

RESEARCH

Open Access



# Efficient iterative methods for finding simultaneously all the multiple roots of polynomial equation

Mudassir Shams<sup>1</sup>, Naila Rafiq<sup>2</sup>, Nasreen Kausar<sup>3</sup>, Praveen Agarwal<sup>4,5</sup>, Choonkil Park<sup>6\*</sup>  and Shaher Momani<sup>7</sup>

\*Correspondence:  
baak@hanyang.ac.kr

<sup>6</sup>Department of Mathematics,  
Research Institute of Natural  
Science, Hanyang University, Seoul,  
South Korea

Full list of author information is  
available at the end of the article

## Abstract

Two new iterative methods for the simultaneous determination of all multiple as well as distinct roots of nonlinear polynomial equation are established, using two suitable corrections to achieve a very high computational efficiency as compared to the existing methods in the literature. Convergence analysis shows that the orders of convergence of the newly constructed simultaneous methods are 10 and 12. At the end, numerical test examples are given to check the efficiency and numerical performance of these simultaneous methods.

**Keywords:** Multiple roots; Polynomial equation; Iterative methods; Simultaneous methods; Computational efficiency and CPU-time

## 1 Introduction

A wide range of theoretical and practical problems arise in various fields of mathematical, economical, physical, and engineering sciences which can be formulated as a polynomial equation of degree  $n$  with arbitrary real or complex coefficient:

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

where  $\zeta_1 \dots \zeta_n$  denote all the simple or complex roots of (1). Approximating all roots of the nonlinear polynomial equation using simultaneous methods has a lot of applications in sciences and engineering because simultaneous iterative methods are less time consuming since they can be implemented for parallel processing as well. Further details about their convergence properties, computational efficiency, and parallel processing may be found in [1–25] and the references cited there in. The main objective of this paper is to develop simultaneous methods which have a higher convergence order and are more efficient as compared to the existing methods. A very high computational efficiency is achieved by using two suitable corrections [26, 27] with convergence orders equal to ten and twelve with a minimal number of function evaluations in each step.

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### 1.1 Construction of simultaneous methods for multiple roots

Consider two-step fourth-order Newton’s method [26] for finding multiple roots of non-linear equation (1)

$$\begin{cases} y_i = x_i - \sigma \frac{f(x_i)}{f'(x_i)}, \\ z_i = y_i - \sigma \frac{f(y_i)}{f'(y_i)}, \end{cases} \tag{2}$$

where  $\sigma$  is the multiplicity of exact root, say  $\zeta$ , of (1). We would like to convert (2) into a simultaneous method for extracting all the distinct as well as multiple roots of (1). We use the third-order Dong et al. method [26] as a correction to increase the efficiency and convergence order requiring no additional evaluation of the function:

$$\begin{cases} v_i = x_i - \sqrt{\sigma} \frac{f(x_i)}{f'(x_i)}, \\ u_i = v_i - \sigma \left(1 - \frac{1}{\sqrt{\sigma}}\right)^{1-\sigma} \frac{f(v_i)}{f'(v_i)}. \end{cases} \tag{3}$$

Suppose that the nonlinear polynomial equation (1) has  $n$  roots. Then

$$f(x) = \prod_{i=1}^n (x - x_i) \quad \text{and} \quad f'(x) = \sum_{j=1}^n \prod_{\substack{i=1 \\ i \neq j}}^n (x - x_i). \tag{4}$$

This implies

$$\frac{f(x_i)}{f'(x_i)} = \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{(x - x_i)} = \frac{1}{x - x_i} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{(x - x_j)}.$$

This gives

$$x - x_i = \frac{1}{\frac{1}{N_i(x_i)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{(x - x_j)}},$$

where  $\frac{1}{N_i(x_i)} = \frac{f'(x_i)}{f(x_i)}$  or

$$\frac{f(x_i)}{f'(x_i)} = \frac{1}{\frac{1}{N_i(x_i)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{(x - x_j)}}. \tag{5}$$

The multiple root equation (5) can be written as

$$\sigma_i \frac{f(x_i)}{f'(x_i)} = \frac{\sigma_i}{\frac{\sigma_i}{N_i(x_i)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x - x_j)}}. \tag{6}$$

Replacing  $x_j$  by  $x_j^*$  in (6), we have

$$\sigma_i \frac{f(x_i)}{f'(x_i)} = \frac{\sigma_i}{\frac{\sigma_i}{N_i(x_i)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x - x_j^*)}}, \tag{7}$$

where

$$x_j^* = u_j \quad (\text{using (3)}).$$

Using (7) in the first step of (2), we have

$$\begin{cases} y_i^{(k)} = x_i^{(k)} - \frac{\sigma_i}{\frac{\sigma_i}{N_i(x_i^{(k)})} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i^{(k)} - x_j^{*(k)})}}, & k = 0, 1, \dots, \\ z_i^{(k)} = y_i^{(k)} - \frac{\sigma_i}{\frac{\sigma_i}{N_i(y_i^{(k)})} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(y_i^{(k)} - y_j^{(k)})}}. \end{cases} \tag{8}$$

Thus we have constructed a new simultaneous method (8) abbreviated as MNS10M for extracting all distinct as well as multiple roots of polynomial equation (1).

### 1.2 Convergence analysis

In this section, the convergence analysis of a family of two-step simultaneous methods (8) given in a form of the following theorem is presented.

**Theorem 1** *Let  $\zeta_1, \dots, \zeta_n$  be simple roots of (1). If  $x_1^{(0)}, \dots, x_n^{(0)}$  are the initial approximations of the roots respectively and sufficiently close to the actual roots, then the order of convergence of method (8) equals ten.*

*Proof* Let  $\epsilon_i = x_i - \zeta_i, \epsilon'_i = y_i - \zeta_i$ , and  $\epsilon''_i = z_i - \zeta_i$  be the errors in  $x_i, y_i$ , and  $z_i$  approximations respectively. Consider the first step of (8), which is

$$y_i = x_i - \frac{\sigma_i}{\frac{\sigma_i}{N(x_i)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i - x_j^*)}},$$

where  $N(x_i) = \frac{f'(x_i)}{f(x_i)}$ . Then, obviously, for distinct roots, we have

$$\frac{1}{N(x_i)} = \frac{f'(x_i)}{f(x_i)} = \sum_{j=1}^n \frac{1}{(x_i - \zeta_j)} = \frac{1}{(x_i - \zeta_i)} + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{(x_i - \zeta_j)}.$$

Thus, for multiple roots, we have from (8)

$$\begin{aligned} y_i &= x_i - \frac{\sigma_i}{\frac{\sigma_i}{(x_i - \zeta_i)} + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i - \zeta_j)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i - x_j^*)}}, \\ y_i - \zeta_i &= x_i - \zeta_i - \frac{\sigma_i}{\frac{\sigma_i}{(x_i - \zeta_i)} + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j(x_i - x_j^* - x_i + \zeta_j)}{(x_i - \zeta_j)(x_i - x_j^*)}}, \\ \epsilon'_i &= \epsilon_i - \frac{\sigma_i}{\epsilon_i + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{-\sigma_j(x_j^* - \zeta_j)}{(x_i - \zeta_j)(x_i - x_j^*)}} = \epsilon_i - \frac{\sigma_i \epsilon_i}{\sigma_i + \epsilon_i \sum_{\substack{j=1 \\ j \neq i}}^n \frac{-\sigma_j(x_j^* - \zeta_j)}{(x_i - \zeta_j)(x_i - x_j^*)}} = \epsilon_i - \frac{\sigma_i \cdot \epsilon_i}{\sigma_i + \epsilon_i \sum_{\substack{j=1 \\ j \neq i}}^n E_i \epsilon_j^3}, \end{aligned}$$

where  $x_j^* - \zeta_j = \epsilon_j^3$  [26] and  $E_i = \frac{-\sigma_j}{(x_i - \zeta_j)(x_i - x_j^*)}$ .

Thus

$$\epsilon'_i = \frac{\epsilon_i^2 \sum_{j \neq i}^n E_i \epsilon_j^3}{\sigma_i + \epsilon_i \sum_{j \neq i}^n E_i \epsilon_j^3}. \tag{9}$$

If it is assumed that absolute values of all errors  $\epsilon_j$  ( $j = 1, 2, 3, \dots$ ) are of the same order as, say,  $|\epsilon_j| = O(\epsilon)$ , then from (9) we have

$$\epsilon'_i = O(\epsilon)^5. \tag{10}$$

From the second equation of (8), we get

$$\begin{aligned} z_i &= y_i - \frac{\sigma_i}{N(y_i) - \sum_{j \neq i}^n \frac{\sigma_j}{(y_i - y_j)}}, \\ z_i - \zeta_i &= y_i - \zeta_i - \frac{\sigma_i}{\frac{\sigma_i}{y_i - \zeta_i} + \sum_{j \neq i}^n \frac{\sigma_j}{(y_i - \zeta_j)} - \sum_{j \neq i}^n \frac{\sigma_j}{(y_i - y_j)}}, \\ \epsilon''_i &= \epsilon'_i - \frac{\sigma_i}{\frac{\sigma_i}{\epsilon'_i} + \sum_{j \neq i}^n \frac{\sigma_j}{(y_i - \zeta_j)} - \sum_{j \neq i}^n \frac{\sigma_j}{(y_i - y_j)}} = \epsilon'_i - \frac{\sigma_i \cdot \epsilon'_i}{\sigma_i + \epsilon'_i (\sum_{j \neq i}^n \frac{\sigma_j (y_i - y_j - y_i + \zeta_j)}{(y_i - \zeta_j)(y_i - y_j)})} \\ &= \epsilon'_i - \frac{\sigma_i \cdot \epsilon'_i}{\sigma_i + \epsilon'_i (\sum_{j \neq i}^n \frac{-\sigma_j (y_j - \zeta_j)}{(y_i - \zeta_j)(y_i - y_j)}) - \epsilon'_i \alpha} = \epsilon'_i - \frac{\sigma_i \epsilon'_i}{\sigma_i + \epsilon'_i \sum_{j \neq i}^n \epsilon'_j F_i - \epsilon'_i \alpha}, \end{aligned}$$

where  $F_i = \frac{-\sigma_j}{(y_i - \zeta_j)(y_i - y_j)}$ . This implies

$$\epsilon''_i = \epsilon'_i - \frac{\sigma_i \cdot \epsilon'_i}{\sigma_i + \epsilon'_i (\sum_{j \neq i}^n \epsilon'_j F_i - \alpha \epsilon'_i)} = (\epsilon'_i)^2 \left( \frac{\sum_{j \neq i}^n F_i - \alpha}{\sigma_i + \epsilon'_i (\sum_{j \neq i}^n \epsilon'_j F_i - \alpha)} \right) = (\epsilon'_i)^2 C_i,$$

where  $C_i = \frac{\sum_{j \neq i}^n \epsilon'_j F_i - \alpha}{\sigma_i + \epsilon'_i \sum_{j \neq i}^n (\epsilon'_j F_i - \alpha)}$ . By (10),  $\epsilon'_i = O(\epsilon)^5$  and thus

$$\epsilon''_i = O((\epsilon)^5)^2 = O(\epsilon)^{10},$$

which shows that the convergence order of method (8) is ten. Hence we have proved the theorem. □

### 1.3 Improvement of efficiency and convergence order

To improve the convergence order of method (8) from 10 to 12, using same function evaluation, we use

$$z_j^* = v_j - \sigma_j \frac{f(v_j)}{f'(v_j)} \quad \text{and} \quad v_j = x_j - \sqrt{\sigma_j} \frac{f(x_j)}{f'(x_j)}$$

instead of  $x_j^* = Z_j^*$  in (7), i.e.,

$$\sigma_i \frac{f(x_i)}{f'(x_i)} = \frac{\sigma_i}{\frac{\sigma_i}{N_i(x_i)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i - Z_j^*)}}, \tag{11}$$

where  $Z_j^*$  is a fourth-order method [27]. Using (11) in the first step of (2), we have

$$\begin{cases} y_i^{(k)} = x_i^{(k)} - \frac{\sigma_i}{\frac{\sigma_i}{N_i(x_i^{(k)})} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i^{(k)} - Z_j^{*(k)})}}, \\ z_i^{(k)} = y_i^{(k)} - \frac{\sigma_i}{\frac{\sigma_i}{N_i(y_i^{(k)})} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(y_i^{(k)} - y_j^{(k)})}}. \end{cases} \tag{12}$$

Thus we have constructed a new simultaneous method (12), abbreviated as MNS12M for extracting all multiple roots of polynomial equation (1). For multiplicity unity, we used method (12) for determining all the distinct roots of (1), abbreviated as MNS12D.

### 1.4 Convergence analysis

In this section, the convergence analysis of a family of two-step simultaneous methods (12) is given in a form of the following theorem.

**Theorem 2** *Let  $\zeta_1, \zeta_2, \dots, \zeta_n$  be simple roots of (1). If  $x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \dots, x_n^{(0)}$  are the initial approximations of the roots respectively and sufficiently close to the actual roots, then the order of convergence of method (12) equals twelve.*

*Proof* Let  $\epsilon_i = x_i - \zeta_i, \epsilon'_i = y_i - \zeta_i$ , and  $\epsilon''_i = z_i - \zeta_i$  be the errors in  $x_i, y_i$ , and  $z_i$  approximations respectively. Consider the first step of (12), which is

$$y_i = x_i - \frac{\sigma_i}{\frac{\sigma_i}{N(x_i)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i - Z_j^*)}},$$

where  $N(x_i) = \frac{f(x_i)}{f'(x_i)}$ . Then, obviously, for distinct roots, we have

$$\frac{1}{N(x_i)} = \frac{f'(x_i)}{f(x_i)} = \sum_{j=1}^n \frac{1}{(x_i - \zeta_j)} = \frac{1}{(x_i - \zeta_i)} + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{(x_i - \zeta_j)}.$$

Thus, for multiple roots, we have from (6)

$$\begin{aligned} y_i &= x_i - \frac{\sigma_i}{\frac{\sigma_i}{(x_i - \zeta_i)} + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i - \zeta_j)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(x_i - Z_j^*)}}, \\ y_i - \zeta_i &= x_i - \zeta_i - \frac{\sigma_i}{\frac{\sigma_i}{(x_i - \zeta_i)} + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j(x_i - Z_j^* - x_i + \zeta_j)}{(x_i - \zeta_j)(x_i - Z_j^*)}}, \\ \epsilon'_i &= \epsilon_i - \frac{\sigma_i}{\epsilon_i + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{-\sigma_j(Z_j^* - \zeta_j)}{(x_i - \zeta_j)(x_i - Z_j^*)}} = \epsilon_i - \frac{\sigma_i \epsilon_i}{\sigma_i + \epsilon_i \sum_{\substack{j=1 \\ j \neq i}}^n \frac{-\sigma_j(Z_j^* - \zeta_j)}{(x_i - \zeta_j)(x_i - Z_j^*)}} = \epsilon_i - \frac{\sigma_i \epsilon_i}{\sigma_i + \epsilon_i \sum_{\substack{j=1 \\ j \neq i}}^n G_j \epsilon_j^4}, \end{aligned}$$

where  $Z_j^* - \zeta_j = \epsilon_j^4$  [27] and  $G_i = \frac{-\sigma_j}{(x_i - \zeta_j)(x_i - Z_j^*)}$ . Thus

$$\epsilon'_i = \frac{\epsilon_i^2 \sum_{\substack{j=1 \\ j \neq i}}^n G_i \epsilon_j^4}{\sigma_i + \epsilon_i \sum_{\substack{j=1 \\ j \neq i}}^n G_i \epsilon_j^4}. \tag{13}$$

If it is assumed that absolute values of all errors  $\epsilon_j$  ( $j = 1, 2, 3, \dots$ ) are of the same order as, say,  $|\epsilon_j| = O|\epsilon|$ , then from (13) we have

$$\epsilon'_i = O(\epsilon)^6. \tag{14}$$

From the second equation of (12), we have

$$\begin{aligned} z_i &= y_i - \frac{\sigma_i}{\frac{\sigma_i}{N(y_i)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(y_i - y_j)}}, \\ z_i - \zeta_i &= y_i - \zeta_i - \frac{\sigma_i}{\frac{\sigma_i}{y_i - \zeta_i} + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(y_i - \zeta_j)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(y_i - y_j)}}, \\ \epsilon''_i &= \epsilon'_i - \frac{\sigma_i}{\frac{\sigma_i}{\epsilon'_i} + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(y_i - \zeta_j)} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j}{(y_i - y_j)}} = \epsilon'_i - \frac{\sigma_i \epsilon'_i}{\sigma_i + \epsilon'_i \left( \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\sigma_j (y_i - y_j - y_i + \zeta_j)}{(y_i - \zeta_j)(y_i - y_j)} \right)}, \\ &= \epsilon'_i - \frac{\sigma_i \epsilon'_i}{\sigma_i + \epsilon'_i \left( \sum_{\substack{j=1 \\ j \neq i}}^n \frac{-\sigma_j (y_j - \zeta_j)}{(y_i - \zeta_j)(y_i - y_j)} \right)} = \epsilon'_i - \frac{\sigma_i \epsilon'_i}{\sigma_i + \epsilon'_i \sum_{\substack{j=1 \\ j \neq i}}^n \epsilon'_j H_i}, \end{aligned}$$

where  $H_i = \frac{-\sigma_j}{(y_i - \zeta_j)(y_i - y_j)}$ . This implies

$$\epsilon''_i = \epsilon'_i - \frac{\sigma_i \epsilon'_i}{\sigma_i + \epsilon'_i \left( \sum_{\substack{j=1 \\ j \neq i}}^n \epsilon'_j H_i \right)}.$$

If it is assumed that absolute values of all errors  $\epsilon_j$  ( $j = 1, 2, 3, \dots$ ) are of the same order as, say,  $|\epsilon_j| = O|\epsilon|$ , then we have

$$= (\epsilon'_i)^2 \left( \frac{\sum_{\substack{j=1 \\ j \neq i}}^n H_i}{\sigma_i + (\epsilon'_i)^2 \left( \sum_{\substack{j=1 \\ j \neq i}}^n H_i \right)} \right) = (\epsilon'_i)^2 D_i,$$

where  $D_i = \frac{\sum_{\substack{j=1 \\ j \neq i}}^n H_i}{\sigma_i + (\epsilon'_i)^2 \sum_{\substack{j=1 \\ j \neq i}}^n H_i}$ . By (14),  $\epsilon'_i = O(\epsilon)^6$  and thus

$$\epsilon''_i = O((\epsilon)^6)^2 = O(\epsilon)^{12},$$

which shows that the convergence order of method (12) is twelve. Hence we have proved the theorem. □

**Table 1** Number of operations (real arithmetic)

Methods	$AS_m$	$M_m$	$D_m$
New method (8)	$19m^2 + O(m)$	$12m^2 + O(m)$	$2m^2 + O(m)$
Petkovic method (PJM10D)	$22m^2 + O(m)$	$18m^2 + O(m)$	$2m^2 + O(m)$
New method (12)	$18m^2 + O(m)$	$10m^2 + O(m)$	$2m^2 + O(m)$

### 2 Computational analysis

Here we compare the computational efficiency and convergence behavior of the Petkovic et al. [28] method (abbreviated as PJM10D) and the new simultaneous iterative methods (8) and (12). As presented in [28], the efficiency of an iterative method can be estimated using the efficiency index given by

$$EF(n) = \frac{\log r}{D}, \tag{15}$$

where  $D$  is the computational cost and  $r$  is the order of convergence of the iterative method. The number of addition and subtraction, multiplications, and divisions per iteration for all  $n$  roots of a given polynomial of degree  $m$  is denoted by  $AS_m$ ,  $M_m$ , and  $D_m$ . The computational cost can be approximated as

$$D = D(m) = w_{as}AS_m + w_mM_m + w_dD_m, \tag{16}$$

and thus (15) becomes

$$EF(m) = \frac{\log r}{w_{as}AS_m + w_mM_m + w_dD_m}. \tag{17}$$

Applying (17) and by data given in Table 1, we calculate the percentage ratio  $\rho((8), (X))$  and  $\rho((12), (X))$  [28] given by

$$\rho((8), (X)) = \left( \frac{EF(8)}{EF(X)} - 1 \right) \times 100 \quad (\text{in percent}), \tag{18}$$

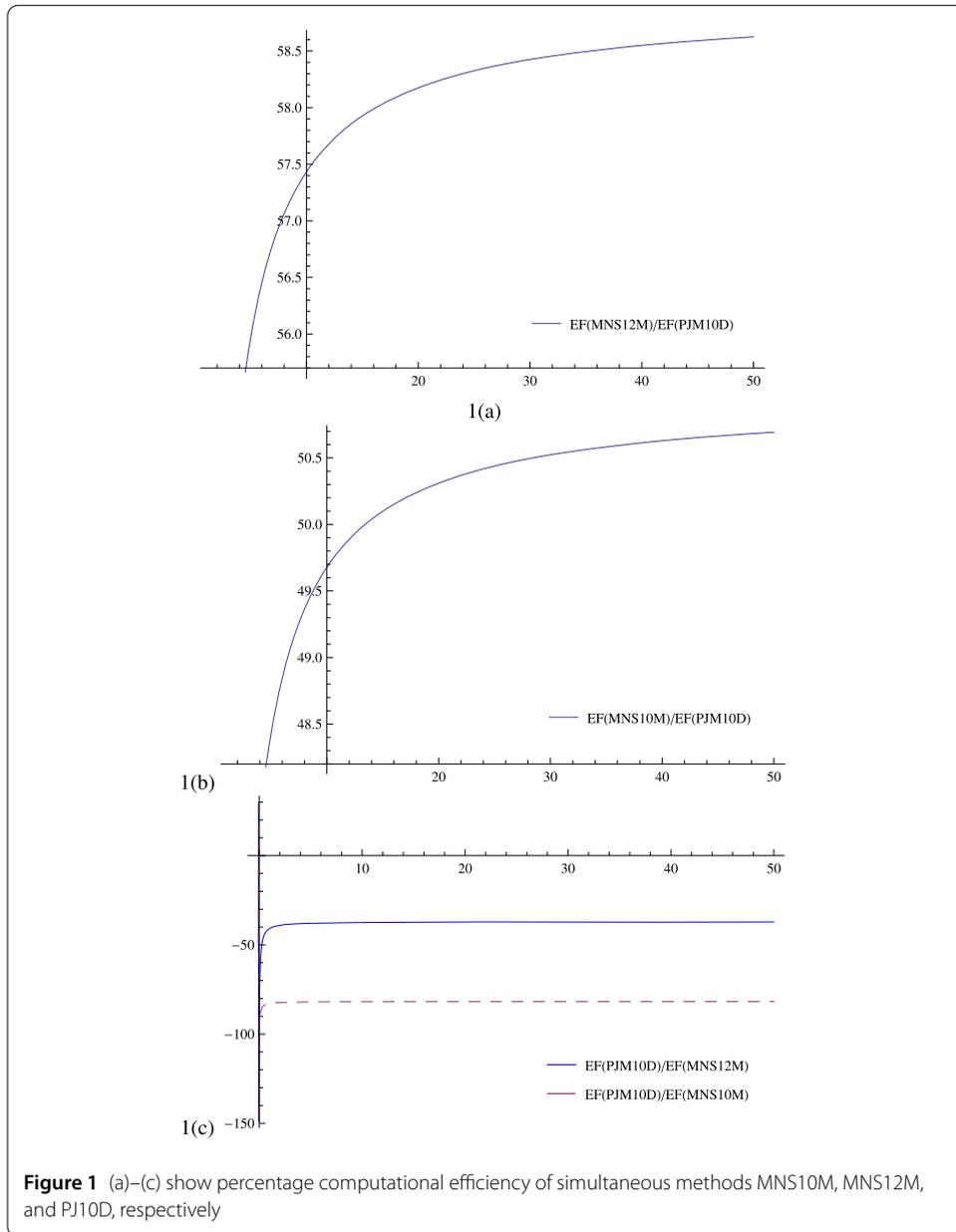
$$\rho((12), (X)) = \left( \frac{EF(12)}{EF(X)} - 1 \right) \times 100 \quad (\text{in percent}), \tag{19}$$

where  $X$  is the Petkovic method PJM10D. These ratios are graphically displayed in Fig. 1(a), (b), (c). It is evident from Fig. 1(a), (b), (c) that the new methods (8) and (12) are more efficient as compared to the Petkovic method PJM10D.

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 that CPU times of the methods MMS10M and MNS12M are less than those of PJM10D, showing the dominant efficiency of our methods (8) and (12) as compared to them.

### 3 Numerical results

Here some numerical examples are considered in order to demonstrate the performance of our family of two-step tenth-order simultaneous methods, namely  $MNS10M$  (8) and  $MNS12M$  (12). We compare our family of methods with the Petkovic et al. [28] method of



convergence of order ten for finding all distinct roots of (1) (abbreviated as PJM10D). All the computations are performed using *Maple 15* with 64 digits floating point arithmetic. We take  $\epsilon = 10^{-30}$  as a tolerance and use the following stopping criteria for estimating the roots:

$$(i) \quad e_i = |f(x_i^{(k+1)})| < \epsilon,$$

where  $e_i$  represents the absolute error of function values in (i)

Numerical test examples from [10, 28, 29] are provided in Tables 2, 3, and 4. In all tables, CO represents the convergence order, n represents the number of iterations, and CPU represents execution time in seconds. All calculations are done using Maple 15 on (Processor Intel(R) Core(TM) i3-3110m CPU@2.4 GHz with 4 GB (3.89 GB USABLE)) with

**Table 2** Residual errors of simultaneous methods PJM10D, MNS10D and MNS12D for finding all the distinct roots of polynomial equation used in Example 1

Method	CO	n	CPU	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$	$e_9$	$e_{10}$	$e_{11}$	$e_{12}$
PJM10D	10	3	1.234	1.7e-4	8.6e-5	3.4e-5	2.5e-4	1.2e-3	2.4e-4	3.5e-4	1.2e-5	113.9e-0	209.0e-0	8.8e-4	2.8e-4
MNS10D	10	3	0.656	1.8e-36	5.8e-39	5.0e-41	1.6e-22	1.1e-21	7.5e-45	8.8e-30	7.3e-46	3.5e-4	1.6e-3	1.9e-33	3.6e-59
MNS12D	12	3	0.563	1.4e-60	2.2e-62	0.0	0.0	3.0e-63	1.0e-62	2.2e-62	0.0	9.2e-62	1.2e-58	1.0e-59	1.4e-59

**Table 3** Residual errors of simultaneous methods PJM10D, MNS10D, MNS10M, MNS12D and MNS12M for finding all the distinct as well as multiple roots of polynomial equation used in Example 2

Method	CO	n	CPU	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$
PJM10D	10	2	0.594	1.2e-1	3.0e-2	2.7e-1	7.9e-2	1.8e-1	3.1e-2	1.2e-1	8.2e-3
MNS10D	10	2	0.25	5.6e-62	3.9e-59	0.0	5.3e-58	1.2e-62	2.0e-58	1.8e-63	4.2e-59
MNS10M	10	2	0.65	9.1e-115	3.9e-174	0.0	1.0e-113	1.4e-120	1.7e-114	6.7e-193	10e-113
MNS12D	12	2	0.343	4.2e-64	1.2e-65	0.0	0.0	1.3e-71	1.7e-64	3.1e-67	0.0
MNS12M	12	2	0.282	1.1e-113	5.0e-177	0.0	2.3e-121	6.7e-125	6.6e-114	6.5e-197	5.4e-121

**Table 4** Residual errors of simultaneous methods PJM10D, MNS10D, MNS10M, MNS12D and MNS12M for finding all the distinct as well as multiple roots of nonlinear equation used in Example 3

Method	CO	n	CPU	$e_1$	$e_2$	$e_3$	$e_4$
PJM10D	10	2	0.125	9.3e-3	2.6e-4	1.2e-3	9.3e-3
MNS10D	10	2	0.079	1.0e-9	0.0	0.0	1.0e-9
MNS10M	10	2	0.068	0.0	0.0	0.0	0.0
MNS12D	12	2	0.078	1.0e-9	0.0	0.0	1.0e-9
MNS12M	12	2	0.093	0.0	0.0	0.0	0.0

64-bit operating system. For multiplicity unity in MNS10M and MNS12M, we get the numerical results for distinct roots, i.e., MNS10D and MNS12D respectively. We observed that numerical results of the methods MNS10D, MNS10M, MNS12D, and MNS12M are comparable with those of the PJM10D method but have a lower number of iterations.

**Example 1** Consider

$$f(x) = x^{12} - (2 + 5i)x^{11} - (1 - 10i)x^{10} + (12 - 25i)x^9 - 30x^8 - x^4 + (2 + 5i)x^3 + (1 - 10i)x^2 - (12 - 25i)x + 30,$$

with exact roots

$$\zeta_{1,2} = \pm 1, \quad \zeta_{3,4} = \pm i, \quad \zeta_{5,6} = \frac{\sqrt{2}}{2} \pm \frac{\sqrt{2}i}{2}, \quad \zeta_{7,8} = -\frac{\sqrt{2}}{2} \pm \frac{\sqrt{2}i}{2}, \quad \zeta_9 = 2i, \\ \zeta_{10} = 3i, \quad \zeta_{11,12} = 1 \pm 2i.$$

The initial approximations have been taken as

$$\begin{aligned} x_1^{(0)} &= 1.3 + 0.2i, & x_2^{(0)} &= -1.3 + 0.2i, & x_3^{(0)} &= -0.3 - 1.2i, & x_4^{(0)} &= -0.3 + 1.2i, \\ x_5^{(0)} &= 0.5 + 0.5i, & x_6^{(0)} &= 0.5 - 0.5i, & x_7^{(0)} &= -0.5 + 0.5i, & x_8^{(0)} &= -0.5 - 0.5i, \\ x_9^{(0)} &= -0.2 + 2.2i, & x_{10}^{(0)} &= 0.2 + 2.3i, & x_{11}^{(0)} &= 1.3 + 2.2i, & x_{12}^{(0)} &= 1.3 - 2.2i. \end{aligned}$$

**Example 2** Consider

$$f(x) = (x+1)^2(x+2)^3(x^2-2x+2)^2(x^2+1)^2(x-2)^3(x+2-i)^2,$$

with exact roots

$$\zeta_1 = -1, \quad \zeta_2 = -2, \quad \zeta_{3,4} = 1 \pm i, \quad \zeta_{5,6} = \pm i, \quad \zeta_7 = 2, \quad \zeta_8 = -2 + i.$$

The initial approximations have been taken as

$$\begin{aligned} x_1^{(0)} &= -1.3 + 0.2i, & x_2^{(0)} &= -2.2 - 0.3i, & x_3^{(0)} &= 1.3 + 1.2i, & x_4^{(0)} &= 0.7 - 1.2i, \\ x_5^{(0)} &= -0.2 + 0.8i, & x_6^{(0)} &= 0.2 - 1.3i, & x_7^{(0)} &= 2.2 - 0.3i, & x_8^{(0)} &= -2.2 + 0.7i. \end{aligned}$$

**Example 3** Consider

$$f(x) = (e^{x(x-1)(x-2)(x-3)} - 1)^4,$$

with exact roots

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

The initial approximations have been taken as

$$x_1^{(0)} = 0.1, \quad x_2^{(0)} = 0.9, \quad x_3^{(0)} = 1.8, \quad x_4^{(0)} = 2.9,$$

### 3.1 Results and discussion

From Tables 2–4 and from Fig. 1(a)–(c), we conclude that

- Our methods MNS10D and MNS12D are more efficient as compared to PJM10D in terms of the number of iterations and CPU time.
- Our methods MNS10M and MNS12M are applicable for multiple as well as distinct roots, whereas PJM10D is applicable for distinct roots only.

## 4 Conclusion

We have developed here two simultaneous two-step methods of order ten and twelve, namely MNS10D, MNS10M, MNS12D, and MNS12M for determination of all the distinct as well as multiple roots of nonlinear polynomial equation (1). From Tables 1–4, we observed that our methods are very effective and more efficient as compared to the existing method PJM10D [28].

### Funding

The authors declare that there is no funding available for this paper.

### Availability of data and materials

Not applicable.

## Declarations

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

All authors read and approved the final manuscript.

### Author details

<sup>1</sup>Department of Mathematics and Statistics, Riphah International University I-14, Islamabad, 44000, Pakistan.

<sup>2</sup>Department of Mathematics, NUML, Islamabad, Pakistan. <sup>3</sup>Department of Mathematics, Faculty of Arts and Sciences, Yildiz Technical University, Esenler, 34210, Istanbul, Turkey. <sup>4</sup>Department of Mathematics, Anand International College of Engineering, Jaipur, 303012, Rajasthan, India. <sup>5</sup>Nonlinear Dynamics Research Center (NDRC), Ajman University, Ajman, UAE. <sup>6</sup>Department of Mathematics, Research Institute of Natural Science, Hanyang University, Seoul, South Korea.

<sup>7</sup>College of Humanities and Sciences, Ajman University, Ajman, UAE.

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 25 August 2021 Accepted: 28 October 2021 Published online: 15 November 2021

### References

1. Cosnard, M., Fraigniaud, P.: Finding the roots of a polynomial on an MIMD multicomputer. *Parallel Comput.* **15**, 75–85 (1990)
2. Kanno, S., Kjurkchiev, N., Yamamoto, T.: On some methods for the simultaneous determination of polynomial zeros. *Jpn. J. Appl. Math.* **13**, 267–288 (1995)
3. Proinov, P.D., Cholakov, S.I.: Semilocal convergence of Chebyshev-like root-finding method for simultaneous approximation of polynomial zeros. *Appl. Math. Comput.* **236**, 669–682 (2014)
4. Proinov, P.D., Vasileva, M.T.: On the convergence of family of Weierstrass-type root-finding methods. *C. R. Acad. Bulg. Sci.* **68**, 697–704 (2015)
5. Sendov, B., Andreev, A., Kjurkchiev, N.: *Numerical Solutions of Polynomial Equations*. Elsevier, New York (1994)
6. Wang, X., Liu, L.: Modified Ostrowski's method with eight-order convergence and high efficiency index. *Appl. Math. Lett.* **23**, 549–554 (2010)
7. Li, T.F., Li, D.S., Xu, Z.D., Fang, Y.L.: New iterative methods for non-linear equations. *Appl. Math. Comput.* **197**, 755–759 (2008)
8. Shams, M., Rafiq, N., Ahmad, B., Mir, N.A.: Inverse numerical iterative technique for finding all roots of nonlinear equations with engineering applications. *J. Math.* **2021**, Article ID 6643514 (2021)
9. Proinov, P.D., Vasileva, M.T.: On the convergence of high-order Ehrlich-type iterative methods for approximating all zeros of polynomial simultaneously. *J. Inequal. Appl.* **2015**, Article ID 336 (2015)
10. Mir, N.A., Muneer, R., Jabeen, I.: Some families of two-step simultaneous methods for determining zeros of non-linear equations. *ISRN Appl. Math.* **2011**, Article ID 817174 (2011)
11. Aberth, O.: Iteration methods for finding all zeros of a polynomial simultaneously. *Math. Comput.* **27**, 339–344 (1973)
12. Nourein, A.W.M.: An improvement on two iteration methods for simultaneously determination of the zeros of a polynomial. *Int. J. Comput. Math.* **6**, 241–252 (1977)
13. Proinov, P.D., Vasileva, M.T.: On the convergence of high-order Gargantini–Farmer–Loizou type iterative methods for simultaneous approximation of polynomial zeros. *Appl. Math. Comput.* **361**, 202–214 (2019)
14. Agarwal, P., Filali, D., Akram, M., Dilshad, M.: Convergence analysis of a three-step iterative algorithm for generalized set-valued mixed-ordered variational inclusion problem. *Symmetry* **13**, Article ID 444 (2021)
15. Sunarto, A., Agarwal, P., Sulaiman, J., Chew, J.V.L., Aruchunan, E.: Iterative method for solving one-dimensional fractional mathematical physics model via quarter-sweep and PAOR. *Adv. Differ. Equ.* **2021**, Article ID 147 (2021)
16. Attary, M., Agarwal, P.: On developing an optimal Jarratt-like class for solving nonlinear equations. *Ital. J. Pure Appl. Math.* **43**, 523–530 (2020)
17. Kumar, S., Kumar, D., Sharma, J.R., Cesarano, C., Agarwal, P., Chu, Y.M.: An optimal fourth order derivative-free numerical algorithm for multiple roots. *Symmetry* **12**, Article ID 1038 (2020)
18. Agarwal, P., Agarwal, R.P., Ruzhansky, M.: *Special Functions and Analysis of Differential Equations*. Chapman and Hall/CRC, London (2020)
19. El-Sayed, A.A., Agarwal, P.: Numerical solution of multiterm variable-order fractional differential equations via shifted Legendre polynomials. *Math. Methods Appl. Sci.* **42**, 3978–3991 (2019)
20. Shah, N.A., Agarwal, P., Chung, J.D., El-Zahar, E.R., Hamed, Y.S.: Analysis of optical solitons for nonlinear Schrodinger equation with detuning term by iterative transform method. *Symmetry* **12**, Article ID 1850 (2021)
21. Ugur, D., Mehmet, A., Serkan, A.: On some polynomials derived from  $(p, q)$ -calculus. *J. Comput. Theor. Nanosci.* **13**, 7903–7908 (2016)
22. Agarwal, P., El-Sayed, A.A.: Vieta–Lucas polynomials for solving a fractional-order mathematical physics model. *Adv. Differ. Equ.* **2020**, Article ID 626 (2020)
23. Agarwal, P., Dragomir, S.S., Jleli, M., Samet, M.B.: *Advances in Mathematical Inequalities and Applications*. Birkhäuser, Basel (2018)
24. Ruzhansky, M., Cho, Y.J., Agarwal, P., Area, I.: *Advances in Real and Complex Analysis with Applications*. Birkhäuser, New York (2017)
25. Agarwal, P., Singh, A., Kilicman, A.: Development of key-dependent dynamic  $S$ -boxes with dynamic irreducible polynomial and affine constant. *Adv. Mech. Eng.* **10**(7), 1–17 (2018)
26. Wu, Z., Li, X.: A fourth-order modification of Newton's method for multiple roots. *Int. J. Res. Rev. Appl. Sci.* **10**, 166–170 (2012)
27. Dong, C.: A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation. *Math. Numer. Sin.* **11**, 445–450 (1982)
28. Petkovic, M.S., Petkovic, L.D., Džunic, J.: On an efficient simultaneous method for finding polynomial zeros. *Appl. Math. Lett.* **28**, 60–65 (2014)
29. Farmer, M.R.: *Computing the Zeros of Polynomials using the Divide and Conquer Approach*. Ph.D. Thesis, University of London (2014)