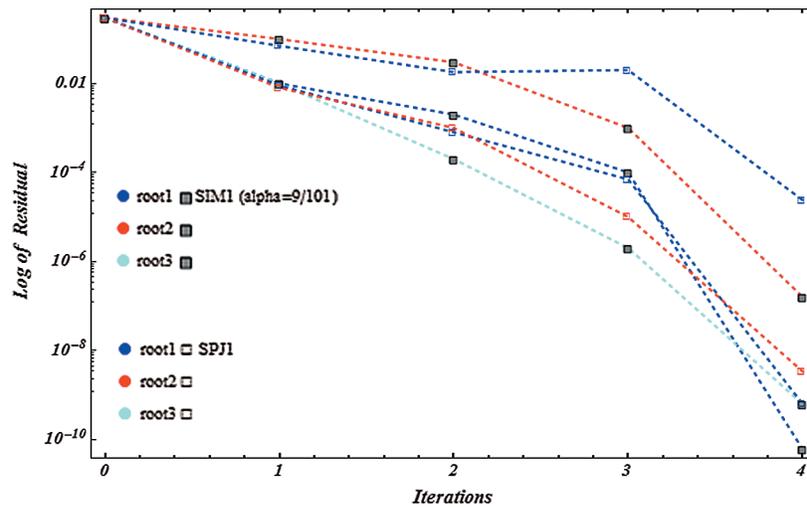


Figure 15: Shows residual graph of SIM1 ($\alpha = \frac{9}{101}$) and SPJ1 for non-linear function $f_5(\eta)$

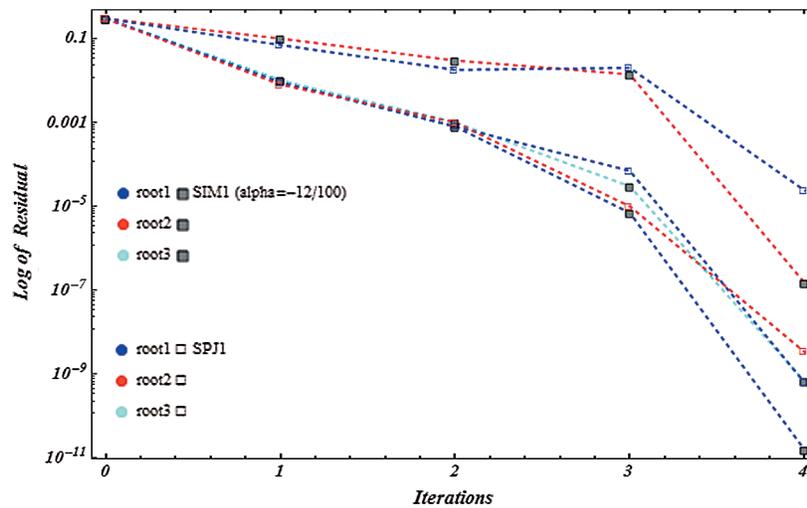


Figure 16: Shows residual graph of SIM1 ($\alpha = -\frac{12}{100}$) and SPJ1 for non-linear function $f_5(\eta)$

We also calculate the CPU execution time, as all the calculations are done using Maple 18 on (Processor Intel(R) Core(TM) i3-3110m CPU@2.4 GHz with 64-bit Operating System). We observe from Tables that CPU time of the methods SIM1 is comparable or better than method SPJ1, showing the efficiency of our family of derivative free methods SIM1 as compared to them.

Algorithm for simultaneous iterative method

Step 1: Given $\eta_1^{(0)}, \eta_2^{(0)}, \eta_3^{(0)}, \dots, \eta_n^{(0)}$ for $t = 0$, such that

$$\sigma_i^{(t)} = \eta_i^{(t)} - \frac{f(\eta_i^{(t)})}{\prod_{\substack{j=1 \\ j \neq i}}^n (\eta_i^{(t)} - z_j^{(t)})}, \quad (i, j = 1, 2, \dots, n),$$

where $z_j^{(t)} = u_j^{(t)} - \left(\frac{f(\sigma_j^{(t)})f(v_j^{(t)}) \left(\sigma_j^{(t)} - \eta_j^{(t)} + \frac{f(\eta_j^{(t)})}{f(\eta_j^{(t)}) - f(u_j^{(t)})} \right)}{(f(\sigma_j^{(t)}) - f(u_j^{(t)})) (f(v_j^{(t)}) - f(u_j^{(t)}))} \right) + \left(\frac{f(\sigma_j^{(t)})}{\sigma_j^{(t)} - u_j^{(t)}} \right),$

and $u_j^{(t)} = \sigma_j^{(t)} - \left(\frac{f(\sigma_j^{(t)})f(v_j^{(t)})}{(f(v_j^{(t)}) - f(\sigma_j^{(t)})) \left(\frac{f(\eta_j^{(t)}) - f(\sigma_j^{(t)})}{\eta_j^{(t)} - \sigma_j^{(t)}} \right)} \right), \sigma_j^{(t)} = \eta_j^{(t)} - \frac{\alpha f(\eta_j^{(t)})^2}{f(v_j^{(t)}) - f(\eta_j^{(t)})}, v_j^{(t)} = \eta_j^{(t)} + \alpha f(\eta_j^{(t)}).$

Step 2: Set $\eta_i^{(t+1)} = \sigma_i^{(t)}$

Step 3: For a given $\epsilon > 0$, if $\|\eta_i^{(t+1)} - \eta_i^{(t)}\|_2 < \epsilon$, then stop.

Step 4: Set $t = t + 1$ and go to Step 1.

Example 1 [18]:

Consider

$$f_4(\eta) = e^{\eta(\eta-1)(\eta-2)(\eta-3)} - 1 \tag{14}$$

with exact roots:

$$\zeta_1 = 0, \zeta_2 = 1, \zeta_3 = 2, \zeta_4 = 3.$$

The initial estimates have been taken as:

$$\eta_1^{(0)} = 0.1, \eta_2^{(0)} = 0.8, \eta_3^{(0)} = 1.8, \eta_4^{(0)} = 2.9,$$

Example 2 [17]:

Consider

$$f_5(\eta) = \eta^3 + 5\eta^2 - 4\eta - 20 + \cos(\eta^3 + 5\eta^2 - 4\eta - 20) - 1 \tag{15}$$

with exact roots are $\zeta_1 = -5, \zeta_2 = -2, \zeta_3 = 2.$

The initial estimates have been taken as:

$$\eta_1^{(0)} = -5.1, \eta_2^{(0)} = -1.8, \eta_3^{(0)} = 1.9.$$

Example 3 [16]:

The acidity of a saturated solution of magnesium hydroxide in hydrochloric acid HCl is given by

$$\frac{3.64 \times 10^{-11}}{[H_3O^+]} = [H_3O^+] + 3.6 \times 10^{-4} \quad (16)$$

for the hydronium ion concentration $[H_3O^+]$. If we set $\eta = 10^4[H_3O^+]$, we obtained the following non-linear equation

$$f_6(\eta) = \eta^3 + 3.6\eta^2 - 36.4 \quad (17)$$

with exact roots are 2.4, $-3.0 \pm 2.3i$ up to one decimal places. The initial estimates have been taken as:

$$\eta_1^{(0)} = 2.45, \eta_2^{(0)} = -3.0261 + 2.3834i, \eta_3^{(0)} = -3.0261 - 2.3834i.$$

Table 4: Simultaneous determination of all roots $f_6(\eta)$

Method	CO	CPU	α	n	$\bar{\epsilon}_1$	$\bar{\epsilon}_2$	$\bar{\epsilon}_3$
SPJ1	10	0.047	—	4	2.6e-25	480.5	480.5
SIM1	9	0.031	-0.8181	4	4.1e-25	1.3e-5	2.7e-5
SIM1	9	0.030	-0.05	4	3.1e-25	3.1e-5	4.1e-6

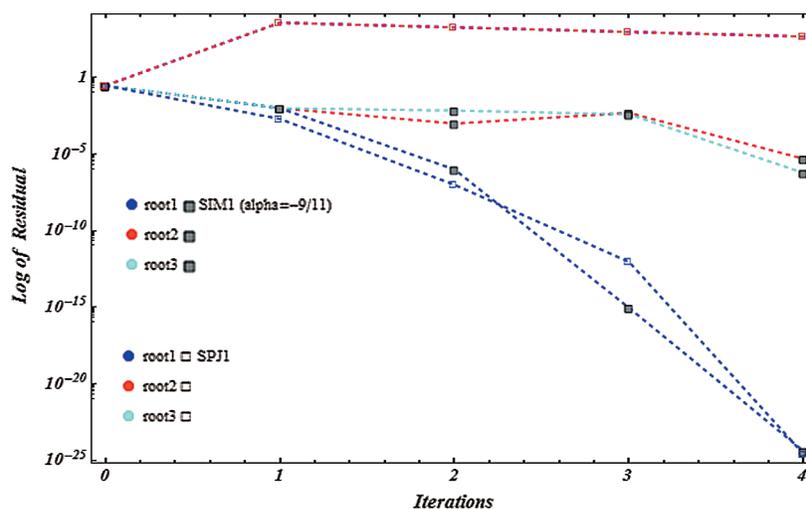


Figure 17: Shows residual graph of SIM1 ($\alpha = -\frac{9}{11}$) and SPJ1 for non-linear function $f_6(\eta)$

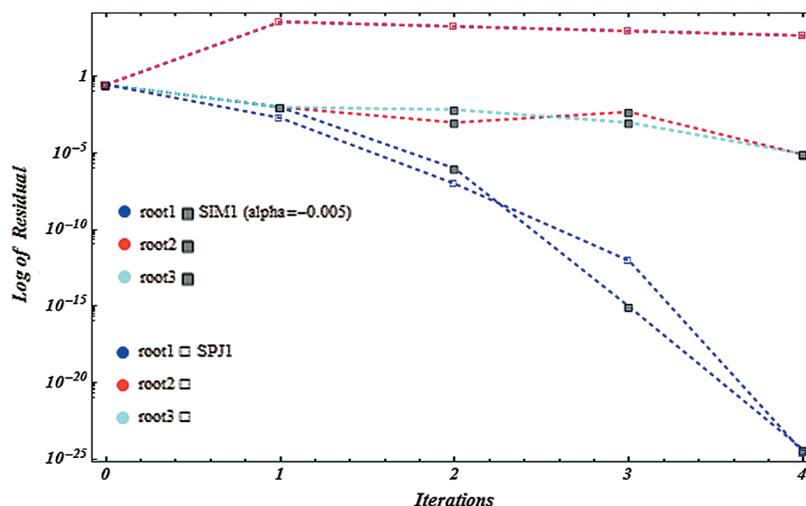


Figure 18: Shows residual graph of SIM1 ($\alpha = -0.005$) and SPJ1 for non-linear function $f_6(\eta)$

6 Conclusions

We have developed here derivative free family of simultaneous methods of order nine for determining all the roots of non-linear equations. It must be pointed out that so far there exists derivative free method of order four only in the literature. We have made here comparison with method SPJ1 of order 10 involving derivative. The dynamical behavior/basins of attractions of our family of simultaneous methods SIM1 is also discussed here to show the global convergence. An example of single root finding derivative free method of order 8 of King–Traub is discussed to show that the single root finding methods may have divergence region. The computational efficiency of our method SIM1 is very large as compare to the method SPJ1 as given in Tabs. 2–4, which is also obvious from Figs. 11–12. We have made the numerical comparison with SPJ1 method. From Tabs. 2–4 and Figs. 1, 4, 7, 13–18, we observe that our numerical results are comparable or better in term of absolute error, number of iterations and CPU time and for log of residual graphs and lapsed time of dynamical planes.

Acknowledgement: The work is supported by the Natural Science Foundation of China (Grant Nos. 61673169, 11301127, 11701176, 11626101, and 11601485) and The Natural Science Foundation of Huzhou City (Grant No. 2018YZ07).

Funding Statement: The article processing charges (APC) will be paid by Natural Science Foundation and Natural Science Foundation of Huzhou City, China.

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

References

- [1] P. D. Proinov and M. D. Petkova, “A new semilocal convergence theorem for the Weierstrass method for finding zeros of a polynomial simultaneously,” *Journal of Complexity*, vol. 30, no. 3, pp. 366–380, 2014.
- [2] F. I. Chicharro, A. Cordero, N. Garrido and J. R. Torregrosa, “Stability and applicability of iterative methods with memory,” *Journal of Mathematical Chemistry*, vol. 57, no. 5, pp. 1282–1300, 2018.
- [3] C. Chun and M. Y. Lee, “A new optimal eight-order family of iterative methods for the solution of nonlinear equations,” *Applied Mathematics and Computation*, vol. 223, pp. 506–519, 2013.

- [4] T. Lotfi, S. Sharifi, M. Salimi and S. Siegmund, "A new class of three-point methods with optimal convergence order eight and its dynamics," *Numerical Algorithms*, vol. 68, no. 2, pp. 261–288, 2015.
- [5] V. K. Kyncheva, V. V. Yotov and S. I. Ivanov, "Convergence of Newton, Halley and Chebyshev iterative methods as methods for simultaneous determination of multiple polynomial zeros," *Applied Numerical Mathematics*, vol. 112, pp. 146–154, 2017.
- [6] P. D. Proinov, S. I. Ivanov and M. S. Petkovi'c, "On the convergence of Gander's type family of iterative methods for simultaneous approximation of polynomial zeros," *Applied Mathematics and Computation*, vol. 349, pp. 168–183, 2019.
- [7] N. A. Mir, M. Shams, N. Rafiq, S. Akram and M. Rizwan, "Derivative free iterative simultaneous method for finding distinct roots of polynomial equation," *Alexandria Engineering Journal*, vol. 59, no. 3, pp. 1629–1636, 2020.
- [8] N. A. Mir, R. Muneer and I. Jabeen, "Some families of two-step simultaneous methods for determining zeros of non-linear equations," *ISRN Applied Mathematics*, vol. 2011, pp. 1–11, 2011.
- [9] F. Peng, Q. Long, Z. X. Lin and M. Long, "A reversible watermarking for authenticating 2D CAD engineering graphics based on iterative embedding and virtual coordinates," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 26885–26905, 2019.
- [10] P. D. Proinov, S. I. Ivanov and M. S. Petkovi'c, "On the convergence of Gander's type family of iterative methods for simultaneous approximation of polynomial zeros," *Applied Mathematics and Computation*, vol. 349, pp. 168–183, 2019.
- [11] P. D. Proinov and S. I. Ivanov, "On the convergence of Halley's method for multiple polynomial zeros," *Mediterranean Journal of Mathematics*, vol. 12, no. 2, pp. 555–572, 2015.
- [12] F. Yu, L. Liu, L. Xiao, K. L. Li and S. Cai, "A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function," *Neurocomputing*, vol. 350, pp. 108–116, 2019.
- [13] L. L. Zhou, F. Tan, F. Yu and W. Liu, "Cluster synchronization of two-layer nonlinearly couple multiplex networks with multi-links and time delays," *Neurocomputing*, vol. 359, pp. 264–275, 2019.
- [14] M. S. Petkovic, L. D. Petkovic and J. Džunic, "On an efficient simultaneous method for finding polynomial zeros," *Applied Mathematics Letters*, vol. 28, pp. 60–65, 2014.
- [15] F. Chicharro, A. Cordero, J. M. Gutierrez and J. R. Torregrosa, "Complex dynamics of derivative-free methods for nonlinear equations," *Applied Mathematics and Computation*, vol. 219, no. 12, pp. 7023–7035, 2013.
- [16] S. C. Chapra, *Applied Numerical Methods with MATLAB® for Engineers and Scientists*. 6th Ed. New York, 2010.
- [17] M. R. Farmer, "Computing the zeros of polynomials using the divide and conquer approach," Ph.D thesis. Department of Computer Science and Information Systems, Birkbeck, University of London, 2014.
- [18] G. H. Nedzhibov, "Iterative methods for simultaneous computing arbitrary number of multiple zeros of nonlinear equations," *International Journal of Computer Mathematics*, vol. 90, no. 5, pp. 994–1007, 2013.