Research Article

Inverse Family of Numerical Methods for Approximating All Simple and Roots with Multiplicity of Nonlinear Polynomial Equations with Engineering Applications

Mudassir Shams [iD],[1] **Naila Rafiq** [iD],[2] **Nasreen Kausar** [iD],[3] **Shams Forruque Ahmed** [iD],[4] **Nazir Ahmad Mir** [iD],[2] **and Suvash Chandra Saha** [iD][5]

[1]*Department of Mathematics and Statistics, Riphah International University I-14, Islamabad 44 000, Pakistan*
[2]*Department of Mathematics, National University of Modern Languages, Islamabad, Pakistan*
[3]*Deperament of Mathematics, Yildiz Technical University, Faculty of Arts and Science, Esenler 34 210, Istanbul, Turkey*
[4]*Science and Math Program, Asian University for Women, Chattogram 4000, Bangladesh*
[5]*School of Mechanical and Mechatronic Engineering University of Technology Sydney (UTS), Sydney, Australia*

Correspondence should be addressed to Suvash Chandra Saha; suvash.saha@gmail.com

A new inverse family of the iterative method is interrogated in the present article for simultaneously estimating all distinct and multiple roots of nonlinear polynomial equations. Convergence analysis proves that the order of convergence of the newly constructed family of methods is two. The computer algebra systems CAS-Mathematica is used to determine the lower bound of convergence order, which justifies the local convergence of the newly developed method. Some nonlinear models from physics, chemistry, and engineering sciences are considered to demonstrate the performance and efficiency of the newly constructed family of inverse simultaneous methods in comparison to classical methods in the literature. The computational time in seconds and residual error graph of the inverse simultaneous methods are also presented to elaborate their convergence behavior.

1. Introduction

Considering nonlinear polynomial equation of degree n,

$$f(r) = r^n + a_{n-1}r^{n-1} + \cdots + a_0 = \prod_{j=1}^{n}(r - \zeta_j)$$
$$= (r - \zeta_i) * \prod_{\substack{j=1 \\ j \neq i}}^{n}(r - \zeta_j), \quad (1)$$

with arbitrary real or complex coefficient $a_{n-1}, \ldots, a_0$. Let $\zeta_1 \ldots \zeta_n$ denote all the simple or complex roots of (1) with multiplicity $\sigma_1 \ldots \sigma_n$. Newton's method [11] is one of the most basic and ancient methods that is used to estimate single roots of (1) at a time as below:

$$s^{(t)} = r^{(t)} - \frac{f(r^{(t)})}{f'(r^{(t)})}, \quad (t = 0, 1, \ldots). \quad (2)$$

Iterative method (2) has local quadratic convergence. Nedzhibov et al. [13] presented corresponding inverse numerical technique of the same convergence order as

$$s^{(t)} = \frac{(r^{(t)})^2 f'(r^{(t)})}{r^{(t)} f'(r^{(t)}) + f(r^{(t)})}. \quad (3)$$

Here, we propose the following family of the optimal second-order convergence method for finding simple roots of (1) as

$$s^{(t)} = r^{(t)} - \frac{f(r^{(t)})}{f'(r^{(t)})} \frac{1}{\left(1 - \alpha f(r^{(t)})/1 + f(r^{(t)})\right)}, \quad (4)$$

where $\alpha \in \mathbb{R}$. Method (4) is optimal and the convergence order of (4) is 2 if ζ is a simple root of (1) and $\epsilon = r - \zeta$. The error equation of (4) is obtained using Maple-18:

$$\frac{s^{(t)} - \zeta}{\left(r^{(t)} - \zeta\right)^2} \longrightarrow \left(-\alpha + C_2\right); C_k(r) = \frac{f^{(k)}(x)}{k! f'(x)}, \qquad (5)$$

or

$$s^{(t)} - \zeta = O\left(\epsilon^2\right). \qquad (6)$$

Corresponding inverse methods of (4) is constructed as

$$g(r) = \frac{\left(r^{(t)}\right)^2 * \left(f'\left(r^{(t)}\right)\right)}{r^{(t)} f'\left(r^{(t)}\right) + f\left(r^{(t)}\right)\left(1/\left(1 - \alpha f\left(r^{(t)}\right)/1 + f\left(r^{(t)}\right)\right)\right)}, g(\zeta) = \zeta \text{ (fixed point)}, \qquad (8)$$

$$g'(\zeta) = 0. \qquad (9)$$

Inverse iterative schemes (7) are second-order convergence as it is easy to prove $g''(\zeta) \neq 0$.

Besides simple root finding methods [3–5, 13, 15, 16, 18–20] in literature, there exists another class of numerical methods which estimate all real and complex roots of (1) at a time, known as simultaneous methods. Simultaneous numerical iterative schemes are very prevalent due to their global convergence properties and its parallel execution on computers [1, 6, 8–10, 12, 14].

The most prominent method among simultaneous derivative-free iterative technique is the Weierstrass–Dochive [17] method (abbreviated as MWM1), which is defined as

$$s_i^{(t)} = r_i^{(t)} - w\left(r_i^{(t)}\right), \qquad (10)$$

where

$$\frac{f\left(r_i^{(t)}\right)}{f'\left(r_i^{(t)}\right)} = w\left(r_i^{(t)}\right) = \frac{f\left(r_i^{(t)}\right)}{\prod\limits_{\substack{j=1 \\ j \neq i}}^{n}\left(r_i^{(t)} - r_j^{(t)}\right)}, \qquad (i, j = 1, 2, 3, \ldots, n), \qquad (11)$$

is Weierstrass' correction. Method (10) has local quadratic convergence. For finding all multiple roots of (1), we use the following correction [17]:

$$w_*\left(r_i^{(t)}\right) = \frac{f\left(r_i^{(t)}\right)}{\prod\limits_{\substack{j=1 \\ j \neq i}}^{n}\left(r_i^{(t)} - r_j^{(t)}\right)^{\sigma_j}}, \qquad (i, j = 1, 2, 3, \ldots, n), \qquad (12)$$

where σ is the multiplicity of the roots.

2. Construction of the Inverse Simultaneous Method

Using Weierstrass correction $w\left(r_i^{(t)}\right) = f\left(r_i^{(t)}\right) / \prod_{\substack{j=1 \\ j \neq i}}^{n}\left(r_i^{(t)} - r_j^{(t)}\right)$ in (7), we get a new family of inverse modified Weierstrass method (abbreviated as MWM2):

$$s^{(t)} = \frac{\left(r^{(t)}\right)^2 * f'\left(r^{(t)}\right)}{r^{(t)} f'\left(r^{(t)}\right) + f\left(r^{(t)}\right)\left(1/\left(1 - \alpha f\left(r^{(t)}\right)/1 + f\left(r^{(t)}\right)\right)\right)}. \qquad (7)$$

If ζ is an exact root of (1), $f(\zeta) = 0$, and the following is obtained:

$$s_i^{(t)} = \frac{\left(r_i^{(t)}\right)^2\left(1 + (1 - \alpha)f\left(r_i^{(t)}\right)\right)}{r_i^{(t)}\left(1 + (1 - \alpha)f\left(r_i^{(t)}\right)\right) + w\left(r_i^{(t)}\right)\left(1 + f\left(r_i^{(t)}\right)\right)}. \qquad (13)$$

Inverse simultaneous iterative method (13) can also be written as

$$s_i^{(t)} = r_i^{(t)} - \frac{r_i^{(t)} w\left(r_i^{(t)}\right)\left(1 + f\left(r_i^{(t)}\right)\right)}{r_i^{(t)}\left(1 + (1 - \alpha)f\left(r_i^{(t)}\right)\right) + w\left(r_i^{(t)}\right)\left(1 + f\left(r_i^{(t)}\right)\right)}. \qquad (14)$$

Thus, we construct a new derivative-free family of inverse iterative simultaneous scheme (13), abbreviated as MWM2, for estimating all distinct roots of (1). To estimate all multiple roots of (1), we use correction (12) instead of (11) in (7).

2.1. Convergence Framework.
In this section, we demonstrate convergence theorem of inverse iterative scheme MWM2.

Theorem 1. Let $\zeta_1, \ldots, \zeta_n$ be single zero of (1) and for necessarily close primary distinct guess $r_1^{(0)}, \ldots, r_n^{(0)}$ of the zero, respectively; then, MWM2 has local 2nd-order convergence.

Proof. Let $\varepsilon_i = r_i^{(t)} - \zeta_i$ and $\varepsilon_i' = s_i^{(t)} - \zeta_i$ be the errors in r_i and s_i respectively. For the simplicity of the calculation, we omit the iteration index. Then,

$$s_i = \frac{\left(r_i\right)^2\left(1 + (1 - \alpha)f\left(r_i\right)\right)}{r_i\left(1 + (1 - \alpha)f\left(r_i\right)\right) + w\left(r_i\right)\left(1 + f\left(r_i\right)\right)}, \qquad (15)$$

or

$$s_i = r_i - \frac{r_i w\left(r_i\right)\left(1 + f\left(r_i\right)\right)}{r_i\left(1 + (1 - \alpha)f\left(r_i\right)\right) + w\left(r_i\right)\left(1 + f\left(r_i\right)\right)}. \qquad (16)$$

Using $f\left(r_i\right) / \prod_{\substack{j=1 \\ j \neq i}}^{n}\left(r_i - r_j\right) = w\left(r_i\right)$ in (15), we have

$$s_i = r_i - \frac{w\left(r_i\right)\left(1 + f\left(r_i\right)\right)}{\left(1 + (1 - \alpha)f\left(r_i\right)\right) + w\left(r_i\right)/r_i\left(1 + f\left(r_i\right)\right)}. \qquad (17)$$

Thus, we obtain

$$s_i - \zeta_i = r_i - \zeta_i - \frac{w(r_i)(1 + f(r_i))}{(1 + (1-\alpha)f(r_i)) + w(r_i)/r_i(1 + f(r_i))}, \tag{18}$$

$$\varepsilon_i' = \varepsilon_i \left[1 - \frac{\prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)/(r_i - r_j)\left(1 + \varepsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right)}{\left((1 + (1-\alpha))\varepsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right) + \left(\prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)/(r_i - r_j)\right)\left(1 + \varepsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right)} \right], \tag{19}$$

$$= \epsilon_i \left[\frac{\left(((1 + 1 - \alpha)\epsilon_i) \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right)}{\left((1 + (1-\alpha))\epsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right) + \left(\prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)/(r_i - r_j)\right)\left(1 + \epsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right)} \right]. \tag{20}$$

Using the expression $\prod_{\substack{j\neq i}}^{n}(r_i - \zeta_j)/(r_i - r_j) - 1 = \sum_{k\neq i}^{n}\epsilon_k/r_i - r_k\varepsilon\prod_{\substack{j\neq i}}^{k-1}(r_i - \zeta_k)/(r_i \overset{j=1}{=} r_j)$ [9] in (20), we have

$$\left\{\varepsilon_i' = \varepsilon_i \left[\frac{\left(((1 + 1 - \alpha)\varepsilon_i) \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right)}{\left((1 + 1 - \alpha)\varepsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right) + \left(\sum_{k\neq i}^{n}\varepsilon_k/r_i - r_k\prod_{j\neq i}^{k-1}(r_i - \zeta_k)/(r_i - r_j) + 1\right)\left(1 + \varepsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right)} \right]\right\}. \tag{21}$$

If all the errors are assumed of the same order, i.e., $|\epsilon_i| = |\epsilon_k| = O(|\epsilon|)$, then

$$\varepsilon_i' = |\varepsilon| \left[\frac{\left(((1 + (1-\alpha))\varepsilon_i)\varepsilon \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right)}{\left(1 + (1-\alpha)\varepsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right) + \left(\sum_{k\neq i}^{n}\varepsilon_k/r_i - r_k\prod_{j\neq i}^{k-1}(r_i - \zeta_k)/(r_i - r_j) + 1\right)\left(1 + \varepsilon_i \prod_{\substack{j\neq i \\ j=1}}^{n}(r_i - \zeta_j)\right)} \right], \tag{22}$$

$$= O(|\varepsilon|^2).$$

Hence, it is proved. □

3. Lower Bound of Convergence of MWM1 and MWM2

Computer algebra system, Mathematica, has been used to find the lower bound of convergence of MWM1 and MWM2.

Consider

$$f(r) = (r - \varrho_1)(r - \varrho_2)(r - \varrho_3), \tag{23}$$

where ϱ_1, ϱ_2, and ϱ_3 are exact zeros of (23). The first component of $\aleph_1(\mathbf{r})$ (where $\mathbf{r} = [r_1 r_2 r_3]$) of numerical iterative methods is for finding zeros of (23), $r^{(t+1)} = \aleph(r^{(t)})$,

simultaneously. We have to express the derivatives of $\aleph(r)$, i.e., the partial derivatives of $\aleph(r)$ with respect to $\mathbf{r}$ are as follows:

$$\frac{\partial \aleph_1(\mathbf{r})}{\partial r_1} \frac{\partial \aleph_1(\mathbf{r})}{\partial r_2} \frac{\partial \aleph_1(\mathbf{r})}{\partial r_3},$$

$$\frac{\partial_1^2 \aleph(\mathbf{r})}{\partial r_1^2} \frac{\partial_1^2 \aleph(\mathbf{r})}{\partial r_1 \partial r_2} \frac{\partial_1^2 \aleph(\mathbf{r})}{\partial r_2^2} \frac{\partial^2 \Gamma_1(\mathbf{r})}{\partial r_2 \partial r_3},$$

$$\frac{\partial_1^3 \aleph(\mathbf{r})}{\partial r_1^3} \frac{\partial_1^3 \aleph(\mathbf{r})}{\partial r_1^2 \partial r_2} \frac{\partial_1^3 \aleph(\mathbf{r})}{\partial r_1 \partial r_2^2} \frac{\partial_1^3 \aleph(\mathbf{r})}{\partial r_2^3} \frac{\partial_1^3 \aleph(\mathbf{r})}{\partial r_2^2 \partial r_3},$$

$$\tag{24}$$

and so on.

We obtain the lower bound of convergence order till the first nonzero element of row is found. The Mathematica

notebook codes are used for the following MWM1 and MWM2:

Weierstrass–Dochive method, MWM1:

$$\aleph_1(r_1, r_2, r_3) := r_1 - \frac{f(r_1)}{(r_1 - r_2) * (r_1 - r_3)}, \tag{25}$$

$$\text{In}[1] := \frac{D[\aleph_1[r_1, r_2, r_3], r_1]}{\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\}},$$

$$\text{Out}[1] := 0,$$

$$\text{In}[2] := D\frac{[\aleph_1[r_1, r_2, r_3], r_2]}{\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\}},$$

$$\text{Out}[2] := 0,$$

$$\text{In}[3] := D\frac{[\aleph_1[r_1, r_2, r_3], r_3]}{\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\}}, \tag{26}$$

$$\text{Out}[3] := 0,$$

$$\text{In}[4] := \text{Simplify} \frac{D[\aleph_1[r_1, r_2, r_3], r_1, r_2]}{\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\}},$$

$$\text{Out}[4] := 0,$$

$$\text{In}[6] := \text{Simplify} \frac{[D[\aleph_1[r_1, r_2, r_3], r_1, r_2]]}{\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\}},$$

$$\text{Out}[6] := \frac{1}{(\varrho_1 - \varrho_2)}.$$

Modified inverse family of iterative schemes, MWM2:

$$\aleph_1(r_1, r_2, r_3) := \frac{(r_1)^2 (1 + (1 - \alpha)f(r_1))}{r_1 (1 + (1 - \alpha)f(r_1)) + (r_1 - r_2)(r_1 - r_3)(1 + f(r_1))}, \tag{27}$$

$$\text{In}[1] := D[\aleph_1[r_1, r_2, r_3], r_2]/\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\},$$

$$\text{Out}[1] := 0,$$

$$\text{In}[2] := D[\aleph_1[r_1, r_2, r_3], r_3]/\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\},$$

$$\text{Out}[2] := 0,$$

$$\text{In}[3] := D[\aleph_1[r_1, r_2, r_3], r_3]/.\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\}, \tag{28}$$

$$\text{Out}[3] = 0,$$

$$\text{In}[4] := \text{Simplify}[D[\aleph_1[r_1, r_2, r_3], r_1, r_1]].\{r_1 \longrightarrow \varrho_1, r_2 \longrightarrow \varrho_2, r_3 \longrightarrow \varrho_3\},$$

$$\text{Out}[4] = \frac{4 + 8 * \alpha * \varrho_1}{\varrho_3^3}.$$

4. Numerical Results

Some engineering problems are considered to demonstrate the performance and effectiveness of the simultaneous method, MWM2 and MWM1. For computer calculations, we use CAS-Maple-18, and the following stopping criteria for termination of computer are programmed:

$$e_i^{(t)} = \left| r_i^{(t+1)} - r_i^{(t)} \right| < \epsilon = 10^{-30}, \tag{29}$$

where $e_i^{(t)}$ signifies the absolute error. In Tables 1–5, C-Time represents computational time in second.

4.1. Engineering Applications. Some engineering applications are deliberated in this section in order to show the feasibility of the present work.

Example 1. (see [2]). Considering a physical problem of beam positioning results in the following nonlinear polynomial equation:

$$f(r) = r^4 + 4r^3 - 24r^2 + 16r + 1,$$
$$= (r-2)^2 (r^2 + 8r + 4). \tag{30}$$

The exact root of (30), $\zeta_{1,2}$, is 2 with multiplicity 2 and the remaining other two roots are simple, i.e., $\zeta_3 = -4 - 2\sqrt{3}$ and $\zeta_3 = -4 + 2\sqrt{3}$. We take the following initial estimates:

$$\overset{(0)}{r_1} = 1.17,$$
$$\overset{(0)}{r_2} = 1.17,$$
$$\overset{(0)}{r_3} = -7.4641, \tag{31}$$
$$\overset{(0)}{r_4} = -0.5354.$$

Table 1 clearly demonstrates the superiority of MWM2 over MWM1 in terms of predicted absolute error and CPU time for guesstimating all real roots of (30) on the same number of iterations $n = 3$.

Example 2. (see [16]). In this engineering application, we consider a reactor of stirred tank. Items H_1 and H_2 are fed to the reactor at rates of ß and q-ß, respectively. Composite reaction improves in the apparatus as below:

$$H_1 + H_2 \longrightarrow H_3; H_3 + H_2 \longrightarrow H_4; H_4 + H_2 \longrightarrow H_5; H_4$$
$$+ H_2 \longrightarrow H_6. \tag{32}$$
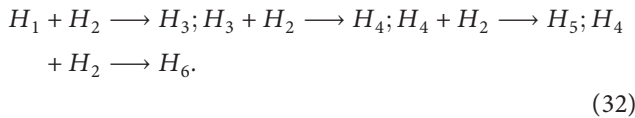
Douglas et al. [7] first examined this complex control system and obtained the following nonlinear polynomial equation:

$$-\frac{2.98 * (r + 2.25)}{(r + 1.45) * (r + 2.85)^2 * (r + 4.35)} = \frac{1}{\mathbf{T}_c}, \tag{33}$$

where $\mathbf{T}_c$ is the gain of the proportional controller. By taking $\mathbf{T}_c = 0$, we have

$$f(r) = r^4 + 11.50r^3 + 47.49r^2 + 83.06325r + 51.23266875 = 0. \tag{34}$$

The exact distinct roots of (34) are calculated as $\zeta_1 = -1.45, \zeta_2 = -2.85, \zeta_3 = -2.85,$ and $\zeta_4 = -4.45,$ and we take the following initial guessed values:

$$\overset{(0)}{r_1} = -1.0, \overset{(0)}{r_2} = -1.1, \overset{(0)}{r_3} = -1.8, \overset{(0)}{r_3} = -3.9. \tag{35}$$

Table 2 evidently illustrates the supremacy behavior of MWM2 over MWM1 in terms of the estimated absolute

TABLE 1: Residual error for finding all distinct roots.

Method	C-Time	$e_1^{(3)}$	$e_2^{(3)}$	$e_3^{(3)}$	$e_4^{(3)}$	$\sigma^{(2)}$
MWM1	1.103	Div	Div	$1.3e-27$	0.0	2.101
MWM2	0.048	0.0	0.0	0.0	0.0	2.307
Residual error for finding all multiple roots						
MWM2	0.312	0.0	0.0	0.0	0.0	2.735

TABLE 2: Residual error for finding all distinct roots.

Method	C-Time	$e_1^{(7)}$	$e_2^{(7)}$	$e_3^{(7)}$	$e_4^{(7)}$	$\sigma^{(6)}$
MWM1	2.119	0.07	0.02	0.1	0.1	2.125
MWM2	0.115	$3.5e-324$	$2.0e-319$	0.005	0.005	2.968

TABLE 3: Residual error for finding all distinct roots.

Method	C-Time	$e_1^{(8)}$	$e_2^{(8)}$	$e_3^{(8)}$	$e_4^{(8)}$	$\sigma^{(7)}$
MWM1	1.739	$1.1e-9$	Div	$3.6e-13$	Div	1.341
MWM2	0.032	$2.0e-317$	$4.0e-289$	$1.6e-139$	$4.0e-289$	3.167

TABLE 4: Residual error for finding all distinct roots.

Method	C-Time	$e_1^{(5)}$	$e_2^{(5)}$	$e_3^{(5)}$	$e_4^{(5)}$	$\sigma^{(4)}$
MWM1	0.125	$4.7e-26$	$1.2e-28$	$9.1e-28$	$0.2e-26$	2.314
MWM2	0.046	$1.3e-27$	$3.1e-30$	0.0	0.0	2.753
Residual error for finding all multiple roots at n = 1						
MWM2	0.103	0.0	0.0	0.0	0.0	

TABLE 5: Residual error for finding all distinct roots.

Method	C-Time	$e_1^{(8)}$	$e_2^{(8)}$	$e_3^{(8)}$	$\sigma^{(7)}$
MWM1	0.067	0.0	$6.4e-27$	$6.4e-27$	2.101
MWM2	0.043	0.0	0.0	0.0	2.231

error and in CPU time on the same number of iterations $n = 7$ for guesstimating all real roots of (34).

Example 3. (see [4]). Consider the function

$$\frac{8(4-r)^2 r^2}{(6-3r)^2 (2-r)} - 0.186 = 0, \tag{36}$$

$$f(r) = 8r^4 - 62.326r^3 + 117.956r^2 + 20.088r - 13.392. \tag{37}$$

The problem describes the fractional alteration of nitrogen-hydrogen (NH) feed into ammonia at 250 atm pressure and 500° C temperature. Since the (37) is of order four, it has four roots:

$$\zeta_1 = 0.2777,$$
$$\zeta_2 = 3.9485 + 0.3161i,$$
$$\zeta_3 = -0.3840,$$
$$\zeta_4 = 3.9485 - 0.3161i. \tag{38}$$

The initial approximated value for (27) is taken as

$$\overset{(0)}{r_1} = 0.4, \overset{(0)}{r_2} = 3.7 + 0.5i, \overset{(0)}{r_3} = -0.4, \overset{(0)}{r_4} = 3.7 - 0.5i. \tag{39}$$

Table 3 evidently shows the supremacy behavior of MWM2 over MWM1 in terms of estimated absolute error and in CPU time on the same number of iterations $n = 8$ for guesstimating all real and complex roots of (37). Minuscule alteration of nitrogen-hydrogen (NH) feed into ammonia lies between (0,1); therefore, our desire root is ζ_1 up to 1900 decimal places:

$$\zeta_1 = 1.1295656841257983352197345 2e - 1912. \tag{40}$$

Remaining other approximating roots are ζ_2 $= -1.283\,404\,526e - 1457 - 8.93\,219\,631\,521e - 1457i$, ζ_3 $= -8.21\,745\,235\,223\,462e - 1091 + 0i$, and $\zeta_4 =$ $-1.2\,834\,045\,268\,801e - 1457 + 8.99\,321\,963\,152e - 1457i$.

Example 4. (see [8]).Consider

$$f(r) = (r - 0.3 - 0.6i)^{100}(r - 0.1 - 0.7i)^{200}(r - 0.7 - 0.5i)^{300}$$
$$(r - 0.3 - 0.4i)^{400}, \tag{41}$$

with multiple exact roots:

$$\zeta_1 = 0.3 + 0.6i, \zeta_2 = 0.1 + 0.7i, \zeta_3 = 0.7 + 0.5i, \zeta_4 = 0.3 + 0.4i, \tag{42}$$

The initial estimations have been taken as

$$\overset{(0)}{r_1} = 0.301 + 0.601i, \overset{(0)}{r_2} = 0.100 + 0.702i, \overset{(0)}{r_3}$$
$$= 0.702 + 0.489i, \overset{(0)}{r_4} = 0.289 - 0.400i, \tag{43}$$

For distinct roots,

$$f_*(r) = (r - 0.3 - 0.6i)(r - 0.1 - 0.7i)(r - 0.7 - 0.5i)$$
$$(r - 0.3 - 0.4i), \tag{44}$$

Table 4 evidently shows the supremacy behavior of MWM2 over MWM1 in terms of estimated absolute error and in CPU time on the same number of iterations $n = 5$ for guesstimating all real and complex roots of (41).

Example 5. (see [5]). The sourness of a soaked solution of magnesium-hydroxide (MgOH) in hydroelectric acid (HCl) is given by

$$\frac{3.64 \times 10^{-11}}{[H_3O^+]} = [H_3O^+] + 3.6 \times 10^{-4}, \tag{45}$$

TABLE 6: Residual error for finding all distinct roots.

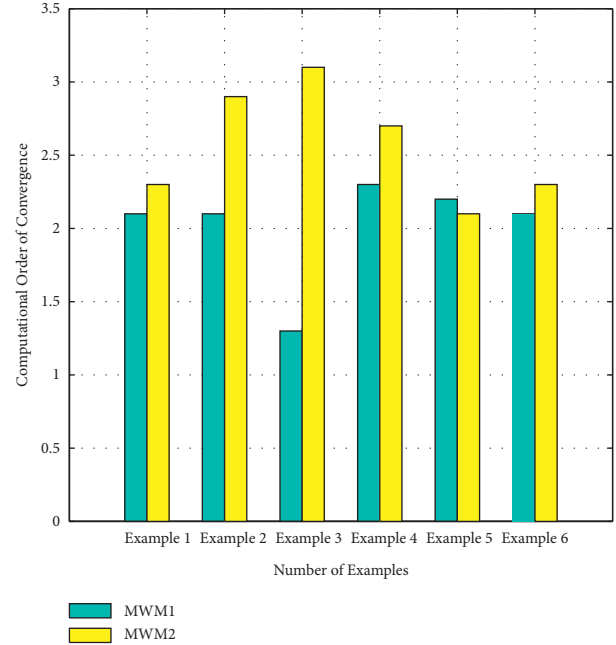| Method	C-Time	$e_1^{(4)}$	$e_2^{(4)}$	$e_3^{(4)}$	$e_4^{(4)}$	$\sigma^{(3)}$
MWM1	0.203	$4.1e-25$	$3.6e-30$	$7.1e-21$	$5.6e-23$	2.131
MWM2	0.115	$5.0e-38$	$4.8e-37$	$8.4e-32$	$7.9e-32$	2.707

FIGURE 1: Computational order of convergence [22] of simultaneous methods, MWM1 and MWM2.
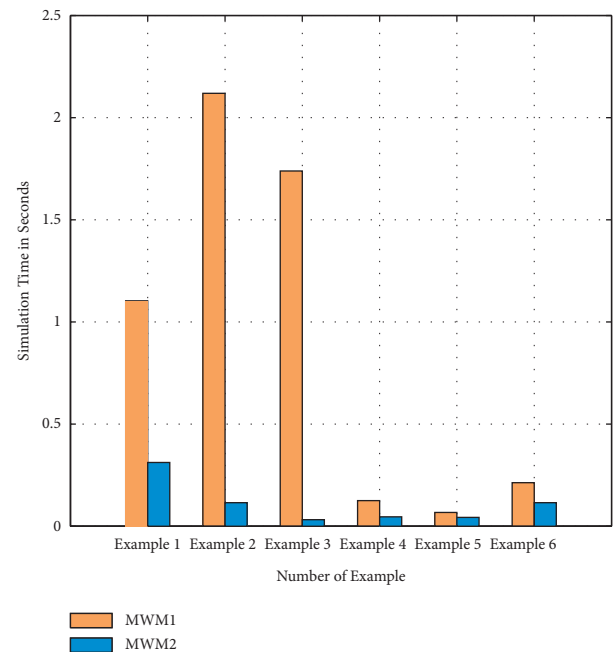

FIGURE 2: Simulation time in second of simultaneous methods, MWM1 and MWM2, for approximating all roots of polynomial equations used in Examples 1–6.
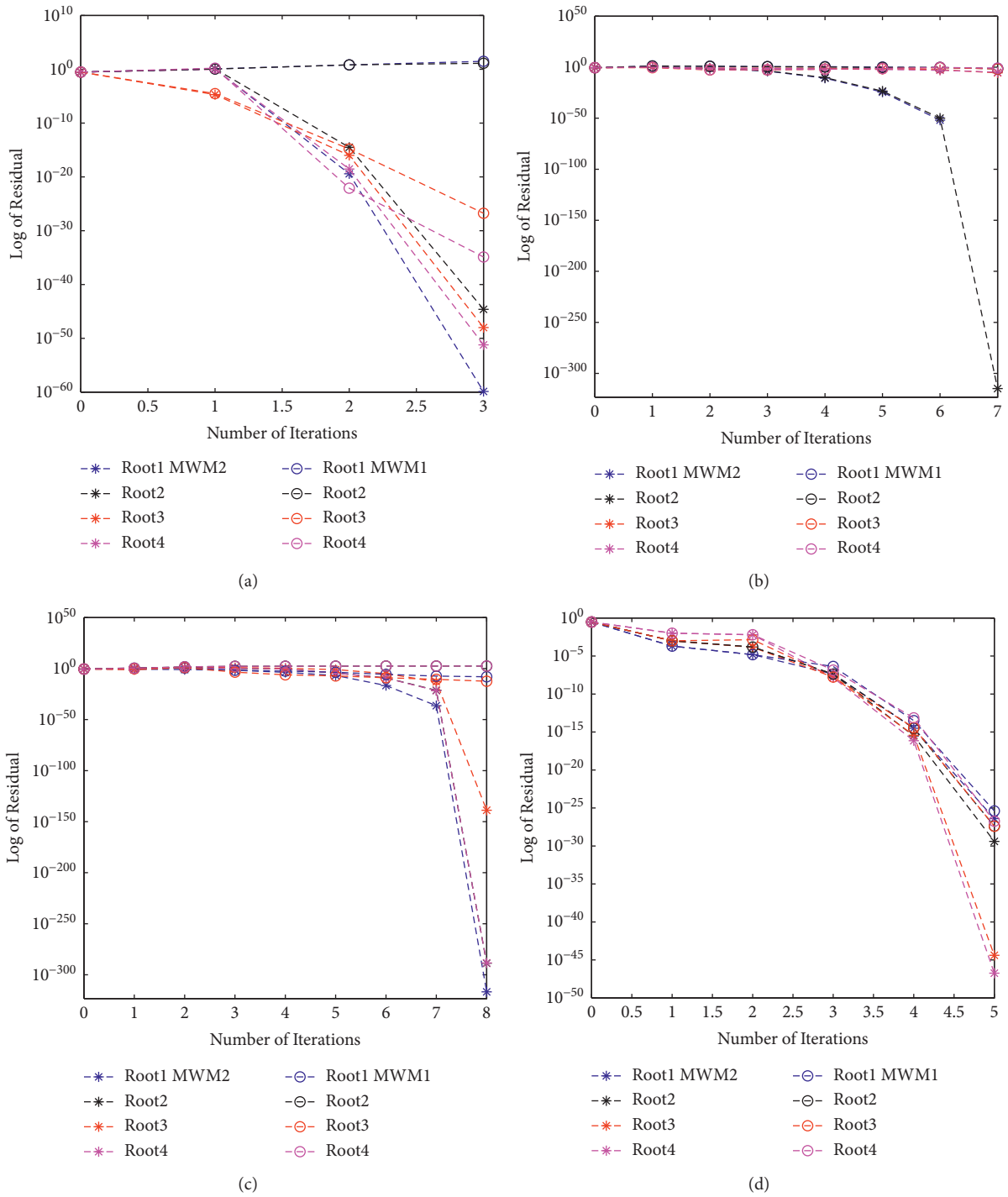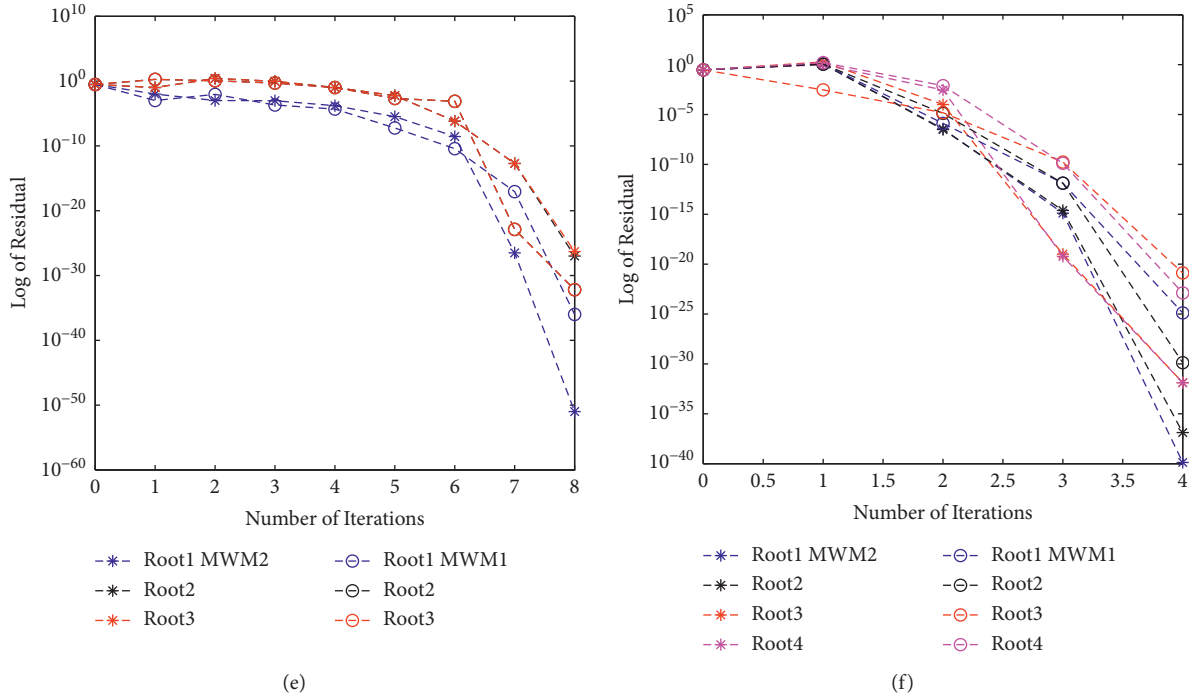
FIGURE 3: Continued.

(e)

(f)

Figure 3: (a-f) Error graph of MWM1 and MWM2 for nonlinear polynomial equations (30), (34), (37), (41), (46) and (49), respectively. (a) Residual error graph for Example 1. (b) Residual error graph for Example 2. (c) Residual error graph for Example 3. (d) Residual error graph for Example 4. (e) Residual error graph for Example 5. (f) Residual error graph for Example 6.

for the cranium ion concentration $[H_3O^+]$. If we set $r = 10^4[H_3O^+]$, we obtain the following polynomial:

$$f(r) = r^3 + 3.6r^2 - 36.4, \tag{46}$$

with exact roots of (46), $\zeta_1 = 2.4$ and $\zeta_{2,3} = -3.0 \pm 2.3i$ up to one decimal places. The initial estimates have been taken as

$$\overset{(0)}{r_1} = 2.45, \overset{(0)}{r_2} = -3.0261 + 2.3834i, \overset{(0)}{r_3} = -3.0261 - 2.3834i. \tag{47}$$

Table 5 evidently illustrates the supremacy behavior of MWM2 over MWM1 in terms of estimated absolute error and in CPU time on the same number of iterations $n = 8$ for guesstimating all real and complex roots of (46).

Example 6. (see [21]). In general, mechanical engineering, as well as the majority of other scientists, uses thermodynamics extensively in their research work. The following polynomial is used to relate the zero-pressure specific heat of dry air, C_ρ, to temperature:

$$C_\rho = 1.9520 \times 10^{-14} r^4 - 9.5838 \times 10^{-11} r^3 + 9.7215 \times 10^{-8} r^2 + 1.671 \times 10^{-4} r + 0.99403. \tag{48}$$

The temperature that corresponds to specific heat of $1.2 (kJ/kgK)$ needs to be determined. Putting $C_\rho = 1.2$ in (48), we have

$$f(r) = 1.9520 \times 10^{-14} r^4 - 9.5838 \times 10^{-11} r^3 + 9.7215 \times 10^{-8} r^2 + 1.671 \times 10^{-4} r + 0.99403. \tag{49}$$

with exact roots $\zeta_1 = 1126.009751$, $\zeta_2 = 2536.837119 + 910.5010371i$, and $\zeta_3 = -1289.950382, 2536.837119 - 910.5010371i$. The initial estimations of (49) have been taken as

$$\overset{(0)}{r_1} = 1126, \overset{(0)}{r_2} = 2536 + 910i, \overset{(0)}{r_3} = -1289, \overset{(0)}{r_4} = 2536 - 910i. \tag{50}$$

Table 6 clearly illustrates the supremacy behavior of MWM1 over MWM2 in estimated absolute error and in CPU time on the same number of iterations $n = 4$ for guesstimating all real and complex roots of (49).

5. Conclusion

A new derivative-free family of inverse numerical methods of convergence order 2 for simultaneous estimations of all distinct and multiple roots of (1) was introduced and discussed in this paper. Tables 1–5 and Figure 1 clearly show that computational order of convergence of the proposed and existing methods are agreed with the theoretical results. Simulation time, from Figure 2, clearly indicates the supremacy of our newly proposed method MWM2 over

existing Weierstrass method MWM1. The results of numerical test cases from Tables 1–5, CPU time, and residual error graph from Figure 3 demonstrated the effectiveness and rapid convergence of our proposed iterative method MWM2 as compared to MWM1.

Data Availability

No data were used to support this study.

Disclosure

The statements made and views expressed are solely the responsibility of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed equally and significantly in writing this article. All authors read and approved the final manuscript.

References

[1] M. Shams, N. Rafiq, and N. Kausar, "On highly efficient derivative-free family of numerical methods for solving polynomial equation simultaneously," *Advances in Difference Equations*, vol. 2021, p. 465, 2021.

[2] F. I. Chicharro, A. Cordero, and J. R. Garrido N &Torregrosa, "Stability and applicability of iterative methods with memory," *Journal of Mathematical Chemistry*, vol. 2, 2018.

[3] F. I. Chicharro, A. Cordero, and J. R. Garrido N &Torregrosa, "Generating root-finder iterative methods of second order convergence and stability,axioms," 2019.

[4] A. Cordero, H. Ramos, and J. R. Torregrosa, "Some variants of Halley's method with memory and their applications for solving several chemical problems," *Journal of Mathematical Chemistry*, vol. 58, no. 4, pp. 751–774, 2020.

[5] Y. M. Chu, N. Rafiq, M. Shams, S. Akram, N. A. Mir, and H. Kalsoom, "Computer methodologies for the comparison of some efficient derivative free simultaneous iterative methods for finding roots of non-linear equations," vol. 16, Computers, Materials & Continua, 2020.

[6] S. I. Cholakov, "Local and semilocal convergence of Wang-Zheng's method for simultaneous finding polynomial zeros," *Symmetry*, vol. 736, p. 15, 2019.

[7] J. M. Douglas, *Process Dynamics and Control*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1972.

[8] 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, London, UK, 2014.

[9] S. Kanno and T. Yamamoto, "On some methods for the simultaneous determination of polynomial zeros," *Japan Journal of Applied Mathematics*, vol. 13, pp. 267–288, 1995.

[10] 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.

[11] 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.

[12] A. W. M. Anourein, "An improvement on two iteration methods for simultaneous determination of the zeros of a polynomial," *International Journal of Computer Mathematics*, vol. 6, no. 3, pp. 241–252, 1977.

[13] 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.

[14] P. D. Proinov and M. T. Vasileva, "On a family of Weierstrass-type root-finding methods with accelerated convergence," *Applied Mathematics and Computation*, vol. 273, pp. 957–968, 2016.

[15] N. Rafiq, S. Akram, N. A. Mir, and M. Shams, "Study of dynamical behaviour and stability of iterative methods for non-linear equationswith application in engineering," *Mathematical Problems in Engineering*, vol. 2020, p. 20, Article ID 3524324, 2020.

[16] M. Shams, N. A. Mir, N. Rafiq, and S. Akram, "On dynamics of iterative techniques for non-linear equations with application in engineering," *Mathematical Problems in Engineering*, vol. 2020, p. 17, Article ID 5853296, 2020.

[17] K. N. Weierstrass & Beweis, "Des Satzes dass jede ganze rationale Function einer Ver̈anderlichen dargestellt werden kann als ein Product aus linearen Functionen derselben Ver̈anderlichen," *Sitzungsber. K̈onigl. Preuss. Akad. Wiss. Berlinn II*, vol. 32, pp. 1085–1101, 1891.

[18] F. Ahmad, E. Tohidi, and J. A. Carrasco, "A parameterized multi-step Newton method for solving systems of nonlinear equations," *Numerical Algorithms*, vol. 71, no. 3, pp. 631–653, 2016.

[19] F. Ahmad, E. Tohidi, M. Z. Ullah, and J. A. Carrasco, "Higher order multi-step Jarratt-like method for solving systems of nonlinear equations: application to PDEs and ODEs," *Computers & Mathematics with Applications*, vol. 70, no. 4, pp. 624–636, 2015.

[20] F. Soleymani, R. Sharma, X. Li, and E. Tohidi, "An optimized derivative- free form of the Potra–Pták method," *Mathematical and Computer Modelling*, vol. 56, no. 5-6, pp. 97–104, 2012.

[21] D. V. Griffithms and I. M. Smith, *Numerical Methods for Engineers*, Chapman and Hall/CRC (Taylor and Francis Grpup), Special Indian Edition, Stanford, CL, USA, 2011.

[22] M. G-Sanchez, M. Noguera, A. Grau, and J. R. Herrero, "On new computational local orders of convergence," *Applied Mathematics Letters*, vol. 25, no. 12, pp. 2023–2030, 2012.