Applied Mathematics Letters 145 (2023) 108738

Contents lists available at ScienceDirect

Applied Mathematics Letters

www.elsevier.com/locate/aml

An iterative scheme to obtain multiple solutions simultaneously

Alicia Cordero[1], Neus Garrido[1], Juan R. Torregrosa[1,*], Paula Triguero-Navarro[1]

Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, València, Spain

ARTICLE INFO

Article history:
Received 24 April 2023
Accepted 31 May 2023
Available online 2 June 2023

Keywords:
Nonlinear equations
Iterative methods
Simultaneous solutions
Multiple roots
Simultaneous multiple roots

ABSTRACT

In this manuscript, we propose an iterative step that, combined with any other method, allows us to obtain an iterative scheme for approximating the simple roots of a polynomial simultaneously. We show that adding this step, the order of convergence of the new scheme is tripled respect to the original one. With this idea, we also present an iterative method that obtains multiple solutions of any nonlinear equation simultaneously, without the need to know the multiplicity of the solutions. We conclude with several numerical experiments to confirm the behaviour of the proposed methods.

1. Introduction

Solving nonlinear equations $f(x) = 0$ is a old problem of science and engineering. To solve this type of equation is, in general, neither easy nor possible. So, iterative methods are used for approximating their solutions. These schemes can be classified, among others, in to single and simultaneous root finding methods. Numerous authors have studied the problem of simultaneously obtaining all the roots of polynomial equations, designing iterative schemes with or without derivatives. See for example [1–6] and the references therein.

In the paper [7], the authors presented an iterative step to obtain roots simultaneously whatever the type of nonlinear equation. From an iterative method of order p, we can combine that method, used as a predictor, with the mentioned iterative step, to obtain a scheme that obtains roots simultaneously with order $2p$. In this manuscript, we are going to modify step suggested in [7] to increase the order to $3p$ when dealing with polynomial equations. In both cases, these methods assume that the solutions have multiplicity one.

For this reason, we also focus in this article on proposing a method that obtains roots simultaneously when they are not simple, but without the need to know the multiplicity, that is, we do not employ the multiplicity in the iterative expression of the method.

* Corresponding author.
E-mail address: jrtorre@mat.upv.es (J.R. Torregrosa).
[1] All the authors of this paper have contributed equally.

https://doi.org/10.1016/j.aml.2023.108738

In the manuscript [8], the authors proposed a method with quadratic order of convergence, and that obtains multiple roots without knowing this multiplicity. We add to this method the iterative step proposed in this manuscript, obtaining a method of order 4 for any type of equations and of order 6 for polynomial equations, which, as we have already mentioned, obtains multiple roots simultaneously.

The manuscript is structured as follows. In Section 2, the order of convergence of the proposed method for simple roots is presented and analysed. In Section 3, the order of convergence of the method for simultaneous multiple solutions (2) is studied. In Section 4, several numerical experiments are carried out to see the behaviour of the proposed iterative methods, and the paper concludes with Section 5, where conclusions are drawn from the work done.

2. Convergence analysis

Let us denote by ϕ an iterative scheme having order of convergence p. We can define ϕ_s, using ϕ as a predictor, as follows:

$$\begin{cases} y_i^{(k+1)} = \phi(x_i^{(k)}), \\ x_i^{(k+1)} = y_i^{(k+1)} - \dfrac{f\left(y_i^{(k+1)}\right)}{f'\left(y_i^{(k+1)}\right) - f\left(y_i^{(k+1)}\right)\sum_{j\neq i} \dfrac{1}{y_i^{(k+1)} - y_j^{(k+1)}}}, \quad i = 1, 2, \ldots, n \end{cases} \tag{1}$$

Similar to what happens in [7], the order of convergence of the above method is $2p$ when we solve nonlinear equations. Next, we are going to prove that method (1) has order of convergence $3p$ for polynomial equations.

Theorem 1. *Let us define a polynomial function of degree n, $f : D \subseteq \mathbb{C} \longrightarrow \mathbb{C}$ in a neighbourhood D of α_i, such that $f(\alpha_i) = 0$ for $i = 1, \ldots, n$. Let us suppose that $f'(\alpha_i) \neq 0$ is satisfied for $i = 1, \ldots, n$. If ϕ is an iterative method with order p, then, taking an initial estimation $x^{(0)}$ close enough to $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, sequence $\{x^{(k)}\}$ generated by method (1) converges to α with order $3p$.*

Proof. Let us denote by $e_{i,k} = x_i^{(k)} - \alpha_i$, the error of ith component of iterate $x^{(k)}$ and by $e_{y,i,k} = y_i^{(k)} - \alpha_i$, the error of ith component of iterate $y^{(k)}$. We can assume $e_{y,i,k+1} \sim e_{i,k}^p$ since the order of convergence of iterative scheme ϕ is p. If we have the iterates $y_i^{(k)}$ close to α_i, for $i = 1, \ldots, n$, we can approximate $f(x)$ and $f'(x)$ by

$$f(y_i^{(k)}) \approx \prod_{j=1}^{n} \left(y_i^{(k)} - \alpha_j\right), \quad f'(y_i^{(k)}) \approx \sum_{r=1}^{n} \prod_{j=1, j\neq r}^{n} \left(y_i^{(k)} - \alpha_j\right).$$

Thus, we obtain $\dfrac{f'(y_i^{(k)})}{f(y_i^{(k)})} \approx \dfrac{\sum_{r=1}^{n} \prod_{j=1,j\neq r}^{n} \left(y_i^{(k)} - \alpha_j\right)}{\prod_{j=1}^{n} \left(y_i^{(k)} - \alpha_j\right)} = \sum_{j=1}^{n} \dfrac{1}{y_i^{(k)} - \alpha_j}$. Therefore,

$$\begin{aligned} \frac{f'(y_i^{(k)})}{f(y_i^{(k)})} - \sum_{j=1, j\neq i}^{n} \frac{1}{y_i^{(k)} - y_j^{(k)}} &\approx \sum_{j=1}^{n} \frac{1}{y_i^{(k)} - \alpha_j} - \sum_{j=1, j\neq i}^{n} \frac{1}{y_i^{(k)} - y_j^{(k)}} \\ &\approx \frac{1}{y_i^{(k)} - \alpha_i} + \sum_{j=1, j\neq i}^{n} \left(\frac{1}{y_i^{(k)} - \alpha_j} - \frac{1}{y_i^{(k)} - y_j^{(k)}}\right) \\ &\approx \frac{1}{y_i^{(k)} - \alpha_i} + \sum_{j=1, j\neq i}^{n} \frac{\alpha_j - y_j^{(k)}}{\left(y_i^{(k)} - \alpha_j\right)\left(y_i^{(k)} - y_j^{(k)}\right)}. \end{aligned}$$

Since method $y_j^{(k)} = \phi(x_j^{(k)})$ has order of convergence p, this means that $y_k^{(k)}$ satisfies $y_j^{(k)} - \alpha_j = M_{j,k}e_{j,k}^p + O\left(e_{j,k}^{p+1}\right)$, where $M_{j,k}$ is a constant, for $j = 1, \dots, n$. If we replace this error in the above, we obtain

$$\frac{f'(y_i^{(k)})}{f(y_i^{(k)})} - \sum_{j=1, j\neq i}^{n} \frac{1}{y_i^{(k)} - y_j^{(k)}} \approx \frac{1}{y_i^{(k)} - \alpha_i} + \sum_{j=1, j\neq i}^{n} \frac{\alpha_j - y_j^{(k)}}{\left(y_i^{(k)} - \alpha_j\right)\left(y_i^{(k)} - y_j^{(k)}\right)} \approx \frac{1}{e_{y,i,k+1}}$$
$$+ \sum_{j=1, j\neq i}^{n} \frac{M_{j,k}e_{j,k}^p + O\left(e_{j,k}^{p+1}\right)}{\left(y_i^{(k)} - \alpha_j\right)\left(y_i^{(k)} - y_j^{(k)}\right)}.$$

If we denote by $E_{j,k} = \left(y_i^{(k)} - \alpha_j\right)\left(y_i^{(k)} - y_j^{(k)}\right)$, then $\frac{f'(y_i^{(k)})}{f(y_i^{(k)})} - \sum_{j=1, j\neq i}^{n} \frac{1}{y_i^{(k)} - y_j^{(k)}} \approx \frac{1}{e_{y,i,k+1}} + \sum_{j=1, j\neq i}^{n} \frac{M_{j,k}e_{j,k}^p + O\left(e_{j,k}^{p+1}\right)}{E_{j,k}}$. To simplify notation, we denote $R_i(y^{(k)}) = \sum_{j=1, j\neq i}^{n} \frac{M_{j,k}e_{j,k}^p + O\left(e_{j,k}^{p+1}\right)}{E_{j,k}}$. Thus, the error equation can be expressed as

$$\begin{aligned}
x_i^{(k+1)} - \alpha_i &= y_i^{(k)} - \alpha_i - \cfrac{1}{\cfrac{f'(y_i^{(k)})}{f(y_i^{(k)})} - \sum_{j=1, j\neq i}^{n} \cfrac{1}{y_i^{(k)} - y_j^{(k)}}} \\
&= e_{y,i,k+1} - \cfrac{1}{\cfrac{1}{e_{y,i,k+1}} + R_i(y^{(k)})} = \frac{e_{y,i,k+1}\left(1 + e_{y,i,k+1}R_i(y^{(k)})\right) - e_{y,i,k+1}}{1 + M_{i,k}e_{i,k}^p R_i(y^{(k)})} \\
&= \frac{e_{y,i,k+1}^2 R_i(y^{(k)})}{1 + M_{i,k}e_{i,k}^p R_i(y^{(k)})}.
\end{aligned}$$

By applying that $e_{y,i,k+1} = M_{i,k}e_{i,k}^p + O\left(e_{i,k}^{p+1}\right)$, we have

$$x_i^{(k+1)} - \alpha_i = \frac{\left(e_{i,k}^{2p} + O\left(e_{i,k}^{2p+1}\right)\right) R_i(y^{(k)})}{1 + M_{i,k}e_{i,k}^p R_i(y^{(k)})} = \frac{e_{i,k}^{2p} R_i(y^{(k)}) + O_{3p+1}\left(e_k\right)}{1 + M_{i,k}e_{i,k}^p R_i(y^{(k)})},$$

where $O_{3p+1}\left(e_k\right)$ denotes terms where the sum of the orders of the error product of e_k is at least $3p+1$, as the order of $R_i(y^{(k)})$ is p. Then,

$$x_i^{(k+1)} - \alpha_i \sim e_{i,k}^{2p} R_i(y^{(k)}) \sim e_{i,k}^{2p} \sum_{j=1, j\neq i}^{n} \frac{M_{j,k}e_{j,k}^p}{E_{j,k}} \sim e_{i,k}^{2p} \sum_{j=1, j\neq i}^{n} e_{j,k}^p.$$

The previous relation shows that the order of convergence of ϕ_s is $3p$ when $f(x) = 0$ is a polynomial equation. $\square$

3. Solving multiple solutions simultaneously

In the previous cases, we have assumed that we only have solutions with multiplicity one, but what happens if we have different multiplicities? For this reason, what we do is to combine the method proposed in [8], KM method, with the iterative step defined in this manuscript, in order to find as many solutions as we wish, and are possible, regardless of whether they have multiplicity 1 or higher.

The proposed method, which we denote by KM_s, has the following iterative expression, where $g(x) = \frac{f(x)}{f'(x)}$,

$$\begin{cases} y_i^{(k)} = x_i^{(k)} - \dfrac{g(x_i^{(k)})}{g[2x_i^{(k)} - x_i^{(k-1)}, x_i^{(k-1)}]}, \quad i = 1, \dots, n \\ x_i^{(k+1)} = y_i^{(k)} - \dfrac{g(y_i^{(k)})}{g'(y_i^{(k)}) - g(y_i^{(k)}) \sum_{j=1, j\neq i}^{n} \dfrac{1}{y_i^{(k)} - y_j^{(k)}}}, \quad i, j = 1, \dots, n. \end{cases} \tag{2}$$

This scheme has order of convergence 4, for nonlinear equations, and order 6 for polynomial equations.

Theorem 2. *Let us consider a sufficiently differentiable function $f : D \subseteq \mathbb{C} \longrightarrow \mathbb{C}$ in a neighbourhood D of α_i for $i = 1, \dots, n$, such that $f(\alpha_i) = 0$ for $i = 1, \dots, n$ with unknown multiplicity $m_i \in \mathbb{N} - \{1\}$, for $i = 1, \dots, n$. Taking an initial estimation $x^{(0)}$ close enough to $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$, then method KM_s converges to α with order 4.*

Proof. We denote $C_{i,j} = \dfrac{m_i!}{(m_i + j)!} \dfrac{f^{(m_i+j)}(\alpha_i)}{f^{(m_i)}(\alpha_i)}$ for $j = 1, 2, \dots$ and $i = 1, 2, \dots, n$.

On the one hand, in [8] we proved that the first step has order 2, that is,

$$e_{i,y,k} \sim \left(\frac{-1}{m_i} C_{i,1} e_{i,k}^2 + \frac{(m_i + 1)C_{i,1}^2 - 2m_i C_{i,2}}{m_i^2} (-5e_{i,k}^3 + 2e_{i,k}^2 e_{i,k-1} - e_{i,k} e_{i,k-1}^2) \right).$$

We first obtain the Taylor development of $f(y_i^{(k)})$ around α_i where $e_{i,y,k} = y_i^{(k)} - \alpha_i$:

$$f(y_i^{(k)}) = \frac{f^{(m_i)}(\alpha_i)}{m_i!} \left(e_{i,y,k}^{m_i} + C_{i,1} e_{i,y,k}^{m_i+1} + C_{i,2} e_{i,y,k}^{m_i+2} \right) + O(e_{i,y,k}^{m_i+3}).$$

The derivative of the Taylor development gives that

$$f'(y_i^{(k)}) = \frac{f^{(m_i)}(\alpha)}{m_i!} \left(m_i e_{i,y,k}^{m_i-1} + (m_i + 1) C_{i,1} e_{i,y,k}^{m_i} + (m_i + 2) C_{i,2} e_{i,y,k}^{m_i+1} \right) + O(e_{i,y,k}^{m_i+2}).$$

Then,

$$g(y_i^{(k)}) = \frac{f(y_i^{(k)})}{f'(y_i^{(k)})} = \frac{1}{m_i} \left(e_{i,y,k} - \frac{1}{m_i} C_{i,1} e_{i,y,k}^2 + \frac{(m_i + 1)C_{i,1}^2 - 2m_i C_{i,2}}{m_i^2} e_{i,y,k}^3 \right) + O(e_{i,y,k}^4).$$

We derive the last expression to obtain $g'(y_i^{(k)})$

$$g'(y_i^{(k)}) = \frac{1}{m_i} \left(1 - \frac{2}{m_i} C_{i,1} e_{i,y,k} + 3 \frac{(m_i + 1)C_{i,1}^2 - 2m_i C_{i,2}}{m_i^2} e_{i,y,k}^2 \right) + O(e_{i,y,k}^3).$$

On the other hand,

$$y_i^{(k)} - y_j^{(k)} = y_i^{(k)} - \alpha_i + \alpha_i - y_j^{(k)} + \alpha_j - \alpha_j = e_{i,y,k} - e_{j,y,k} + \alpha_i - \alpha_j.$$

Moreover, $\sum_{j=1, j\neq i}^{n} \dfrac{1}{y_i^{(k)} - y_j^{(k)}} = \sum_{j=1, j\neq i}^{n} \dfrac{1}{e_{i,y,k} - e_{j,y,k} + \alpha_i - \alpha_j}$. By denoting $S_i(y^{(k)}) = \sum_{j=1, j\neq i}^{n} \dfrac{1}{e_{i,y,k} - e_{j,y,k} + \alpha_i - \alpha_j}$, we have

$$g'(y_i^{(k)}) - g(y_i^{(k)}) \sum_{j=1, j\neq i}^{n} \frac{1}{y_i^{(k)} - y_j^{(k)}} = \frac{1}{m_i} \left(1 - \frac{2}{m_i} C_{i,1} e_{i,y,k} + 3 \frac{(m_i + 1)C_{i,1}^2 - 2m_i C_{i,2}}{m_i^2} e_{i,y,k}^2 \right)$$

$$-\frac{1}{m_i}\left(e_{i,k,j}-\frac{1}{m_i}C_{i,1}e_{i,y,k}^2\right)S_i(y^{(k)})$$
$$=\frac{1}{m_i}-\frac{1}{m_i}\left(\frac{2}{m_i}C_{i,1}+S_i(y^{(k)})\right)e_{i,y,k}+O_2(e_{i,y,k}).$$

Thus,

$$\frac{g(y_i^{(k)})}{g'(y_i^{(k)})-g(y_i^{(k)})\sum_{j=1,j\neq i}^{n}\dfrac{1}{y_i^{(k)}-y_j^{(k)}}}=\frac{m_i g(y_i^{(k)})}{1-\left(\frac{2}{m_i}C_{i,1}+S_i(y^{(k)})\right)e_{i,y,k}+O_2(e_{i,y,k})}.$$

Therefore,

$$\begin{aligned} e_{i,k+1} &= e_{i,y,k}-\frac{g(y_i^{(k)})}{g'(y_i^{(k)})-g(y_i^{(k)})\sum_{j=1,j\neq i}^{n}\dfrac{1}{y_i^{(k)}-y_j^{(k)}}}\\ &= e_{i,y,k}-\frac{m_i g(y_i^{(k)})}{1-\left(\frac{2}{m_i}C_{i,1}+S_i(y^{(k)})\right)e_{i,y,k}+O_2(e_{i,y,k})}\\ &= \frac{e_{i,y,k}-\left(\frac{2}{m_i}C_{i,1}+S_i(y^{(k)})\right)e_{i,y,k}^2-\left(e_{i,y,k}-\frac{1}{m_i}C_{i,1}e_{i,y,k}^2\right)+O_3(e_{i,y,k})}{1-\left(\frac{2}{m_i}C_{i,1}+S_i(y^{(k)})\right)e_{i,y,k}+O_2(e_{i,y,k})}\\ &= \frac{-\left(\frac{1}{m_i}C_{i,1}+S_i(y^{(k)})\right)e_{i,y,k}^2+O_3(e_{i,y,k})}{1-\left(\frac{2}{m_i}C_{i,1}+S_i(y^{(k)})\right)e_{i,y,k}+O_2(e_{i,y,k})}, \end{aligned}$$

that is, $e_{i,k+1}\sim e_{i,y,k}^2$. And, since $e_{i,y,k}$ has order of convergence 2, it is proven that method KM_s has order of convergence 4 for nonlinear equations. Similar to Theorem 1, we can prove that the KM_s method has order of convergence 6 for polynomial equations. □

4. Numerical results

In this section, different numerical experiments are carried out in order to analyse the selected methods for solving nonlinear equations. Here, we modify Newton's method (denoted by N), Steffensen's method [9] (denoted by S), the N_4 and N_8 methods designed in [10], and the M_4 and M_6 schemes constructed in [11]. According to the same notation used in the previous section, let us denote by ϕ_s the step-added variant of an iterative scheme ϕ.

The numerical results shown in this section have been performed using Matlab 2020b with variable precision arithmetic with 2000 digits. The iterative process ends when $\|F\left(x^{(k+1)}\right)\|_2$ is less than the chosen tolerance, where $F\left(x^{(k+1)}\right)=\left(f\left(x_1^{(k+1)}\right),f\left(x_2^{(k+1)}\right),\ldots,f\left(x_n^{(k+1)}\right)\right)$, or the maximum of 50 iterations is reached. In the different tables, we shown: $\|F\left(x^{(k+1)}\right)\|_2$, where $x^{(k+1)}$ is the last iteration, $\|x^{(k+1)}-x^{(k)}\|_2$, the number of iterations necessary to satisfy the required tolerance, and the approximated computational order of convergence (ACOC), defined in [12], by:

$$p\approx ACOC=\frac{\ln(\|x^{(k+1)}-x^{(k)}\|_2/\|x^{(k)}-x^{(k-1)}\|_2)}{\ln(\|x^{(k)}-x^{(k-1)}\|_2/\||x^{(k-1)}-x^{(k-2)}\|_2)}.$$

We perform different numerical experiments, both with single and multiple roots.

- $f_1(x)=(x-1)(x+2)(x-5)=0$. Initial estimations $x^{(0)}=(0.5,-1,4)$ and $x^{(-1)}=0.95(0.5,-1,4)$, and tolerance 10^{-200}.

Table 1
Results for the equation $f_1(x) = 0$.

Method	$\|x^{(k+1)} - x^{(k)}\|_2$	$\|F(x^{(k+1)})\|_2$	Iteration	ACOC
N_s	1.5973×10^{-72}	3.2438×10^{-436}	4	6.0624
S_s	2.1948×10^{-178}	2.431×10^{-1066}	8	5.9526
$N_{4,s}$	2.9452×10^{-43}	4.945×10^{-524}	3	12.766
$N_{8,s}$	3.2577×10^{-150}	0	3	25.388
$M_{4,s}$	3.971×10^{-171}	0	4	12.449
$M_{6,s}$	1.1206×10^{-11}	6.2692×10^{-207}	3	19.283
KM_s	2.2214×10^{-165}	3.0604×10^{-534}	7	3.2246

Table 2
Results for the equation $f_2(x) = 0$.

Method	$\|x^{(k+1)} - x^{(k)}\|_2$	$\|F(x^{(k+1)})\|_2$	Iteration	ACOC
N_s	1.6047×10^{-7}	2.1743×10^{-26}	24	1.0
S_s	4.7822×10^{-8}	1.5417×10^{-26}	26	1.0
$N_{4,s}$	3.4075×10^{-7}	3.2454×10^{-26}	15	1.0
$N_{8,s}$	3.3765×10^{-7}	6.0992×10^{-27}	12	1.0
$M_{4,s}$	2.1164×10^{-7}	9.1334×10^{-27}	17	1.0
$M_{6,s}$	3.8861×10^{-6}	3.9141×10^{-40}	37	5.0101
KM_s	5.1263×10^{-10}	1.2125×10^{-28}	4	5.6266

- $f_2(x) = (x-1)^4(x-3)^2(x+2) = 0$. Initial estimations $x^{(0)} = (0.8, 3.5, -1.5)$ and $x^{(-1)} = 0.95(0.8, 3.5, -1.5)$.
- $f_3(x) = (e^{x^2-1} - e^{x^3-2x^2-x+2})^2 = 0$. Initial estimations $x^{(0)} = (-1.2, 1.2, 2.8)$ and $x^{(-1)} = 1.05(-1.2, 1.2, 2.8)$.
- $f_4(x) = (x^2-1)^2 = 0$. Initial estimations $x^{(0)} = (-1.5, 1.5)$ and $x^{(-1)} = 0.95(-1.5, 1.5)$. For f_2, f_3 and f_4, we use as tolerance 10^{-25}.

Table 1 shows the data obtained for the equation $f_1(x) = 0$. We can observe that the approximations obtained are similar in all cases. We can observe that the methods that obtain better approximations are those with higher ACOC. Also, we see that in this case Steffensen's method performs more iterations than method KM_s, which is focused on finding solutions with multiplicities greater than 1.

The results for the equation $f_2(x) = 0$ are given in Table 2. For the methods KM_s, S_s and $M_{4,s}$, the vector of obtained solutions is approximately $(1, 3, -2)$, while for the rest of the methods it is approximately $(1, 3, 3)$. Thus, we can observe that the KM_s method converges to more roots in fewer iterations. Moreover, as is shown in the ACOC column, the only method that obtains an approximate computational convergence order that is similar to the theoretical one, is the KM_s method.

Table 3 summarizes the results for solving $f_3(x) = 0$. For the methods different from KM_s, the vector of approximations obtained is approximately $(-1, 1, 1)$, while for method KM_s it is approximately $(-1, 1, 3)$. Thus, we can observe that method KM_s converges to more roots than the rest of the methods, and moreover, it does so in fewer iterations. Moreover, we can see that it is the only method that obtains an ACOC similar to the theoretical order.

The numerical results obtained for $f_4(x) = 0$ are shown in Table 4. For the methods S_s, $M_{4,s}$ and $M_{6,s}$, the vector of approximations obtained is approximately $(1, 1)$, while for the other methods it is approximately $(-1, 1)$. The conclusions for this example are similar to the previous one.

As a conclusion of these numerical experiments, we observe that the method to use when the roots are not simple is the method KM_s since the rest of them do not converge or do not obtain as many roots as possible. Besides, for a tolerance that is not too demanding, they need too many iterations to satisfy the stopping criterion.

Table 3
Results for the equation $f_3(x) = 0$.

Method	$\|x^{(k+1)} - x^{(k)}\|_2$	$\|F(x^{(k+1)})\|_2$	Iteration	ACOC
N_s	9.0686×10^{-14}	5.8438×10^{-26}	21	0.99958
S_s	n.c	n.c	n.c	n.c
$N_{4,s}$	2.2828×10^{-13}	6.6358×10^{-26}	14	0.99165
$N_{8,s}$	1.3799×10^{-13}	4.755×10^{-27}	11	0.97952
$M_{4,s}$	n.c	n.c	n.c	n.c
$M_{6,s}$	n.c	n.c	n.c	n.c
KM_s	4.0863×10^{-12}	2.6753×10^{-33}	4	8.9077

Table 4
Results for the equation $f_4(x) = 0$.

Method	$\|x^{(k+1)} - x^{(k)}\|_2$	$\|F(x^{(k+1)})\|_2$	Iteration	ACOC
N_s	1.6904×10^{-13}	8.9803×10^{-27}	22	1.0
S_s	2.7531×10^{-13}	8.5756×10^{-27}	18	1.0
$N_{4,s}$	1.8699×10^{-13}	2.0183×10^{-27}	15	1.0
$N_{8,s}$	8.317×10^{-13}	8.6956×10^{-27}	11	1.0
$M_{4,s}$	3.2951×10^{-11}	2.6339×10^{-44}	28	2.0108
$M_{6,s}$	1.384×10^{-6}	7.035×10^{-27}	49	2.3229
KM_s	3.1386×10^{-22}	3.9569×10^{-69}	4	4.0326

5. Conclusions

In this manuscript, we have defined an iterative method for polynomial equations from any other iterative method that obtains roots simultaneously, and this new iterative scheme is three times the order of the original iterative method. We have also added the iterative step to a method proposed in [8] in order to obtain a new iterative method that obtains solutions with several multiplicities simultaneously, without the need to use the value of the multiplicity in its iterative expression. This method has order 4 for nonlinear equations, and increases to order 6 for polynomial equations. After the convergence analysis, different numerical experiments are made with the proposed simple root methods based on known methods in order to study the performance of these new iterative schemes, showing that the ACOC is close to the expected order of convergence. We have also performed experiments with the method for multiple solutions simultaneously to observe the behaviour of the method.

Data availability

No data was used for the research described in the article.

Financial disclosure

This research was partially supported by Universitat Politècnica de València Contrato Predoctoral PAID-01-20-17 (UPV), Spain.

References

[1] M. Petković, L. Petković, J. Dzunic, On an efficient simultaneous method for finding polynomial zeros, Appl. Math. Lett. 28 (2014) 60–65, http://dx.doi.org/10.1016/j.aml.2013.09.011.
[2] P. Proinov, M. Vasileva, On the convergence of high-order Ehrlich-type iterative methods for approximating all zeros of polynomial simultaneously, J. Inequal. Appl. 2015:336 (2015) 1–25, http://dx.doi.org/10.1186/s13660-015-0855-5.
[3] P. Proinov, On the local convergence of Ehrlich method for numerical computation of polynomial zeros, Calcolo 53 (2016) 413–426, http://dx.doi.org/10.1007/s10092-015-0155-y.
[4] P. Proinov, S. Ivanov, M. Petković, On the convergence of Gander's type family of iterative methods for simultaneous approximationof polynomial zeros, Appl. Math. Comput. 349 (2019) 168–183, http://dx.doi.org/10.1016/j.amc.2018.12.041.

[5] S. Cholakov, Local and semilocal convergence of Wang-Zheng's method for simultaneous finding polynomial zeros, Symmetry 219: 736 (2019) 1–15, http://dx.doi.org/10.3390/sym11060736.

[6] N. Ahmad Mir, M. Shams, N. Rafiq, S. Akram, M. Rizwan, Derivative free iterative simultaneous method for finding distint roots of polynomial equation, Alex. Eng. J. 59 (2020) 1629–1636, http://dx.doi.org/10.1016/j.aej.2020.04.009.

[7] A. Cordero, N. Garrido, J. Torregrosa, P. Triguero-Navarro, Iterative schemes for finding all roots simultaneously of nonlinear equations, Appl. Math. Lett. 134 (2022) http://dx.doi.org/10.1016/j.aml.2022.108325.

[8] A. Cordero, N. Garrido, J. Torregrosa, P. Triguero-Navarro, Modifying Kurchatov's method to find multiple roots, Appl. Numer. Math. (2023) submitted for publication.

[9] J.F. Steffensen, Remarks on iteration, Skand. Aktuarietidskr. 16 (1933) 64–72, http://dx.doi.org/10.1080/03461238.1933.10419209.

[10] A. Cordero, J.R. Torregrosa, P. Triguero-Navarro, A general optimal iterative scheme with arbitrary order of convergence, Symmetry 13 (5) (2021) http://dx.doi.org/10.3390/sym13050884.

[11] A. Cordero, N. Garrido, J.R. Torregrosa, P. Triguero-Navarro, Memory in the iterative processes for nonlinear problems, Math. Methods Appl. Sci. (2022) 1–14, http://dx.doi.org/10.1002/mma.8746.

[12] A. Cordero, J.R. Torregrosa, Variants of Newton's method using fifth-order quadrature formulas, Appl. Math. Comput. 190 (2007) 686–698, http://dx.doi.org/10.1016/j.amc.2007.01.062.