AIMS Mathematics, 9(1): 1755–1771.
DOI: 10.3934/math.2024085
Received: 29 September 2023
Revised: 07 November 2023
Accepted: 07 November 2023
Published: 14 December 2023

http://www.aimspress.com/journal/Math

Research article

Highly efficient family of two-step simultaneous method for all polynomial roots

Mudassir Shams[1,2], **Nasreen Kausar**[3], **Serkan Araci**[4], **Liang Kong**[5] **and Bruno Carpentieri**[1,*]

[1] Faculty of Engineering, Free University of Bozen-Bolzano (BZ), 39100, Italy

[2] Department of Mathematics and Statistics, Riphah International University I-14, Islamabad 44000, Pakistan

[3] Department of Mathematics, Faculty of Arts and Science, Yildiz Technical University, Esenler 34220, Istanbul, Türkiye

[4] Department of Basic Sciences, Faculty of Engineering, Hasan Kalyoncu University, TR-27010 Gaziantep, Türkiye

[5] Department of Mathematical Sciences and Philosophy, University of Illinois Springfield, USA

* **Correspondence:** Email: bruno.carpentieri@unibz.it.

Abstract: In this article, we constructed a derivative-free family of iterative techniques for extracting simultaneously all the distinct roots of a non-linear polynomial equation. Convergence analysis is discussed to show that the proposed family of iterative method has fifth order convergence. Nonlinear test models including fractional conversion, predator-prey, chemical reactor and beam designing models are included. Also many other interesting results concerning symmetric problems with application of group symmetry are also described. The simultaneous iterative scheme is applied starting with the initial estimates to get the exact roots within the given tolerance. The proposed iterative scheme requires less function evaluations and computation time as compared to existing classical methods. Dynamical planes are exhibited in CAS-MATLAB (R2011B) to show how the simultaneous iterative approach outperforms single roots finding methods that might confine the divergence zone in terms of global convergence. Furthermore, convergence domains, namely basins of attraction that are symmetrical through fractal-like edges, are analyzed using the graphical tool. Numerical results and residual graphs are presented in detail for the simultaneous iterative method. An extensive study has been made for the newly developed simultaneous iterative scheme, which is found to be efficient, robust and authentic in its domain.

Keywords: polynomial equations; numerical algorithm; iterative methods; fractals; CPU-time
Mathematics Subject Classification: 65H04, 65H05, 65Y05, 65M12

1. Introduction

Many theoretical and applied problems in different fields of study, such as physical sciences, biomedical engineering, chemistry, biological and chemical sciences, computer sciences, finance, statistics, economics and applied mathematical sciences contain symmetry, which may result a nonlinear polynomial equation

$$\vartheta(\varkappa) = 0, \tag{1.1}$$

with m distinct roots $\wp_1, ..., \wp_m$. There are numerous numerical iterative methods for computing the roots of polynomial equations. These methods can be classified into two types: numerical iterative methods that approximate a single root at a time and iterative numerical methods that approximate all of the roots of a nonlinear polynomial equation simultaneously.

The Newton-Raphson method given as

$$\varkappa^{(i+1)} = \varkappa^{(i)} - \frac{\vartheta(\varkappa^{(i)})}{\vartheta'(\varkappa^{(i)})}, \tag{1.2}$$

is one the most commonly used numerical methods to approximate a single root of Eq (1.1) at a time, thanks to its local quadratic convergence. Iterative methods, such as Eq (1.2), have a significant drawback in that they require derivative evaluation at each iteration, which incurs a significant computational cost, and diverge if $\vartheta'(\varkappa^{(i)}) = 0$ or the initial guess is far from the root. See, for example, the work of Akram et al. [1], Cordero et al. [2], Agarwal et al. [3], Chicharro et al. [4], Behl et al. [5], Chun et al. [6], Zafar et al. [7] and many others). Simultaneous methods having global convergence property, are stable and applicable for parallel execution as well. Convergence analysis, computational efficiency and parallel implementation of simultaneous iterative techniques have been studied by McNamee [8], Nedzhibov [9], Proinov et al. [10], Mir et al. [11], Cholakov [12], Proinov and Vasileva [13], Proinov and Petkova [14], Sanchez et al. [15], Ivanov [16], Kanno et al. [17] as well as the references cited therein.

Weierstrass method (WDM) [18], which approximates all the roots of Eq (1.1) simultaneously, given by

$$\varkappa_t^{(i+1)} = \varkappa_t^{(i)} - \frac{\vartheta(\varkappa_t^{(i)})}{\prod\limits_{\substack{r=1 \\ r\neq t}}^{m} (\varkappa_t^{(i)} - \varkappa_r^{(i)})}, \tag{1.3}$$

is one of the most famous iterative techniques used in comparison to Newton Raphson. The term $\frac{\vartheta(\varkappa_t^{(i)})}{\prod\limits_{\substack{r=1 \\ r\neq t}}^{m} (\varkappa_t^{(i)} - \varkappa_r^{(i)})}$ is called Weierstrass correction. The Weierstrass method, which simultaneously approximates all roots of Eq (1.1), has local quadratic convergence. Weierstrass introduced the method in 1891, and it was rediscovered by Kerner [19], Durand [20], Dochev [21], and Presic [22]. Dochev demonstrated the first local quadratic convergence of the Weierstrass iterative technique in 1962.

Later on, in 2015, Proinov [23] presented the general convergence theorem for the iterative process and discussed the use of the Weierstrass root approximating technique. In 2016, Nedzibove [24] developed a modified version of Eq (1.3), and Marcheva et al. [25] presented the local convergence theorem of the modified Weierstrass method in 2020. Song et al. [26] created and verified three new

derivative-free simultaneous iterative Weierstrass computational methods for calculating all polynomial problem roots.

The goal of this paper is to develop and analyze the derivative-free family of Weierstrass-type methods with accelerated convergence in order to find the roots of the above problem, which should have better convergence behavior than the Weierstrass method. We analyze the convergence behavior of the proposed method by computing CPU-time, log of residual errors, computational errors, and drawing dynamical planes while solving numerical test problems. Several graphs are plotted to demonstrate that the newly constructed method outperforms existing methods in the literature in efficiency and accuracy.

2. Derivation and theoretical convergence analysis of the simultaneous root finding method

First, we present the third order family of the single root finding method (MS), and then we generalize this scheme to simultaneously find all the roots of Eq (1.1):

$$\varkappa^{(i+1)} = \varkappa^{(i)} - \left(\frac{(3\alpha-1)\vartheta'(\varkappa^{(i)}) + (1-\alpha)\vartheta'(y^{(i)})}{(2\alpha-1)\vartheta'(\varkappa^{(i)}) + \vartheta'(y^{(i)})}\right)\frac{\vartheta(\varkappa^{(i)})}{\vartheta'(\varkappa^{(i)})}, i = 0, 1, 2, ..., \tag{2.1}$$

where $y^{(i)} = \varkappa^{(i)} - \frac{\vartheta(\varkappa^{(i)})}{\vartheta'(\varkappa^{(i)})}$ and $\alpha \in \Re$.

The convergence theorem that follows validates our assumption.

Theorem 1. *Let $\overline{\mathcal{F}}$ be an open interval and $\wp \in \overline{\mathcal{F}}$ indicate the precise root of a function that is sufficiently differentiable for $\vartheta : \overline{\mathcal{F}} \subseteq \Re \longrightarrow \Re$ in an open interval $\overline{\mathcal{F}}$. The following error equation is satisfied if the convergence order of the family of iterative numerical procedures is three and $\varkappa^{(0)}$ is close to $\wp$:*

$$\varsigma^{(i+1)} = (-\frac{\Im_2^2}{\alpha} + 2\Im_2^2 + \frac{1}{2}\Im_3)(\varsigma^{(i)})^3 + O((\varsigma^{(i)})^4), \tag{2.2}$$

where $\Im_m = \frac{\vartheta^m(\wp)}{m!\vartheta'(\wp)}, m \geq 2$.

Proof. Let $\wp$ be a single exact-root of ϑ and $\varkappa^{(i)} = \wp + \varsigma^{(i)}$, where $e^{(i)}$ is the error at ith iterative step. The following is the result of expanding $\vartheta(\varkappa^{(i)})$ around $\varkappa = \wp$, using Taylor's series with $\vartheta(\wp) = 0$:

$$\vartheta(\varkappa^{(i)}) = \vartheta'(\wp)(\varsigma^{(i)} + \Im_2(\varsigma^{(i)})^2 + \Im_3(\varsigma^{(i)})^3 + ...), \tag{2.3}$$

and

$$\vartheta'(\varkappa^{(i)}) = \vartheta'(\wp)(1 + 2\Im_2\varsigma^{(i)} + 3\Im_3(\varsigma^{(i)})^2 + ...). \tag{2.4}$$

Dividing (2.3) by (2.4), we have:

$$\frac{\vartheta(\varkappa^{(i)})}{\vartheta'(\varkappa^{(i)})} = \varsigma^{(i)} - \Im_2(\varsigma^{(r)})^2 + (2\Im_2 + 2\Im_3)(\varsigma^{(i)})^3 + ..., \tag{2.5}$$

and

$$y^{(i)} = \wp + \Im_2(\varsigma^{(i)})^2 + (-2\Im_2^2 + 2\Im_3)(\varsigma^{(i)})^3 + ...,$$

$$\vartheta'(y^{(i)}) = \vartheta'(\wp)(1 + 2\Im_2^2(\varsigma^{(i)})^2 + 2(-2\Im_2^2 + 2\Im_3)(\varsigma^{(i)})^3 + ...).$$

Using $\vartheta'(y^{(i)})$ and $\vartheta'(u^{(i)})$ in the 2^{nd}-step of (2.1), we have:

$$\left(\frac{(3\alpha-1)\vartheta'(\varkappa^{(i)})+(1-\alpha)\vartheta'(y^{(i)})}{(2\alpha-1)\vartheta'(\varkappa^{(i)})+\vartheta'(y^{(i)})}\right)\frac{\vartheta(\varkappa^{(i)})}{\vartheta'(\varkappa^{(i)})}=e^{(i)}+(\frac{\Im_2^2}{\alpha}-\frac{1}{2}\Im_3-2\Im_2^2)(\varsigma^{(i)})^3+... \tag{2.6}$$

Thus,

$$\varkappa^{(i+1)}=\xi+\left(-\frac{\Im_2^2}{\alpha}+\frac{1}{2}\Im_3+2\Im_2^2\right)(\varsigma^{(i)})^3+O((\varsigma^{(i)})^4),$$

and it becomes

$$\varsigma^{(i+1)}=(-\frac{\Im_2^2}{\alpha}+2\Im_2^2+\frac{1}{2}\Im_3)(\varsigma^{(r)})^3+O((\varsigma^{(i)})^4). \tag{2.7}$$

Hence the theorem is proved. □

Now, we convert iterative method (2.1) to simultaneous method by using Weierstrass correction [8] given by

$$\Delta(\varkappa_t^{(i)})=\frac{\vartheta(\varkappa_t^{(i)})}{\prod\limits_{\substack{r\neq t\\ r=1}}^{m}(\varkappa_t^{(i)}-\varkappa_r^{(i)})},(t,r=1,2,3,...,m)$$

as follows:

$$\varkappa_t^{(i+1)}=\varkappa_t^{(i)}-\frac{\vartheta(\varkappa_t^{(i)})}{\prod\limits_{\substack{r\neq t\\ r=1}}^{m}(\varkappa_t^{(i)}-\varkappa_r^{(i)})}\left(\frac{(3\alpha-1)\prod\limits_{\substack{r\neq t\\ r=1}}^{m}(\varkappa_t^{(i)}-\varkappa_r^{(i)})+(1-\alpha)\prod\limits_{\substack{r\neq t\\ r=1}}^{m}(y_t^{(i)}-y_r^{(i)})}{\prod\limits_{\substack{r\neq t\\ r=1}}^{m}(y_t^{(i)}-y_r^{(i)})+(2\alpha-1)\prod\limits_{\substack{r\neq t\\ r=1}}^{m}(\varkappa_t^{(i)}-\varkappa_r^{(i)})}\right), \tag{2.8}$$

where $y_t^{(i)}=\varkappa_t^{(i)}-\frac{\vartheta(\varkappa_t^{(i)})}{\prod\limits_{\substack{r\neq t\\ r=1}}^{m}(\varkappa_t^{(i)}-\widehat{\varkappa}_r^{(i)})},(t,r=1,2,3,...,n)$ and $\widehat{\varkappa}_r^{(i)}$ is a modified two step Steffensen-type [27] fourth-order convergent method, i.e., $\widehat{\varkappa}_r^{(i)}=y_r^{(i)}-\frac{\vartheta(y_r^{(i)})}{\vartheta\left[\varkappa_r^{(i)},y_r^{(i)}\right]+\vartheta\left[y_r^{(i)},t_r^{(i)}\right]-\vartheta\left[\varkappa_r^{(i)},t_r^{(i)}\right]+\alpha\left(y_r^{(i)}-\varkappa_r^{(i)}\right)\left(y_r^{(i)}-t_r^{(i)}\right)}, y_r^{(i)}=\varkappa_r^{(i)}-\frac{\vartheta(\varkappa_r^{(i)})}{\vartheta\left[\varkappa_r^{(i)},t_r^{(i)}\right]}, t_r^{(i)}=\varkappa_r^{(i)}+\vartheta(\varkappa_r^{(i)}), \alpha\in\Re$. In this case, $\vartheta[.,.]$ is the divided difference of order one and two, respectively.

Remark 1. *The family of derivative-free methods (2.8) determines all distinct roots of (1.1) simultaneously and is abbreviated as MM.*

2.1. Convergence analysis

This section discusses MM convergence analysis.

Theorem 2. *Let $\wp_1,\wp_2,...,\wp_m$ be m simple roots of (1.1). If $\varkappa_1^{(0)},\varkappa_2^{(0)},...,\varkappa_m^{(0)}$ are, respectively, the initial estimates of the roots and close to exact roots, then MM has fifth-order convergence.*

Proof. Let $\epsilon_t=\varkappa_t-\wp_t$, $\epsilon_t'=y_t-\wp_t$ and $\epsilon_t''=z_t-\wp_t$ be the residual errors in $\varkappa_t,y_t$ and z_t estimations, respectively. To simplify things, we leave out iteration index i. From the 1^{st}-step of (2.8), we have:

$$y_t-\wp_t \;=\; u_t-\wp_i-\Delta(\varkappa_t),$$

$$\epsilon_t' = \epsilon_t - \epsilon_i \frac{\Delta(\varkappa_t)}{\epsilon_t} = \epsilon_i(1 - G_t), \tag{2.9}$$

where $\Delta(\varkappa_t) = \frac{\vartheta(\varkappa_t)}{\prod\limits_{\substack{j\neq t\\ j=1}}^{m}(\varkappa_i - \widehat{u}_j)}$ and

$$G_t = \frac{\Delta(\varkappa_t)}{\epsilon_t} = \frac{\vartheta(\varkappa_t)}{\epsilon_t \prod\limits_{\substack{j\neq t\\ j=1}}^{m}(\varkappa_i - \widehat{u}_j)} = \frac{\prod\limits_{j=1}^{m}(\varkappa_t - \xi_j)}{\epsilon_t \prod\limits_{\substack{j\neq t\\ j=1}}^{m}(\varkappa_t - \widehat{u}_j)} = \frac{(\varkappa_t - \wp_t)\prod\limits_{\substack{j\neq t\\ j=1}}^{m}(\varkappa_t - \wp_j)}{\epsilon_t \prod\limits_{\substack{j\neq t\\ j=1}}^{m}(\varkappa_t - \widehat{\varkappa}_j)} = \prod_{\substack{j\neq t\\ j=1}}^{m} \frac{(u_t - \wp_j)}{(\varkappa_t - \widehat{\varkappa}_j)}, \tag{2.10}$$

with

$$\frac{\varkappa_t - \wp_j}{\varkappa_t - \widehat{\varkappa}_j} = \frac{\varkappa_t - \widehat{\varkappa}_j + \widehat{\varkappa}_j - \wp_j}{\varkappa_t - \widehat{\varkappa}_j} = 1 + \frac{\widehat{\varkappa}_j - \wp_j}{\varkappa_t - \widehat{\varkappa}_j} = 1 + \mathrm{O}\,(\epsilon^4),$$

and $\widehat{\varkappa}_j - \wp_j = \mathrm{O}\,(\epsilon^4)$ (see [27]). Now, for a small enough ϵ: if $\wp_t$ is a simple root, then $\left|\varkappa_i - \varkappa_j\right|$ is bounded away from zero, resulting in the following expression:

$$\prod_{\substack{j\neq t\\ j=1}}^{m} \frac{(\varkappa_i - \wp_j)}{(\varkappa_t - \varkappa_j)} = (1 + \mathrm{O}\,(\epsilon_i^4))^{m-1} = 1 + (m-1)\,\mathrm{O}\,(\epsilon_t^4) = 1 + \mathrm{O}\,(\epsilon_t^4).$$

Then it implies that:

$$G_i = 1 + \mathrm{O}\,(\epsilon_i{}^4), G_i - 1 = \mathrm{O}\,(\epsilon_i{}^4).$$

From (2.9), we have

$$\epsilon_t^{'} = \mathrm{O}\,(\epsilon_t{}^5). \tag{2.11}$$

Now by taking the 2^{nd}-step of Eq (2.8), we have:

$$\begin{aligned}
\varkappa_{t+1} - \wp_t &= \varkappa_t - \wp_t - \Delta(\varkappa_t)\left(\frac{(3\alpha-1)\prod\limits_{\substack{j\neq t\\ j=1}}^{m}(\varkappa_t - \varkappa_j) + (1-\alpha)\prod\limits_{\substack{j\neq i\\ j=1}}^{m}(y_t - y_j)}{\prod\limits_{\substack{j\neq t\\ j=1}}^{m}(y_t - y_j) + (2\alpha-1)\prod\limits_{\substack{j\neq t\\ j=1}}^{m}(\varkappa_t - \varkappa_j)}\right),\\
\epsilon_t'' &= \epsilon_t - \epsilon_i \frac{\Delta(\varkappa_t)}{\epsilon_t}\left(\frac{(3\alpha-1) + (1-\alpha)\frac{\prod\limits_{\substack{j\neq t\\ j=1}}^{m}(y_i - y_j)}{\prod\limits_{\substack{j\neq t\\ j=1}}^{m}(\varkappa_t - \varkappa_j)}}{\frac{\prod\limits_{\substack{j\neq t\\ j=1}}^{m}(y_t - y_j)}{\prod\limits_{\substack{j\neq i\\ j=1}}^{m}(\varkappa_t - \varkappa_j)} + (2\alpha-1)}\right),\\
\epsilon_t'' &= \epsilon_t - \epsilon_t \frac{\Delta(\varkappa_t)}{\epsilon_t}\left(\frac{(3\alpha-1) + (1-\alpha)\prod\limits_{\substack{j\neq t\\ j=1}}^{m}\left(\frac{y_t - y_j}{\varkappa_t - \varkappa_j}\right)}{\prod\limits_{\substack{j\neq i\\ j=1}}^{m}\left(\frac{y_t - y_j}{\varkappa_t - \varkappa_j}\right) + (2\alpha-1)}\right),
\end{aligned} \tag{2.12}$$

where

$$\prod_{\substack{j\neq i\\ j=1}}^{m}\left(\frac{y_i-y_j}{\varkappa_t-\varkappa_j}\right) = \prod_{\substack{j\neq t\\ j=1}}^{m}\left(\frac{(y_t-\wp_t+\wp_t-y_j+\wp_j-\wp_j)}{(\varkappa_i-\wp_t+\wp_t-\varkappa_j+\wp_j-\wp_j)}\right) \tag{2.13}$$

$$= \prod_{\substack{j\neq t\\ j=1}}^{m}\left(\frac{(\epsilon_t^{'}-\epsilon_j^{'}+\wp_t-\wp_j)}{(\epsilon_t-\epsilon_j+\wp_i-\wp_j)}\right).$$

Assume $\epsilon_t = \epsilon_j$ are same order, say $O(\epsilon_t)$, then it follows that

$$= \prod_{\substack{j\neq t\\ j=1}}^{m}(1) = 1. \tag{2.14}$$

Using Eq (2.14) in Eq (2.12), we have:

$$\epsilon_t^{''} = \epsilon_t - \epsilon_t\left(\frac{\Delta(x_t)}{\epsilon_t}\right)\left(\frac{(3\alpha-1)+(1-\alpha)}{1+(2\alpha-1)}\right) = O\left(\epsilon_i^5\right). \tag{2.15}$$

Hence, the theorem is proved. □

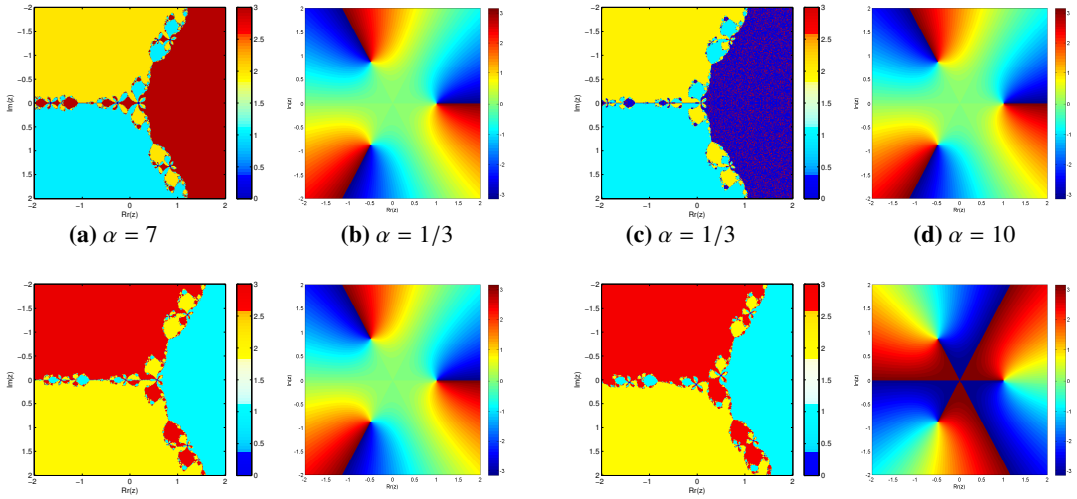
3. Basins of attraction

We employ the real and imaginary portions of the initial approximations across a grid of 250×250 of square $[-2.5\times2.5]^2 \in \mathbb{C}$ in the complex plane to construct the basins of attraction of the iterative procedures used by MS, MM and the Zhang et al. [28] method (ZPH). Due to the global convergence behavior of the simultaneous technique, the maximum number of iterations was set at 15 and the stopping criterion was $\left|\vartheta\left(\varkappa_t^{(i)}\right)\right| < 10^{-5}$. We employ various colors to show the root at which the iterative technique converges, and black in all other cases. Basin colors sharpness signifies fewer iterations. We consider the following two non-linear functions for the generation of basins: $\vartheta_1(\varkappa) = \varkappa^3 - 1$ with exact roots -1, -0.5+0.866i, -0.5-0.866i, and $\vartheta_2(\varkappa) = \varkappa^6 - 1i*\varkappa^3 + 1$ with exact roots -1.0167+0.587i, -1.1740i, 1.0167+0.587i, -0.7377-0.4259i, 0.8518i, 0.7377-0.4259i. Figures 1(a,c,e,g) and 2(a,c,e,g), present the basins of attraction of numerical iterative scheme MS for various parameter values of α and Figures 1(b,d,f,h) and 2(b,d,f,h) show basins of attraction of simultaneous methods MM (for various parameter values of α) and ZPH respectively for non-linear equation $\vartheta_1(\varkappa)$–$\vartheta_2(\varkappa)$. Table 1 and color brightness in Figure 1(b,d,f) shows the ascendancy in efficiency of MM as compared to MS and ZPH.

Table 1 clearly demonstrates that the simultaneous method MM surpasses MS and ZPH in terms of actual time and number of cycles required for convergence to the exact root of the equation $\vartheta_1(\varkappa)$ and $\vartheta_2(\varkappa)$ for any initial guessed value used to generate basins of attraction.

Table 1. Elapsed time and number of iterations for generation of basins of attraction.

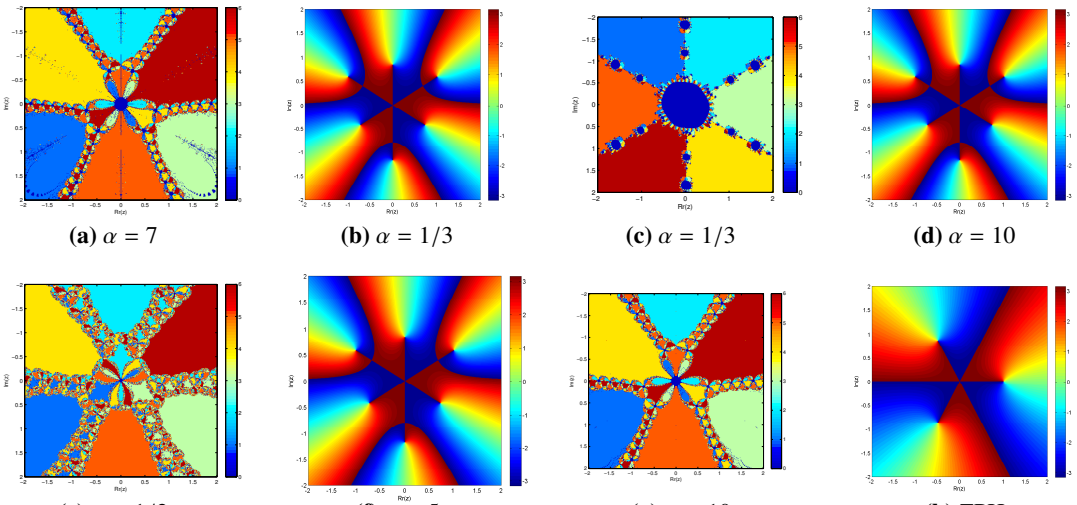
Method	MS		MM		ZPH	
	Elapsed time	Iterations	Elapsed time	Iterations	Elapsed time	Iterations
$\vartheta_1(\varkappa)$	1.583536	25	0.456321	4	1.432156	7
$\vartheta_2(\varkappa)$	2.732514	23	0.632145	5	2.132564	7



(a) $\alpha = 7$ **(b)** $\alpha = 1/3$ **(c)** $\alpha = 1/3$ **(d)** $\alpha = 10$

(e) $\alpha = 1/2$ **(f)** $\alpha = 5$ **(g)** $\alpha = 10$ **(h)** ZPH

Figure 1. (a,c,e,g) depicts the basins of attraction of iterative-scheme MS (for different value of α) and (b,d,f,h) shows the basin of attraction of the MM simultaneous methods (for different value of α), ZPH, respectively, for non-linear equation $\vartheta_1(\varkappa)$.



(a) $\alpha = 7$ **(b)** $\alpha = 1/3$ **(c)** $\alpha = 1/3$ **(d)** $\alpha = 10$

(e) $\alpha = 1/2$ **(f)** $\alpha = 5$ **(g)** $\alpha = 10$ **(h)** ZPH

Figure 2. (a,c,e,g) presents the basins of attraction of iterative-scheme MS (for different value of α) and (b,d,f,h) shows basin of attraction of simultaneous methods MM (for different value of α), ZPH, respectively, for nonlinear equation $\vartheta_2(\varkappa)$.

4. Percentage computational efficiency

Here, we compare the computation % efficiency of Zhang et al.'s [28] convergence order five technique (abbreviated as ZPH), i.e.,

$$\varkappa_t^{(i+1)} = \varkappa_t^{(i)} - \frac{2\Delta_t(\varkappa_t^{(i)})}{1 + \sum\limits_{\substack{r\neq t \\ r=1}}^{n} \frac{\Delta_r(\varkappa_r^{(i)})}{\varkappa_t^{(i)} - \varkappa_r^{(i)}} + \sqrt{\left(1 + \sum\limits_{\substack{r\neq t \\ r=1}}^{n} \frac{\Delta_r(\varkappa_r^{(i)})}{\varkappa_t^{(i)} - \varkappa_r^{(i)}}\right)^2 + 4\Delta_t(\varkappa_t^{(i)}) \sum\limits_{\substack{r\neq t \\ r=1}}^{n} \frac{\Delta_r(\varkappa_t^{(i)})}{\left(\varkappa_t^{(i)} - \varkappa_r^{(i)}\right)\left(\varkappa_t^{(i)} - \Delta_t(\varkappa_t^{(i)}) - \varkappa_r^{(i)}\right)}}}, \tag{4.1}$$

with our newly proposed method based on the efficiency index E defined in [29]

$$E = \frac{\log \tau}{D}, \tag{4.2}$$

where D is the cost of numerical computer calculation

$$\mathrm{D} = \mathrm{D}(m) = \delta_{as} AS_m + \delta_m M_m + \delta_d D_m \tag{4.3}$$

and τ is the convergence order of iterative numerical schemes. Thus, (4.2) becomes:

$$E = \frac{\log \tau}{\delta_{as} AS_m + \delta_m M_m + \delta_d D_m}. \tag{4.4}$$

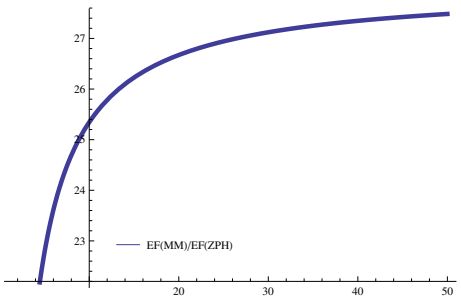
By using (4.4) and the information in Table 2, the percentage efficiency ratio of simultaneous numerical iterative methods can be determined employing the following formula for$\rho((MM), (ZPH))$ given by:

$$\rho(MM), (ZPH)) = \left(\frac{E(MM)}{E(ZPH)} - 1\right) \times 100 \ \text{(in percent)}, \tag{4.5}$$

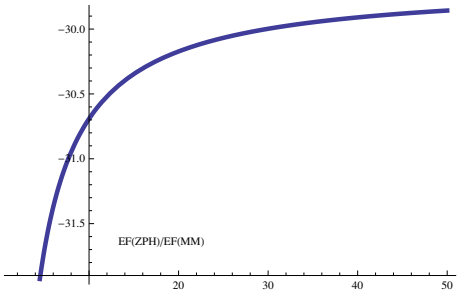
where $O(m) = B$. The dominance efficiency of MM over ZPH is demonstrated clearly in Figure 3(a,b).

Table 2. The number of fundamental mathematical operations.

Methods	AS_m	M_m	D_m
ZPH	$8.0m^2 + B$	$6.0m^2 + B$	$2.0m^2 + B$
MM	$6.0m^2 + B$	$3.0m^2 + B$	$2.0m^2 + B$



(a) Computational efficiency of numerical simultaneous scheme MM w.r.t. ZPH.



(b) Computational efficiency of numerical simultaneous scheme ZPH w.r.t. MM.

Figure 3. (a,b), compares the computational effectiveness of the simultaneous algorithms MM and ZPH.

5. Numerical results

The ZPH and MM family of techniques were utilized to demonstrate the use and effectiveness of several nonlinear problems from biological engineering [12] and applied sciences [29–36] are considered in this section. All calculations utilize CAS-Maple 18 64-digit floating point arithmetic, and the process of iteration is terminated when

$$\overline{\varsigma}_t^{(i)} = \left\| \vartheta\left(\varkappa_t^{(i)}\right) \right\|_2 < \in,$$

where $\overline{\varsigma}_t^{(i)}$ represents the absolute error. For simultaneous calculations of all roots of Eq (1.1), we set

$$\in = 10^{-30}$$

as the fixed stopping criteria. In Tables 3–12, the quantity denoted CPU represents the time required to compute the root using the iterative schemes ZPH and MM, respectively. We use $\alpha = \frac{1}{3}$ and $\beta = \frac{1}{2}$ in all numerical calculations. In Tables 4, 6, 8, 10, 12 for random initial vectors ($\overset{(0)}{\vartheta_1}$-$\overset{(0)}{\vartheta_3}$) taken from Appendix Table 13, MM and ZPH are represented by $\mathrm{MM}_{\vartheta_1}-\mathrm{MM3}_{\vartheta_3}$ and $\mathrm{ZPH}_{\vartheta_1}-\mathrm{ZPH}_{\vartheta_3}$, respectively.

Example 1. *An engineering application.*

A non-linear expression described in [30]

$$\vartheta_3(\varkappa) = \varkappa^4 - 7.79075\varkappa^3 + 14.7445\varkappa^2 + 2.511\varkappa - 1.674, \tag{5.1}$$

depicts the minute modification of nitrogen hydrogen (NH) supply at 250 atm and 227 K. The exact roots of Eq (5.1) are $\wp_1 = 3.9485 + 0.3161i, \wp_2 = 3.9485 - 0.31610i, \wp_3 = -0.38410, \wp_4 = 0.2778$. *The initial guessed values for calculating* $\vartheta_3(\varkappa)$ *are taken as equal to* $\overset{(0)}{\varkappa}_1 = 3.50 + 0.30i$, $\overset{(0)}{\varkappa}_2 = 3.50 - 0.30i$, $\overset{(0)}{\varkappa}_3 = -0.30 + 0.010i$, $\overset{(0)}{\varkappa}_4 = 1.80 + 0.010i$.

On the basis of efficiency achieved after five iterations, Table 3 unequivocally shows that the MM technique outperforms the ZPH method. Table 1 displays the global convergence behavior of the proposed numerical schemes on various randomly generated initial guesses (see Appendix Table 13).

In Table 4, $\mathrm{ZPH}_{\vartheta_1}-\mathrm{ZPH}_{\vartheta_3}$ *and* $\mathrm{MM}_{\vartheta_1}-\mathrm{MM}_{\vartheta_3}$ *the numerical scheme ZPH and MM outputs for random initial vectors generated by the MATLAB computing tool. Numerical results of Table 4 clearly indicate numerical schemes are far better than ZPH for solving (5.1) and more complex engineering problems. The number of iterations and CPU time significantly increase when compared to the numerical results obtained in Table 3. Table 4 shows that for any initial vector choice, the method MM converges to exact roots with greater efficiency than ZPH.*

Table 3. Residual errors comparison.

Method	CPU	$\overline{\varsigma}_1^{(5)}$	$\overline{\varsigma}_2^{(5)}$	$\overline{\varsigma}_3^{(5)}$	$\overline{\varsigma}_4^{(5)}$
ZPH	0.241	1.29e-17	1.62e-17	6.15e-32	4.16e-15
MM	0.016	3.41e-51	1.15e-51	5.92e-53	5.82e-53

Table 4. Residual errors comparison on random initial vector.

Method	CPU	$\overline{\varsigma}_1^{(17)}$	$\overline{\varsigma}_2^{(17)}$	$\overline{\varsigma}_3^{(17)}$	$\overline{\varsigma}_4^{(17)}$
ZPH_{ϑ_1}	2.1456	3.34e-07	2.43e-05	4.30e-07	8.35e-04
ZPH_{ϑ_2}	3.1255	1.21e-03	0.32e-03	0.30e-08	1.13e-05
ZPH_{ϑ_3}	2.9914	4.34e-05	2.31e-04	0.13e-03	4.35e-04
MM_{ϑ_1}	1.1201	9.41e-34	4.3e-29	4.13e-32	7.81e-35
MM_{ϑ_2}	1.32451	3.10e-32	2.3e-26	1.13e-18	8.03e-21
MM_{ϑ_3}	0.93114	1.01e-33	2.3e-21	1.31e-19	8.73e-30

Example 2. *Predator-prey model (PPM).*

Consider PPM in which predication rate [7] is denoted by

$$\overline{P}(\varkappa) = \frac{\overline{k}\varkappa^3}{\overline{a}^3 + \varkappa^3}, \overline{a}, \overline{k} > 0, \tag{5.2}$$

where $\varkappa$ is the amount of aphids as victims [31] and female microbes as a predator. Malthusian Model holds $G(\varkappa) = r\varkappa$, as the growth rate of aphids. For solution of the PPM we take $\overline{P}(\varkappa) = \overline{G}(\varkappa)$. This implies

$$r\varkappa^3 - \overline{k}\varkappa^2 + i\overline{a}^3 = 0. \tag{5.3}$$

Taking $\overline{k} = 30$ (the way pests are devoured), $\overline{a} = 20$ (amount of bugs) and $i = 2^{\frac{-1}{3}}$ (hourly amount) in (5.3), we get:

$$\vartheta_4(\varkappa) = 0.7937005260\varkappa^3 - 30\varkappa^2 + 6349.604208. \tag{5.4}$$

The exact roots of Eq (5.4) are $\wp_1 = 25.198$, $\wp_2 = 25.198$, $\wp_3 = 12.84$. The initial guessed estimates for $\vartheta_4(\varkappa)$ are selected as:

$$\overset{(0)}{\varkappa}_1 = 1.8 + 8.7i, \ \overset{(0)}{\varkappa}_2 = 1.8 - 8.7i, \ \overset{(0)}{\varkappa}_3 = 0.1 + 0.1i.$$

After seventeen iterations, Table 5 clearly shows that the MM methodology outperforms the ZPH method in terms of efficiency. Table 6 displays the global convergence behavior of the proposed numerical schemes on various randomly generated initial guesses (see Appendix Table 13).

Table 6 shows the numerical schemes $ZPH_{\vartheta_1}-ZPH_{\vartheta_3}$ and $MM_{\vartheta_1}-MM_{\vartheta_3}$ outputs for random initial vectors generated by the MATLAB computing tool. The numerical results of the table clearly show that the proposed numerical schemes are superior to ZPH for solving (5.4) and more difficult engineering problems. The number of iterations and CPU time significantly increase when compared to the numerical results obtained in Table 5. Table 6 shows that for any initial vector choice, the method MM converges to exact roots with greater efficiency than ZPH on iteration 37.

Table 5. Residual computational errors comparison.

Method	CPU	$\overline{\varsigma}_1^{(17)}$	$\overline{\varsigma}_2^{(17)}$	$\overline{\varsigma}_3^{(17)}$
ZPH	0.1325	0.0200	0.0200	0.0030
MM	0.0154	0.14e-09	9.15e-09	4.41e-09

Table 6. Residual errors comparison on random initial vector.

Method	CPU	$\overline{\varsigma}_1^{(37)}$	$\overline{\varsigma}_2^{(37)}$	$\overline{\varsigma}_3^{(37)}$
ZPH_{ϑ_1}	5.4371	1.074e-20	3.398e-10	7.213e-18
ZPH_{ϑ_2}	5.1351	2.087e-29	6.385e-11	0.321e-19
ZPH_{ϑ_3}	3.1351	5.450e-27	7.378e-13	0.345e-17
MM_{ϑ_1}	2.1221	0.494e-37	8.375e-30	3.345e-35
MM_{ϑ_2}	1.8321	4.074e-35	0.378e-26	6.345e-35
MM_{ϑ_3}	2.9321	6.0548e-31	1.378e-26	1378e-28

Example 3. *Beam positioning problem.*

In this example, we look at a beam positioning problem [32], which yields a nonlinear function

$$\vartheta_5(\varkappa) = \varkappa^4 + 4\varkappa^3 - 24\varkappa^2 + 16\varkappa + 16. \tag{5.5}$$

The exact roots of Eq (5.5) are

$$\wp_1 = 2,\ \wp_2 = -4 - 2\sqrt{3},\ \wp_3 = 2,\ \wp_4 = -4 + 2\sqrt{3}.$$

The initial estimates for $\vartheta_3(\varkappa)$ are

$$\overset{(0)}{\varkappa}_{1,2} = 1.17,\ \overset{(0)}{\varkappa}_3 = -7.4641,\ \overset{(0)}{\varkappa}_4 = -0.5359.$$

On the basis of efficiency achieved after six iterations, Table 7 unequivocally shows that the MM technique outperforms the ZPH method. Table 8 displays the global convergence behavior of the proposed numerical schemes on various randomly generated initial guesses (see Appendix Table 13).

Table 8 shows the numerical schemes ZPH_{ϑ_1}–ZPH_{ϑ_3} and MM_{ϑ_1}–MM_{ϑ_3} outputs for random initial vectors generated by the MATLAB computing tool. The numerical results of the Table 8 clearly show that the proposed numerical schemes are superior to ZPH for solving (5.5) and more difficult engineering problems. The number of iterations and CPU time significantly increase when compared to the numerical results obtained in Table 7. Table 8 shows that for any initial vector choice, the method MM converges to exact roots with greater efficiency than ZPH on iteration 11.

Table 7. Residual computational errors comparison.

Method	CPU	$\overline{\varsigma}_1^{(6)}$	$\overline{\varsigma}_2^{(6)}$	$\overline{\varsigma}_3^{(6)}$	$\overline{\varsigma}_4^{(6)}$
ZPH	0.331	2.3e-23	2.3e-24	7.4e-33	3.1e-29
MM	0.111	2.6e-30	2.6e-35	2.1e-35	2.3e-36

Table 8. Residual errors comparison on random initial vector.

Method	CPU	$\overline{\varsigma}_1^{(11)}$	$\overline{\varsigma}_2^{(11)}$	$\overline{\varsigma}_3^{(11)}$	$\overline{\varsigma}_4^{(11)}$
ZPH_{ϑ_1}	2.1231	0.07e-30	2.23e-16	0.31e-18	8.73e-31
ZPH_{ϑ_2}	2.4131	2.02e-30	1.32e-16	0.53e-18	1.53e-31
ZPH_{ϑ_3}	3.1131	1.02e-30	5.35e-16	1.39e-18	8.03e-31
MM_{ϑ_1}	1.0212	9.24e-41	4.53e-30	4.32e-32	7.18e-35
MM_{ϑ_2}	1.8312	8.02e-47	2.30e-36	2.23e-28	8.23e-41
MM_{ϑ_3}	1.7315	7.07e-48	2.37e-36	2.13e-28	1.32e-35

Example 4. *[33] Nuclear boiler.*

Consider a nuclear boiler, with the substances ĂĂ and ŘŘ, breastfed to the apparatus at rates of $\widehat{Q}$ and q-$\widehat{Q}$, respectively. The following are the results of the multifaceted reaction system produced in the container:

$$\begin{aligned} \check{A}\check{A} + \check{R}\check{R} &\longrightarrow \beta\beta, \\ \beta\beta + \check{R}\check{R} &\longrightarrow \check{C}\check{C}, \\ \check{C}\check{C} + \check{R}\check{R} &\longrightarrow \check{D}\check{D}, \\ \check{C}\check{C} + \check{R}\check{R} &\longrightarrow \check{E}\check{E}. \end{aligned}$$

Douglas et al. [34] tested this issue in order to create straightforward feedback control systems as:

$$\hat{H}_{\Im} \frac{2.98\,(\varkappa + 2.25)}{(\varkappa + 1.45)\,(\varkappa + 2.85)^2\,(\varkappa + 4.35)} = -1, \tag{5.6}$$

or

$$\hat{H}_{\Im} * 2.98\,(\varkappa + 2.25) = -(\varkappa + 1.45)\,(\varkappa + 2.85)^2\,(\varkappa + 4.35)1, \tag{5.7}$$

where $\hat{H}_{\Im}$ denotes the proportional controller's gain. This control system is stable for all values of $\hat{H}_{\Im}$ that yield transfer function roots with negative real parts. Taking $\hat{H}_{\Im} = 0$ in (5.7), we have:

$$\vartheta_6(\varkappa) = \varkappa^4 + 11.50\varkappa^3 + 47.49\varkappa^2 + 83.06325\varkappa + 51.23266875 = 0, \tag{5.8}$$

$\vartheta_6(\varkappa)$ has the four negative real roots listed below:

$$\wp_1 = -1.45, \wp_2 = -2.85, \wp_3 = -2.85, \wp_4 = -4.45.$$

We take the root as $\wp = -4.45$ and

$$\overset{(0)}{\varkappa}_1 = -1.0,\ \overset{(0)}{\varkappa}_2 = -2.1,\ \overset{(0)}{\varkappa}_3 = -1.8, \overset{(0)}{\varkappa}_3 = -3.9,$$

are chosen as initial estimates for finding all roots of (5.8) simultaneously.

On the basis of efficiency achieved after seventh iterations, Table 9 unequivocally shows that the MM technique outperforms the ZPH method.

Table 10 displays the global convergence behavior of the proposed numerical schemes on various randomly generated initial guesses (see appendix Table 13).

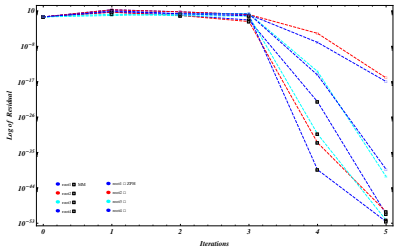
Table 10 shows the numerical schemes ZPH_{ϑ_1}–ZPH_{ϑ_3} and MM_{ϑ_1}–MM_{ϑ_3} outputs for random initial vectors generated by the MATLAB computing tool. The numerical results of Table 10 clearly show that the proposed numerical schemes are superior to ZPH for solving (5.8) and more difficult engineering problems. The number of iterations and CPU time significantly increase when compared to the numerical results obtained in Table 9. Table 10 shows that for any initial vector choice, the method MM converges to exact roots with greater efficiency than ZPH on iteration 10.

Table 9. Residual computational errors comparison of ZPH & MM.

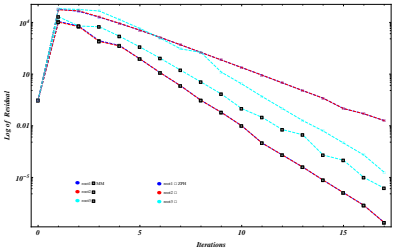
Method	CPU	$\overline{\varsigma}_1^{(7)}$	$\overline{\varsigma}_2^{(7)}$	$\overline{\varsigma}_3^{(7)}$	$\overline{\varsigma}_4^{(7)}$
ZPH	0.13121	3.02e-30	2.32e-16	2.32e-18	0.32e-31
MM	0.02154	9.42e-36	4.13e-30	4.32e-32	7.81e-35

Table 10. Residual errors comparison on random initial vector.

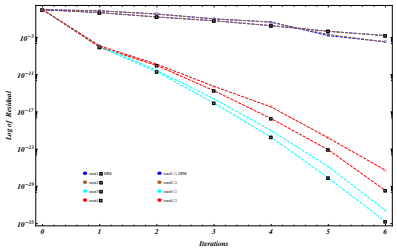
Method	CPU	$\varsigma_1^{(10)}$	$\varsigma_2^{(10)}$	$\varsigma_3^{(10)}$	$\varsigma_4^{(10)}$
ZPH_{ϑ_1}	2.031	3.50e-38	2.3e-18	4.38e-18	4.34e-33
ZPH_{ϑ_2}	3.534	0.01e-39	2.3e-26	5.87e-18	3.53e-36
ZPH_{ϑ_3}	4.1381	4.05e-35	4.31e-26	2.53e-17	2.34e-35
MM_{ϑ_1}	3.0914	9.46e-46	0.30e-30	4.3e-32	7.85e-37
MM_{ϑ_2}	1.0315	4.04e-45	2.35e-26	3.38e-28	8.13e-35
MM_{ϑ_3}	2.1931	1.40e-41	4.35e-26	9.32e-29	1.32e-35



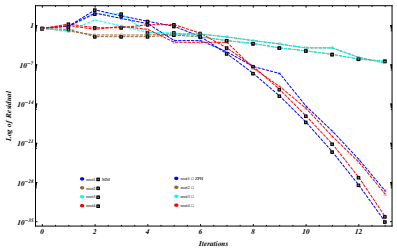
(a) Residual error graph of simultaneous MM and ZPH techniques for locating all roots of (5.1).



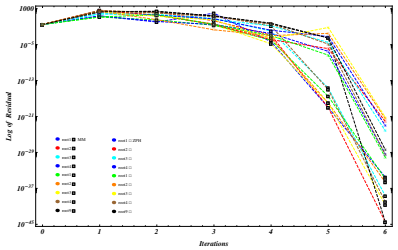
(b) Residual error graph of simultaneous MM and ZPH techniques for locating all roots of (5.4).



(c) Residual error graph of simultaneous MM and ZPH techniques for locating all roots of (5.5).



(d) Residual error graph of simultaneous MM and ZPH techniques for locating all roots of (5.8).



(e) Residual error graph of simultaneous MM and ZPH techniques for locating all roots of (5.9).

Figure 4. (a–e) show residual graphs of simultaneous methods MM and ZPH for nonlinear polynomials $\vartheta_3(\varkappa) - \vartheta_7(\varkappa)$, respectively.

Example 5. *[27, 35, 36] Polynomial of higher degree.*
Consider

$$\vartheta_7(\varkappa) = (\varkappa + 1)(\varkappa + 3)(\varkappa^2 - 2\varkappa + 2)(\varkappa - 1)(\varkappa^2 - 4\varkappa + 5)(\varkappa^2 + 4\varkappa + 5), \tag{5.9}$$

with exact roots:

$$\wp_1 = -1.0,\ \wp_2 = -3.0,\ \wp_{3,4} = 1.0 \pm i,\ \wp_5 = 1.0,\ \wp_{6,7} = -2.0 \pm i,\ \wp_{8,9} = 2.0 \pm 1.0i.$$

The initial approximations of (5.9) have been taken as:

$$\overset{(0)}{\varkappa_1} = -1.30 + 0.20i,\ \overset{(0)}{\varkappa_2} = -2.80 - 0.20i,\ \overset{(0)}{\varkappa}_3 = 1.2 + 1.3i,$$

$$\overset{(0)}{\varkappa}_4 = 0.8 - 1.20i,$$
$$\overset{(0)}{\varkappa}_5 = 0.80 - 0.30i,\ \overset{(0)}{\varkappa}_{6,7} = -1.8 \pm 1.2i,\ \overset{(0)}{\varkappa}_{8,9} = 1.80 \pm 0.8i.$$

On the basis of efficiency achieved after sixth iterations, Table 11 unequivocally shows that the MM technique outperforms the ZPH method. Table 12 displays the global convergence behavior of the proposed numerical schemes on various randomly generated initial guesses (see Appendix Table 13).

Table 12 shows the numerical schemes $ZPH_{\vartheta_1}-ZPH_{\vartheta_3}$ and $MM_{\vartheta_1}-MM_{\vartheta_3}$ outputs for random initial vectors generated by the MATLAB computing tool. The numerical results of the Table 12 clearly show that the proposed numerical schemes are superior to ZPH for solving (5.9) and more difficult engineering problems. The number of iterations and CPU time significantly increase when compared to the numerical results obtained in Table 11. Table 12 shows that for any initial vector choice, the method MM converges to exact roots with greater efficiency than ZPH on iteration 35.

Table 11. Residual computational error of simultaneous methods for the (5.9) root.

Method	CPU	$\overline{\varsigma}_1^{(6)}$	$\overline{\varsigma}_2^{(6)}$	$\overline{\varsigma}_3^{(6)}$	$\overline{\varsigma}_4^{(6)}$	$\overline{\varsigma}_5^{(6)}$	$\overline{\varsigma}_6^{(6)}$	$\overline{\varsigma}_7^{(6)}$	$\overline{\varsigma}_8^{(6)}$	$\overline{\varsigma}_9^{(6)}$
ZPH	0.431	3.0e-30	2.3e-25	0.0	0.0	0.0	2.3e-29	2.3e-35	0.0	0.0
MM	0.221	9.4e-36	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 12. Residual errors comparison on random initial vector.

Method	CPU	$\overline{\varsigma}_1^{(10)}$	$\overline{\varsigma}_2^{(10)}$	$\overline{\varsigma}_3^{(10)}$	$\overline{\varsigma}_4^{(10)}$	$\overline{\varsigma}_5^{(10)}$	$\overline{\varsigma}_6^{(10)}$	$\overline{\varsigma}_7^{(10)}$	$\overline{\varsigma}_8^{(10)}$	$\overline{\varsigma}_9^{(10)}$
ZPH_{ϑ_1}	2.031	4.34e-23	3.50e-28	2.36e-18	4.38e-18	0.34e-23	0.51e-38	0.53e-28	1.31e-18	4.34e-33
ZPH_{ϑ_2}	3.534		0.11e-29	2.31e-26	2.31e-26	5.87e-15	3.13e-26	0.01e-39	0.31e-36	1.87e-18
ZPH_{ϑ_3}	4.1381	2.34e-35	4.15e-25	4.61e-26	2.53e-17	4.05e-35	1.34e-35	4.05e-35	0.311e-36	1.34e-25
MM_{ϑ_1}	3.0914	0.0	0.30e-30	1.30e-30	0.0	9.46e-46	0.0	0.0	5.3e-32	0.85e-47
MM_{ϑ_2}	1.0315	0.0	2.35e-46	3.38e-48	8.13e-45	4.04e-45	0.0	0.0	0.0	8.13e-35
MM_{ϑ_3}	2.1931	1.40e-47	4.35e-46	0.0	0.0	1.40e-41	4.35e-26	9.32e-29	1.32e-35	1.32e-35

6. Conclusions and observation

- In comparison to the well-known Weierstrass-type method ZPH, our newly developed methods is more efficient and stable.
- The results of the numerical test problems from Tables 3−12, computational efficiency from Figure 3, residual error graphs from Figure 4(a-e), and dynamical planes from Figures 1(a-h) and 2(a-h) show that the proposed family of iterative techniques MM outperforms ZPH in terms of efficiency and convergence behavior.
- Furthermore, the elapsed time of iterative technique MM from Table 1 is faster than ZPH.
- This paper proposes a family of Weierstrass-type derivative-free simultaneous techniques of order five. Tables 1–12 and Figures 1–4 show that our MM techniques outperform the Weierstrass-type method ZPH in terms of efficiency, CPU time, basins of attraction, and residual errors. Color brightness inspection in the basins reveals that MM requires fewer iteration steps than the simultaneous method ZPH. The results of the numerical test examples in terms of CPU-time and

residual errors demonstrate the effectiveness and rapid convergence of our proposed derivative-free family of iterative algorithms, MM, in comparison to ZPH.

In summary, the comprehensive empirical results establish that these novel MM techniques outperform ZPH in terms of efficiency, CPU time, basins of attraction, and residual errors. Examination of the color brightness in the basins reveals MM requires fewer iteration steps than ZPH. Overall, the test examples clearly demonstrate the effectiveness, rapid convergence, and computational advantages of the proposed MM algorithms over ZPH.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The work is supported by Provincia autonoma di Bolzano/Alto Adigeâ euro Ripartizione Innovazione, Ricerca, Universitá e Musei (contract nr. 19/34). Bruno Carpentieri is a member of the *Gruppo Nazionale per it Calcolo Scientifico* (GNCS) of the Istituto Nazionale di Alta Matematia (INdAM) and this work was partially supported by INdAM-GNCS under Progetti di Ricerca 2022.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. S. Akram, M. Junjua, N. Yasmin, F. Zafar, A family of weighted mean based optimal fourth order methods for solving system of nonlinear equations, *Life Sci. J.*, **11** (2014), 21–30.
2. A. Cordero, J. García-Maimó, J. R. Torregrosa, M. P. Vassileva, P. Vindel, Chaos in King's iterative family, *Appl. Math. Lett.*, **26** (2013), 842–848. https://doi.org/10.1016/j.aml.2013.03.012
3. P. Agarwal, D. Filali, M. Akram, M. Dilshad, Convergence analysis of a three-step iterative algorithm for generalized set-valued mixed-ordered variational inclusion problem, *Symmetry*, **13** (2021), 444. https://doi.org/10.3390/sym13030444
4. F. I. Chicharro, A. Cordero, N. Garrido, J. R. Torregrosa, Stability and applicability of iterative methods with memory, *J. Math. Chem.*, **57** (2019), 1282–1300. https://doi.org/10.1007/s10910-018-0952-z
5. R. Behl, I. K. Argyros, S. S. Motsa, A new highly efficient and optimal family of eighth order methods for solving nonlinear equations, *Appl. Math. Comput.*, **282** (2016), 175–186. https://doi.org/10.1016/j.amc.2016.02.010
6. C. Chun, Y. I. Kim, Several new third-order iterative methods for solving nonlinear equations, *Acta Appl. Math.*, **109** (2010), 1053–1063. https://doi.org/10.1007/s10440-008-9359-3
7. F. Zafar, A. Cordero, J. R. Torregrosa, An efficient family of optimal eighth-order multiple root finders, *Mathematics*, **6** (2018), 310. https://doi.org/10.3390/math6120310

8. J. M. McNamee, *Numerical Methods for Roots of Polynomials, Part II*, Amsterdam: Elsevier, 2013.
9. G. H. Nedzhibov, Iterative methods for simultaneous computing arbitrary number of multiple zeros of nonlinear equations, *J. Comput. Math.*, **90** (2013), 994–1007. https://doi.org/10.1080/00207160.2012.744000
10. P. D. Proinov, M. T. Vasileva, On the convergence of family of Weierstrass-type root-finding methods, *C. R. Acad. Bulg. Sci.*, **68** (2015), 697–704.
11. N. A. Mir, R. Muneer, I. Jabeen, Some families of two-step simultaneous methods for determining zeros of non-linear equations, *ISRN Appl. Math.*, **2011** (2011), 817174. https://doi.org/10.5402/2011/817174
12. S. I. Cholakov, Local convergence of Chebyshev-like method for simultaneous finding polynomial zeros, *C. R. Acad. Bulg. Sci.*, **66** (2013), 1081–1090.
13. P. D. Proinov, M. T. Vasileva, On a family of Weierstrass-type root-finding methods with accelerated convergence, *Appl. Math. Comput.*, **273** (2016), 957–968. https://doi.org/10.1016/j.amc.2015.10.048
14. P. D. Proinov, M. D. Petkova, A new semilocal convergence theorem for the Weierstrass method for finding zeros of a polynomial simultaneously, *J. Complexity*, **30** (2014), 366–380. https://doi.org/10.1016/j.jco.2013.11.002
15. M. G-S´anchez, M. Noguera, A. Grau, J. R. Herrero, On new computational local orders of convergence, *Appl. Math. Lett.*, **25** (2012), 2023–2030. https://doi.org/10.1016/j.aml.2012.04.012
16. S. I. Ivanov, A unified semilocal convergence analysis of a family of iterative algorithms for computing all zeros of a polynomial simultaneously, *Numer. Algor.*, **75** (2017), 1193–1204. https://doi.org/10.1007/s11075-016-0237-1
17. S. Kanno, N. V. Kjurkchiev, T. Yamamoto, On some methods for the simultaneous determination of polynomial zeros, *Japan J. Indust. Appl. Math.*, **13** (1995), 267–288. https://doi.org/10.1007/BF03167248
18. K. Weierstrass, Neuer beweis des Satzes, dass jede ganze rationale function einer veränderlichen dargestellt werden kann als ein product aus linearen functionen derselben veränderlichen, *Ges. Werke*, 1981, 1085–1101.
19. I. O. Kerner, Ein gesamtschrittverfahren zur berechnung der nullstellen von polynomen, *Numer. Math.*, **8** (1966), 290–294.
20. E. Durand, Solutions numeriques des equations algebriques, tome 1: Equations du Type $F(\varkappa) = 0$, *Racines dun Polynome, Masson*, **17** (1960), 540–546.
21. K. Dochev, Modified Newton methodfor the simultaneous computation of all roots of a given algebraic equation, in Bulgarian, *Phys. Math. J. Bulg. Acad. Sci.*, **5** (1962), 136–139.
22. S. B. Presic, Un proced e iteratif pour la factorisation des polynomes, *C. R. Acad. Sci. Paris. Ser. A.*, **262** (1966), 862–863.
23. P. D. Proinov, General convergence theorems for iterative processes and applications to the Weierstrass root-finding method, *J. Complexity*, **33** (2015), 118–144. https://doi.org/10.1016/j.jco.2015.10.001
24. G. H. Nedzhibov, Improved local convergence analysis of the Inverse Weierstrass method for simultaneous approximation of polynomial zeros, In: *Proceedings of the MATTEX 2018 Conference, Targovishte, Bulgaria*, **1** (2018), 66–73.

25. P. I. Marcheva, S. I. Ivanov, Convergence analysis of a modified Weierstrass method for the simultaneous determination of polynomial zeros, *Symmetry*, **12** (2018), 1408. https://doi.org/10.3390/sym12091408
26. J. S. Song, Simultaneous methods for finding all zeros of a polynomial, *J. Math. Sci. Adv. Appl.*, **33** (2015), 5–18.
27. H. Ren, Q. Wu, W. Bi, A class of two-step Steffensen type methods with fourth-order convergence, *Appl. Math. Comput.*, **209** (2009), 206–210. https://doi.org/10.1016/j.amc.2008.12.039
28. X. Zhang, H. Peng, G. Hu, A high order iteration formula for the simultaneous inclusion of polynomial zeros, *Appl. Math. Comput.*, **179** (2006), 545–552. https://doi.org/10.1016/j.amc.2005.11.117
29. M. Shams, N, Rafiq, N. Kausar, P. Agarwal, C. Park, N. A. Mir, On highly efficient derivative-free family of numerical methods for solving polynomial equation simultaneously, *Adv. Differ. Equ.*, **2021** (2021), 465. https://doi.org/10.1186/s13662-021-03616-1
30. I. K. Argyros, Á. A. Magreñán, L. Orcos, Local convergence and a chemical application of derivative free root finding methods with one parameter based on interpolation, *J. Math. Chem.*, **54** (2016), 1404–1416. https://doi.org/10.1007/s10910-016-0605-z
31. L. Edelstein-Keshet, *Differential Calculus for the Life Sciences*, Vancouver: Univeristy of British Columbia, 2017.
32. J. L. Zachary, *Introduction to Scientific Programming: Computational Problem Solving Using Maple and C*, New York: Springer, 2012.
33. A. Constantinides, M. Mostoufi, *Numerical Methods for Chemical Engineers with MATLAB Applications*, New Jersey: Prentice Hall PTR, 1999.
34. J. M. Douglas, *Process Dynamics and Control*, Englewood Cliffs: Prentice Hall, 1972.
35. 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.
36. M. S. Petkovic, L. D. Petkovic, Construction of zero-finding methods by Weierstrass functions, *Appl. Math. Comput.*, **184** (2007), 351–359. https://doi.org/10.1016/j.amc.2006.05.159

Appendix

Table 13. Random initial vectors generated using built-in commands in MATLAB.

Inital Vector	$\varkappa_1^{(0)}$	$\varkappa_2^{(0)}$	$\varkappa_3^{(0)}$	$\varkappa_4^{(0)}$	$\varkappa_5^{(0)}$	$\varkappa_6^{(0)}$	$\varkappa_7^{(0)}$	$\varkappa_8^{(0)}$	$\varkappa_9^{(0)}$
$\vartheta_1^{(0)}$	0.1234	0.2541	0.0415	0.04125	0.25410	025412	0.5412	0.0354	0..35214
$\vartheta_2^{(0)}$	0.2134	0.5412	0.03251	0.05412	0.10351	0.0541	0.09854	0.0214	0.874515
$\vartheta_3^{(0)}$	0.3251	0.6541	0.210	0.14521	0.2321	0.0012	0.5214	0.2145	0.022145