*Research Article*

# Multipoint Iterative Methods for Finding All the Simple Zeros in an Interval

## T. Lotfi,[1] F. Soleymani,[2] S. Sharifi,[1] S. Shateyi,[3] and F. Khaksar Haghani[2]

[1] *Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran*

[2] *Department of Mathematics, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran*

[3] *Department of Mathematics and Applied Mathematics, School of Mathematical and Natural Sciences, University of Venda, P. Bag X5050, Thohoyandou 0950, South Africa*

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Two new families of multipoint without memory iterative methods with eighth- and sixteenth-orders are constructed using the symbolic software Mathematica. The key idea in constructing such methods is based on producing some generic suitable functions to reduce the functional evaluations and increase the order of convergence along the computational efficiency. Again by applying Mathematica, we design a hybrid algorithm to capture all the simple real solutions of nonlinear equations in an interval. The application of the new schemes in producing fractal pictures is also furnished.

## 1. Introduction

In this work, we are concerned with the numerical solution of nonlinear equations with application. As usual, let the scalar function $f(x)$ be sufficiently differentiable in the open interval $D$ and have a zero. That is, $\alpha$ exists such that $f(\alpha) = 0$. If $f'(\alpha) \neq 0$, we categorize the solution as a simple one, while if $f'(\alpha) = f''(\alpha) = \cdots = f^{(m-1)}(\alpha) = 0$ and $f^{(m)}(\alpha) \neq 0$, then the solution is multiple. The application of such schemes, from finding Moore-Penrose inverse to calculating the $p$-adic inverse of a number to the module $k$, has attracted the researchers for constructing multipoint schemes possessing a much more reliable computational efficiency; see, for example, [1–4].

Basically in the literature (see, e.g., [5, 6]), two important features determine the choice of an iteration method for solving the equation $f(x) = 0$ as follows: first the total number of iterations and second the computational cost. The former is measured by the order of convergence and the latter by the necessary number of evaluations of the function $f$ and its derivatives per full cycle. These two characteristics are linked by the concept of computational efficiency index, defined by Traub in [7] as $\sqrt[n]{p}$ wherein $p$ is the order and

$n$ is the number of functional evaluations per iteration by considering that both function and derivative evaluations have the same computational cost. It should be remarked that, based on the still unproven Kung-Traub conjecture [8], an iterative method without memory with $n$ (functional) evaluation in each iteration has the maximum order of convergence $2^{(n-1)}$ (called optimal order).

The aim of this work is threefold. First is to apply the programming package Mathematica so as to obtain some new multipoint iterative methods of optimal orders eight and sixteen, including three and four steps, respectively.

The second is the application of the new schemes in computer graphics. In fact, each iterative fixed-point type method produces unique basins of attraction and fractal behavior, which results in nice pictures, useful in arts as discussed fully by Kalantari in [9].

The only drawback of such iterative schemes is in the choice of the initial guess (seed) to start the process and guarantee the convergence. In this paper, an algorithm will be constructed to provide a list of simple real initial approximations and then to obtain all the simple real zeros up to any desired tolerance, when high precision computing is needed. This would be the third goal of the present work.

The rest of the paper unfolds the contents as follows. Section 2 gives a generic three-step family of methods, while Section 3 is devoted to propose a new general family of four-step optimal methods along its analysis of convergence. The fractal behaviors for one of the new methods in producing nice self-similar pictures in arts will be given in Section 4. In Section 5, we remind of the approach of Wagon [10] for extracting a list of initial approximations for simple real zeros along some notes and a hybrid algorithm for computing all the simple real zeros. Some numerical examples will also be pointed out therein. Section 6 concludes the paper.

## 2. A New Optimal Eighth-Order Method with Two Generic Functions

At first in this section, we aim at constructing an optimal eighth-order method without memory, which is used as the first three steps in constructing our four-step optimal sixteenth-order family in the next section.

To this end, consider the following three-point method (without the index $n$):

$$
\begin{aligned}
y &= x - \frac{f(x)}{f'(x)}, \\
z &= y - \frac{f(x)}{f(x) - 2f(y)} \frac{f(y)}{f'(x)}, \\
\widehat{x} &= w = z - \frac{f(z)}{f'(z)}.
\end{aligned} \tag{1}
$$

The first two steps are optimal since that is Ostrowski's fourth-order two-step iterative method, since it reaches the order four using three functional evaluations. But, the third step is not good enough. This method performs five functional evaluations per cycle, which does not coincide with Kung and Traub conjecture for iterations without memory. In fact, functional evaluations must be reduced such that it consumes four instead of five.

For this purpose, it is tried to approximate $f'(z)$ in terms of $f(x)$, $f(y)$, $f(z)$, and $f'(x)$; that is

$$
f'(z) \approx \frac{f'(x)}{\mu(t,s) + \rho(u)}, \tag{2}
$$

where $t = f(y)/f(x)$, $s = f(z)/f(x)$, and $u = f(z)/f(y)$. Note that we omit the index $n$ for simplicity only. Substituting (2) into (1), we have

$$
\begin{aligned}
y &= x - \frac{f(x)}{f'(x)}, \\
z &= y - \frac{f(x)}{f(x) - 2f(y)} \frac{f(y)}{f'(x)}, \\
w &= z - (\mu(t,s) + \rho(u)) \frac{f(z)}{f'(x)}.
\end{aligned} \tag{3}
$$

Although the iteration (3) uses four functional evaluations per full cycle, generally it does not have convergence

order eight. It is necessary to find some suitable conditions on the generic functions $\mu(t,s)$ and $\rho(u)$ so that iteration (3) attains an optimal eighth order. The approach of undetermined coefficients, Maclaurin series and Mathematica [11] are our means. For simplicity, we consider the following truncated Maclaurin series:

$$
\mu(t,s) = \mu_{0,0} + \mu_{1,0}t + \mu_{0,1}s + \frac{1}{2!}\mu_{2,0}t^2 + \frac{1}{6}t^3, \tag{4}
$$

where $\mu_{i,j} = (\partial^{i+j}\mu(t,s)/\partial^i t \partial^j s)|_{(0,0)}$ for $i, j = 0, 1, 2, \ldots$, and

$$
\rho(u) = \rho_0 + \rho_1 u + \cdots, \tag{5}
$$

wherein $\rho_i = \rho^{(i)}(0)$ for $i = 0, 1, 2, \ldots$. Note that we defined the truncated Maclaurin series (in the above way purposely) as if to construct the most general iterative methods without memory in the programming package Mathematica.

**Theorem 1.** Let $\alpha \in D$ be a simple zero of the sufficiently differentiable function $f: D \subset R \rightarrow R$ and $c_i = f^{(i)}(\alpha)/(f'(\alpha)i!)$, $i \geq 2$. If $x_0$ is sufficiently close to $\alpha$, then (1) the order of convergence for (3) is eight if

$$
\begin{aligned}
\mu(0,0) &= 0, \qquad \mu_t(0,0) = 2, \qquad \mu_s(0,0) = 4, \\
\mu_{tt}(0,0) &= 10, \qquad \mu_{ttt}(0,0) = 72, \\
\rho(0) &= \rho'(0) = 1.
\end{aligned} \tag{6}
$$

(2) Its error equation reads

$$
e_{n+1} = c_2 \left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2 c_3 + c_2 c_4\right)e_n^8 + O\left(e_n^9\right). \tag{7}
$$

*Proof.* Let $e_n = x_n - \alpha$, $\widetilde{e}_n = y_n - \alpha$, $\widehat{e}_n = z_n - \alpha$, and $e_{n+1} = x_{n+1} - \alpha$. Denote $c_i = (1/i!)(f^{(i)}(\alpha)/f'(\alpha))$, $i \geq 2$. Using Taylor's expansion and taking into account $f(\alpha) = 0$, we have $f(x_n) = f'(\alpha)[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + c_6 e_n^6 + c_7 e_n^7 + c_8 e_n^8 + O(e_n^9)]$, also by differentiation, we obtain $f'(x_n) = f'(\alpha)[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + 6c_6 e_n^5 + 7c_7 e_n^6 + 8c_8 e_n^7 + O(e_n^8)]$. Using the last two equations gives us $f(x_n)/f'(x_n) = e_n - c_2 e_n^2 + 2(c_2^2 - c_3)e_n^3 + (7c_2 c_3 - 4c_2^3 - 3c_4)e_n^4 + (8c_2^4 - 20c_2^2 c_3 + 6c_3^2 + 10c_2 c_4 - 4c_5)e_n^5 + \cdots + O(e_n^9)$. Hence,

$$
\begin{aligned}
\widetilde{e}_n = {} & c_2 e_n^2 + 2\left(-c_2^2 + c_3\right)e_n^3 + \left(-7c_2 c_3 + 4c_2^3 + 3c_4\right)e_n^4 \\
& - \left(8c_2^4 - 20c_2^2 c_3 + 6c_3^2 + 10c_2 c_4 - 4c_5\right)e_n^5 \\
& - \left[-16c_2^5 + 52c_2^3 c_3 - 28c_2^2 c_4 + 17c_3 c_4 \right. \\
& \left. \quad + c_2\left(-33c_3^2 + 13c_5\right) - 5c_6\right]e_n^6 \\
& + \cdots + e_n^8 + O\left(e_n^9\right).
\end{aligned} \tag{8}
$$

Similarly, we get $f(y_n) = f'(\alpha)[c_2 e_n^2 + 2(-c_2^2 + c_3)e_n^3 + (5c_2^3 - 7c_2 c_3 + 3c_4)e_n^4 - 2(6c_2^4 - 12c_2^2 c_3 + 3c_3^2 + 5c_2 c_4 - 2c_5)e_n^5 +$

$(28c_2^5 - 73c_2^3c_3 + 34c_2^2c_4 - 17c_3c_4 + c_2(37c_3^2 - 13c_5) + 5c_6)]e_n^6 + O(e_n^7)$. Now, we have

$$\frac{f(x_n)}{f(x_n) - 2f(y_n)}$$

$$= 1 + 2c_2e_n + \left(-2c_2^2 + 4c_3\right)e_n^2 + \left(-4c_2c_3 + 6c_4\right)e_n^3$$

$$+ \left(4c_2^4 - 6c_2^2c_3 - 4c_2c_4 + 8c_5\right)e_n^4 \tag{9}$$

$$- 2\left(4c_2^5 - 14c_2^3c_3 + 5c_2^2c_4\right.$$

$$\left. -2c_3c_4 + c_2\left(9c_3^2 + 2c_5\right) - 5c_6\right)e_n^5 + O\left(e_n^6\right).$$

Moreover, we obtain

$$\frac{f(y_n)}{f'(x_n)} = c_2e_n^2 + \left(-4c_2^2 + 2c_3\right)e_n^3$$

$$+ \left(13c_2^3 - 14c_2c_3 + 3c_4\right)e_n^4 + \cdots + O\left(e_n^6\right). \tag{10}$$

From (9) and (10), follows

$$\widehat{e_n} = \widetilde{e_n} - \frac{f(x_n)}{f(x_n) - 2f(y_n)} \frac{f(y_n)}{f'(x_n)}$$

$$= \left(c_2^3 - c_2c_3\right)e_n^4 - 2\left(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4\right)e_n^5 \tag{11}$$

$$+ \left(10c_2^5 - 30c_2^3c_3 + 12c_2^2c_4\right.$$

$$\left. -7c_3c_4 + 3c_2\left(6c_3^2 - c_5\right)\right)e_n^6 + O\left(e_n^7\right).$$

Also, $f(z_n) = f(\alpha)[c_2(c_2^2 - c_3)e_n^4 - 2(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4)e_n^5 + (10c_2^5 - 30c_2^3c_3 + 12c_2^2c_4 - 7c_3c_4 + 3c_2(6c_3^2 - c_5))e_n^6] + O(e_n^7)$. We now need $s_n = f(y_n)/f(x_n)$, $t_n = f(z_n)/f(x_n)$, and $u_n = f(z_n)/f(y_n)$. Then,

$$s_n = c_2e_n + \left(-3c_2^2 + 2c_3\right)e_n^2 + \left(8c_2^3 - 10c_2c_3 + 3c_4\right)e_n^3$$

$$+ \left(-20c_2^4 + 37c_2^2c_3 - 8c_3^2 - 14c_2c_4 + 4c_5\right)e_n^4 \tag{12}$$

$$+ \left(48c_2^5 - 118c_2^3c_3 + 51c_2^2c_4 - 22c_3c_4\right.$$

$$\left. +c_2\left(55c_3^2 - 18c_5\right) + 5c_6\right)e_n^5 + O\left(e_n^6\right),$$

$$t_n = \left(c_2^3 - c_2c_3\right)e_n^3 + \left(-5c_2^4 + 9c_2^2c_3 - 2c_3^2 - 2c_2c_4\right)e_n^4$$

$$+ \left(15c_2^5 - 40c_2^3c_3 + 14c_2^2c_4\right. \tag{13}$$

$$\left. -7c_3c_4 + 3c_2\left(7c_3^2 - c_5\right)\right)e_n^5 + O\left(e_n^6\right),$$

$$u_n = \left(c_2^2 - c_3\right)e_n^2 - 2\left(c_2^3 - 2c_2c_3 + c_4\right)e_n^3$$

$$+ \left(c_2^4 - 6c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 3c_5\right)e_n^4 \tag{14}$$

$$+ \left(4c_2^5 - 4c_2^3c_3 - 4c_2^2c_4 + 6c_3c_4\right.$$

$$\left. +c_2\left(-4c_3^2 + 6c_5\right) - 4c_6\right)e_n^5 + O\left(e_n^6\right),$$

while $f(z_n)/f'(x_n) = (c_2^3 - c_2c_3)e_n^4 - 2(3c_2^4 - 5c_2^2c_3 + c_3^2 + c_2c_4)e_n^5 + (22c_2^5 - 53c_2^3c_3 + 16c_2^2c_4 - 7c_3c_4 + c_2(25c_3^2 - 3c_5))e_n^6 + O(e_n^7)$.

It is remarked that although we have not presented higher order terms in the above expressions, one can obtain them simply by the aid of a system of computation such as Mathematica. In the third step of (3), we need to consider Taylor's series of $\mu(t_n, s_n)$ and $\rho(u_n)$ about $(0, 0)$ and $0$, respectively. To our purposes, it suffices to use the following: $\mu(t_n, s_n) = \mu_{0,0} + \mu_{1,0}t + \mu_{0,1}s_n + (1/2)\mu_{2,0}t_n^2 + (1/6)\mu_{3,0}t_n^3$, and $\rho(u_n) = \rho_0 + \rho_1 u_n$.

So, we get the following general error equation if $\alpha$ is subtracted from both sides of (3):

$$e_{n+1} = C_4e_n^4 + C_5e_n^5 + C_6e_n^6 + C_7e_n^7 + C_8e_n^8 + O\left(e_n^9\right), \tag{15}$$

where $C_4 = -c_2(c_2^2 - c_3)(-1 + \rho_0 + \mu_{0,0})$, and

$$C_5 = \left[2c_3^2\left(-1 + \rho_0 + \mu_{0,0}\right) + 2c_2c_4\left(-1 + \rho_0 + \mu_{0,0}\right)\right.$$

$$+ c_2^4\left(-4 + 6\rho_0 + 6\mu_{0,0} - \mu_{1,0}\right)$$

$$\left. + c_2^2c_3\left(8 - 10\rho_0 - 10\mu_{0,0} + \mu_{1,0}\right)\right],$$

$$C_6 = \left[7c_3c_4\left(-1 + \rho_0 + \mu_{0,0}\right)\right.$$

$$- c_2c_3^2\left(-18 + 25\rho_0 + \rho_1 + 25\mu_{0,0} - 4\mu_{1,0}\right)$$

$$+ 2c_2^2c_4\left(6 - 8\rho_0 - 8\mu_{0,0} + \mu_{1,0}\right)$$

$$- \frac{1}{2}c_2^5\left(-20 + 44\rho_0 + 2\rho_1 + 44\mu_{0,0} - 18\mu_{1,0} + \mu_{2,0}\right)$$

$$+ \frac{1}{2}c_2^3c_3\left(106\rho_0 + 4\rho_1 + 106\mu_{0,0}\right.$$

$$\left.\left. -30\left(2 + \mu_{1,0}\right) + \mu_{2,0}\right)\right],$$

$$C_7 = \left[-2\left(-3c_4^2\left(-1 + \rho_0 + \mu_{0,0}\right)\right.\right.$$

$$\left. +c_3^3\left(-6 + 9\rho_0 + \rho_1 + 9\mu_{0,0} - 2\mu_{1,0}\right)\right)$$

$$- 2c_2c_3c_4\left(-26 + 38\rho_0 + 2\rho_1 + 38\mu_{0,0} - 7\mu_{1,0}\right)$$

$$+ c_2^3c_4\left(-40 + 76\rho_0 + 4\rho_1 + 76\mu_{0,0} - 25\mu_{1,0} + \mu_{2,0}\right)$$

$$+ c_2^2c_3^2\left(-80 + 160\rho_0 + 16\rho_1\right.$$

$$\left. +160\mu_{0,0} - \mu_{0,1} - 61\mu_{1,0} + 3\mu_{2,0}\right)$$

$$+ c_2^6\left(-20 + 64\rho_0 + 8\rho_1 + 64\mu_{0,0}\right.$$

$$\left. -\mu_{0,1} - 48\mu_{1,0} + 6\mu_{2,0} - \frac{\mu_{3,0}}{6}\right)$$

$$+ \frac{1}{6}c_2^4c_3\left(-6\left(-80 + 204\rho_0 + 22\rho_1 + 204\mu_{0,0}\right.\right.$$

$$\left.\left.\left. -2\mu_{0,1} - 113\mu_{1,0} + 10\mu_{2,0}\right) + \mu_{3,0}\right)\right],$$

$$
\begin{aligned}
C_8 = \frac{1}{6} \Big[ \Big( &-6c_3^2 c_4 \left(-50 + 79\rho_0 + 11\rho_1 + 79\mu_{0,0} - 20\mu_{1,0}\right) \\
&+ 3c_2^2 c_3 c_4 \left(-418 + 898\rho_0 + 112\rho_1 \right. \\
&\quad\quad\quad \left. + 898\mu_{0,0} - 8\mu_{0,1} - 386\mu_{1,0} + 21\mu_{2,0}\right) \\
&+ 6c_2 \Big( c_4^2 \left(37 - 57\rho_0 - 4\rho_1 - 57\mu_{0,0} + 12\mu_{1,0}\right) \\
&\quad\quad\quad + c_3^3 \left(-91 + 202\rho_0 + 36\rho_1 + 202\mu_{0,0} \right. \\
&\quad\quad\quad\quad\quad \left. - 4\mu_{0,1} - 96\mu_{1,0} + 6\mu_{2,0}\right)\Big) \\
&+ c_2^5 c_3 \Big( 6 \left(-178 + 654\rho_0 + 126\rho_1 + 654\mu_{0,0} \right. \\
&\quad\quad\quad\quad \left. -30\mu_{0,1} - 604\mu_{1,0} + 97\mu_{2,0}\right) - 25\mu_{3,0}\Big) \\
&- 3c_2^7 \Big( -72 + 330\rho_0 + 70\rho_1 + 330\mu_{0,0} \\
&\quad\quad\quad -22\mu_{0,1} - 396\mu_{1,0} + 83\mu_{2,0} - 5\mu_{3,0}\Big) \\
&- 2c_2^4 c_4 \Big( -303 + 831\rho_0 + 111\rho_1 + 831\mu_{0,0} \\
&\quad\quad\quad -12\mu_{0,1} - 516\mu_{1,0} + 51\mu_{2,0} - \mu_{3,0}\Big) \\
&+ c_2^3 c_3^2 \Big( 1512 - 4392\rho_0 - 786\rho_1 - 4392\mu_{0,0} \\
&\quad\quad\quad +138\mu_{0,1} + 3012\mu_{1,0} - 339\mu_{2,0} + 8\mu_{3,0}\Big) \Big].
\end{aligned}
\tag{16}
$$

Moreover, note that the first two steps are optimal because it is Ostrowski's method. Therefore, we find some necessary conditions on the generic functions $\mu(t_n, s_n)$ and $\rho(u_n)$ in such a way that all the coefficients of $e_n^i$ in the general error equation become zero for $i = 4, 5, 6, 7$. To this end, we proceed as follows.

We have $C_4 = -c_2(c_2^2 - c_3)(-1 + \rho_0 + \mu_{0,0})$. Then, to vanish this coefficient it is enough to choose $\rho_0 = 1$ and $\mu_{0,0} = 0$. If we consider this condition, then $C_5 = -c_2^2(c_2^2 - c_3)(-2 + \mu_{1,0})$. Again, to vanish this coefficient, we chose $\mu_{1,0} = 2$. Repeating this argument leads to $C_6 = (1/2)c_2(c_2^2 - c_3)(-2c_3(-1 + \rho_1) + c_2^2(-12 + 2\rho_1 + \mu_{2,0}))$. It turns to the conditions $\rho_1 = 1$ and $\mu_{2,0} = 10$. Finally, $C_7 = -(1/6)c_2^2(c_2^2 - c_3)(-6c_3(-4 + \mu_{0,1}) + c_2^2(-96 + 6\mu_{0,1} + \mu_{3,0}))$ for which we obtain $\mu_{0,1} = 4$ and $\mu_{3,0} = 72$.

Vanishing the coefficient of $e_n^7$ states that, by considering the above conditions for the weight functions (a.k.a. generic functions), we then could have an error equation of the form $e_{n+1} = c_2(c_2^2 - c_3)(41c_2^4 - 15c_2^2 c_3 + c_3^2 + c_2 c_4)e_n^8 + O(e_n^9)$. This error reveals that we have constructed an optimal eighth-order scheme and the proof is completed.                                                    □

At this time, we introduce specific sets of generic functions, satisfying obtained conditions (6) and also applying them to (3) for constructing concrete methods. For example, we could have

$$
\begin{aligned}
\mu(t, s) &= 2t + 4s + 5t^2 + 12t^3, \\
\rho(u) &= 1 + \frac{u}{1 + u^2},
\end{aligned}
\tag{17}
$$

which satisfy the conditions of Theorem 1. Thus, one iteration scheme can be defined as

$$
\begin{aligned}
y &= x - \frac{f(x)}{f'(x)}, \\
z &= y - \frac{f(x)}{f(x) - 2f(y)} \frac{f(y)}{f'(x)}, \\
\hat{x} &= z - \left(1 + 2t + 4s + 5t^2 + 12t^3 + \frac{u}{1 + u^2}\right) \frac{f(z)}{f'(x)}.
\end{aligned}
\tag{18}
$$

We could also have $\mu(t, s) = -1 + 2t + 6s + (1/(1 + 2s - 5t^2 - 12t^3))$, or $\mu(t, s) = (22t^3 + 9t^2 + 2t + 4s)/(1 + 2t)$. And also $\rho(u) = (2 + u)/(2 - u)$.

## 3. A New Optimal Sixteenth-Order Family

This section is concerned with construction of a new sixteenth-order family. We add Newton's step to the obtained concrete method (18) in the previous section as follows:

$$
\begin{aligned}
y &= x - \frac{f(x)}{f'(x)}, \\
z &= y - \frac{f(x)}{f(x) - 2f(y)} \frac{f(y)}{f'(x)}, \\
w &= z - \left(1 + 2t + 4s + 5t^2 + 12t^3 + \frac{u}{1 + u^2}\right) \frac{f(z)}{f'(x)}, \\
\hat{x} &= w - \frac{f(w)}{f'(w)}.
\end{aligned}
\tag{19}
$$

Although the first three steps are optimal and iteration (19) has convergence order sixteen, the whole procedure is not optimal since it is not satisfying Kung and Traub conjecture. The iteration (19) reaches order 16 with 6 functional evaluations. To obtain an optimal method, we must reduce one functional evaluation. To this end, our goal is to approximate $f'(w)$ in terms of the other existing function values, that is, $f(x)$, $f(y)$, $f(z)$, $f(w)$, and $f'(x)$, that is similar to the case of Section 2. Finding this approximation sounds easy at first, but when we tried to do, it was not simple at all. In fact, after trying many generic functions we had succeeded.

In addition, the general approach is very similar to the previous section and to save the space of the present paper we avoid repeating some cumbersome details. Let

$$
\begin{aligned}
f'(w)^{-1} = \Big( &1 + 2t + 5t^2 + 12t^3 + 4s + \frac{u}{1 + u^2} \\
&+ p(t, s) + q(s, u) + g(t, s, v) + h(t, s, v, u) \Big) \\
&\times \left(f'(x)\right)^{-1},
\end{aligned}
\tag{20}
$$

where $v = f(w)/f(z)$; the generic functions $p(t, s)$, $q(s, u)$, $g(t, s, k)$, and $h(t, s, k, u)$ are supposed to be determined in

such a way that scheme (19) has optimal order sixteen. We can express the following relations:

$$p(t,s) = p_{0,0} + p_{1,0}t + p_{0,1}s$$

$$+ \frac{1}{2!}\left(p_{2,0}t^2 + 2p_{1,1}ts + p_{0,2}s^2\right)$$

$$+ \frac{1}{6}p_{3,0}t^3 + \frac{1}{6}p_{3,1}t^3s + \frac{1}{24}p_{4,0}t^4 + \frac{1}{24}p_{4,1}t^4s,$$

$$q(s,u) = q_{0,0} + q_{1,0}s + q_{0,1}u + q_{1,1}su$$

$$+ \frac{1}{2}q_{0,2}u^2 + \frac{1}{6}q_{0,3}u^3 + \frac{1}{2}q_{1,2}su^2 + \frac{1}{6}q_{1,3}su^3,$$

$$g(t,s,v) = g_{0,0,0} + g_{1,0,0}t + g_{0,1,0}s + g_{0,0,1}v$$

$$+ g_{1,1,0}ts + g_{1,0,1}tv + g_{0,1,1}sv + g_{1,1,1}tsv,$$

$$h(t,s,v,u) = h_{0,0,0,0} + h_{1,0,0,0}t + h_{0,1,0,0}s + h_{0,0,1,0}v$$

$$+ h_{0,0,0,1}u + h_{1,1,0,0}ts + h_{1,0,1,0}tv$$

$$+ h_{1,0,0,1}tu + h_{0,0,1,1}vu + h_{0,1,0,1}su$$

$$+ h_{0,1,1,0}sv + h_{1,1,1,0}tsv + h_{1,1,0,1}tsu$$

$$+ h_{1,0,1,1}tvu + h_{0,1,1,1}svu + h_{1,1,1,1}tsvu$$

$$+ \frac{1}{2}\left(h_{2,0,0,0}t^2 + h_{2,0,0,1}t^2u\right.$$

$$\left. + h_{2,0,1,0}t^2v + h_{2,1,0,0}t^2s\right)$$

$$+ \frac{1}{2}\left(h_{2,0,1,1}t^2vu + h_{2,1,0,1}t^2su\right.$$

$$\left. + h_{2,1,1,0}t^2sv + h_{2,1,1,1}t^2svu\right)$$

$$+ \frac{1}{6}\left(h_{3,0,0,0}t^3 + h_{3,1,0,0}t^3s\right.$$

$$\left. + h_{3,0,1,0}t^3v + h_{3,0,0,1}t^3u\right)$$

$$+ \frac{1}{6}\left(h_{3,1,1,0}t^3sv + h_{3,1,0,1}t^3su\right.$$

$$\left. + h_{3,0,1,1}t^3vu + h_{3,1,1,1}t^3svu\right),$$

$$(21)$$

where $p_{i,j} = [\partial^{i+j}p(t,s)/\partial^i t\partial^j s]_{(0,0)}$, $q_{i,j} = [\partial^{i+j}q(s,u)/\partial^i s\partial^j u]_{(0,0)}$, $g_{i,j,k} = [\partial^{i+j+k}g(t,s,v)/\partial^i t\partial^j s\partial^k v]_{(0,0,0)}$,

$$h_{i,j,k,l} = \left[\frac{\partial^{i+j+k+l}h(t,s,v,u)}{\partial^i t\partial^j s\partial^k v\partial^l u}\right]_{(0,0,0,0)}, \quad i,j,k,l \geq 0. \quad (22)$$

Finding the required conditions for the new generic functions $p, q, g$, and $h$ is very similar to the above approach in determination functions $\mu(t_n, s_n)$ and $\rho(u_n)$. Therefore, we provide and address its Mathematica source in the Appendix due to limited space. We summarize the obtained results in the following theorem.

**Theorem 2.** *Let $\alpha \in D$ be a simple zero of the sufficiently differentiable function $f:D \subset R \to R$ and $c_i = f^{(i)}(\alpha)/(f'(\alpha)i!)$, $i \geq 2$. If $x_0$ is sufficiently close to $\alpha$, then* (1) *the order of convergence for* (19)-(20) *is sixteen if*

$$p_{0,0} = p_{1,0} = p_{0,1} = p_{2,0} = p_{0,2}$$
$$= p_{3,0} = p_{4,0} = 0,$$
$$p_{1,1} = 1, \qquad p_{2,1} = 12,$$
$$p_{3,1} = -90, \qquad p_{4,1} = -1968$$

$$q_{0,0} = q_{1,0} = q_{0,1} = q_{0,2} = 0$$

$$q_{1,1} = 2, \qquad q_{0,3} = -6, \qquad q_{1,2} = -8,$$

$$g_{0,0,0} = g_{1,0,0} = g_{0,0,1} = g_{0,1,0} = 0$$

$$g_{0,0,1} = 1, \qquad g_{1,0,1} = 2, \qquad g_{0,1,1} = 8, \qquad (23)$$

$$h_{0,0,0,0} = h_{1,0,0,0} = h_{0,0,0,1} = h_{2,0,0,0}$$
$$= h_{0,1,0,0} = h_{1,0,0,1} = h_{3,0,0,0} = 0,$$

$$h_{0,0,1,0} = h_{1,1,0,0} = h_{2,0,0,1} = h_{0,1,0,1} = h_{1,0,1,0}$$
$$= h_{2,1,0,0} = h_{3,1,0,0} = h_{0,1,1,0} = 0,$$

$$h_{3,0,0,1} = -12, \qquad h_{0,0,1,1} = 2,$$

$$h_{2,0,1,0} = 10, \qquad h_{1,1,0,1} = -3,$$

$$h_{3,0,1,0} = 72, \qquad h_{2,1,0,1} = -44.$$

(2) *Its error equation is*

$$e_{n+1} = \left[-\frac{1}{6}c_2^2\left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2c_3 + c_3^2 + c_2c_4\right)\right.$$

$$\times \left(-18c_3^2c_4 + 3c_2^4c_4\right.$$

$$\times \left(2\left(-98 + g_{1,1,1} + h_{1,1,1,0}\right) + h_{2,0,1,1}\right)$$

$$- 3c_2^2c_3c_4\left(2\left(-47 + g_{1,1,1} + h_{1,1,1,0}\right) + h_{2,0,1,1}\right))$$

$$- \left(3c_2 \times \left(2c_4^2 + 2c_3c_5 + c_3^3\right.\right.$$

$$\times \left(2\left(-40 + g_{1,1,1} + h_{1,1,1,0}\right) + h_{2,0,1,1}\right)))$$

$$- 2c_2^5c_3\left(-4686 + 168g_{1,1,1} + 168g_{1,1,1,0}\right.$$

$$\left. + 84h_{2,0,1,1} + h_{3,1,0,1}\right)$$

$$+ c_2^7\left(3\left(-3752 + 82g_{1,1,1}\right.\right.$$

$$\left. + 82h_{1,1,1,0} + 41h_{2,0,1,1}\right) + h_{3,1,0,1})$$

$$+ c_2^3\left(6c_5 + c_3^2\right.$$

$$\times \left(-2730 + 96g_{1,1,1} + 96h_{1,1,1,0}\right.$$

$$\left.\left.\left. + 48h_{2,0,1,1} + h_{3,1,0,1}\right)\right)\right]e_n^{16} + O\left(e^{17}\right).$$

$$(24)$$

*Proof.* See the Appendix. □

Now, we introduce specific sets of generic functions, satisfying obtained conditions (23), and apply it in (19), to construct concrete methods. Hence, we attain

$$p(t, s) = st + 6t^2 s - 15t^3 s - 82t^4 s,$$

$$p(t, s) = -1 + \frac{1}{1 - st - 6t^2 s + 15t^3 s + 82t^4 s},$$

$$p(t, s) = st\left(1 + 6t + t^2(15 + 82t)\right)(ts - 1),$$

$$q(s, u) = 2su - u^3 - 4su^2,$$

$$q(s, u) = \frac{2su}{1 + 2u} - u^3,$$

$$q(s, u) = 2su(1 + u + 4s) - (s + u)^2(u + 4s),$$

$$g(t, s, v) = v + 8sv + 2tv,$$

$$g(t, s, v) = \frac{v}{1 - 2t} + 8sv,$$

$$g(t, s, v) = (s^2 + v)\left((1 + t)^2 + 8s\right),$$

$$h(t, s, v, u) = -2t^3 u + 2vu - 3tsu$$
$$+ 12t^3 v - 22t^2 su + 5t^2 v,$$

$$h(t, s, v, u) = (1 + tv)\left((12v - 2u)t^3 + v^3\right)$$
$$+ (5v - 22su)(t + s)^2$$
$$- ts(44su + 10v + 3u) + 2vu,$$

$$h(t, s, v, u) = (12v - 2u)\left(t^3 + v^3\right) + (5v - 22su)(t + v)^2$$
$$+ tu(44sv - 3s) + 2k(u - tv).$$

(25)

Each derived method from the scheme (3) by considering the suitable weight functions achieves the optimal efficiency index $16^{1/5} \approx 1.741$, which is the same to [12] and is greater than $2^{1/2} \approx 1.414$ of the Newton's method, $4^{1/3} \approx 1.587$ of the optimal fourth-order methods (e.g., see [13]), and $8^{1/4} \approx 1.682$ of the optimal three-point methods such as (18) or the methods given in [14] and even with memory methods recently developed in [15]. This shows that the contribution in this work is quite effective.

Note that although we have performed all the developments by considering $f: D \subseteq R \to R$, all the theorems could be extended and deduced if the function $f$ is defined in the complex plane as $f: D \subseteq C \to C$, or having a complex zero. In such case a complex seed (initial guess) is needed for converging.

## 4. Application in Art

Kalantari in [9] coined the term polynomiography as a clear application of fixed-point iterative methods in producing beautiful fractal pictures, which are in use in computer graphics and art. Polynomiography is defined to be the art

and science of visualization in approximation of the zeros of complex polynomials, via fractal and nonfractal images created using the mathematical convergence properties of iteration functions.

As is known according to the Fundamental Theorem of Algebra, a polynomial of degree $n$, with real or complex coefficients, has $n$ zeros (roots) which may or may not be distinct.

In this section, using machine precision and the computer programming package Mathematica 8 (see, e.g., [16]), we produce some of such beautiful fractals obtained from our new methods. In fact, an iteration function is a mapping of the plane into itself; that is, given any point in the plane, it is a rule that provides another point in the plane. This section is necessary in this paper to show how the new schemes could be considered in polynomiography.

Consider a polynomial $p(z)$ and a fixed natural number $m \geq 2$. The basins of attraction of a root of $p(z)$ with respect to the iteration function $B_m(z)$ are regions in the complex plane such that, given an initial point $a_0$ within them, the corresponding sequence $a_{k+1} = B_m(a_k)$, $k = 0, 1, \ldots$, will converge to that root (see for more details [17, 18]).

It turns out that the boundary of the basins of attraction of any of the polynomial roots is the same set. This boundary is known as the Julia set and its complement is known as the Fatou set.

From now on, a complex rectangle $D = [-3, 3] \times [-3, 3]I \subseteq C$ will be considered and we subsequently assign a color to each complex point $z_0 \in D$ according to the root, at which the corresponding method starting from $z_0$ converges. We assign a color to each attraction basin of a root. But, we further make the color lighter or darker according to the number of iterations needed to reach the root with the fixed precision required $10^{-2}$.

An important aspect of the work of Kalantari which emerged in his *Carpet design* ([9]) is in applying iterations functions for higher order polynomials with various styles of coloring. We herein use six different polynomials with different coloring forms in Mathematica 8. Toward such an option, we used three different colorings as indicated in Figure 1. The results of basins of attraction (by (18)), which provide beautiful pictures, are given in Figures 2, 3, and 4, via coloring described in Figure 1.

## 5. All the Simple Real Zeros

The fixed-point type iterative methods might diverge if the conditions of the main theorems given in Sections 2 and 3 fail. Actually, an important advantage of multipoint optimal schemes, as the ones in Sections 2 and 3, is that their convergence could not be achieved without having a robust approximation of the position of the zeros. This sometimes is referred to as the most important difficulty of iterative methods in practical problems. Besides, in practice one wants to find all the real solutions of a nonlinear function at the same time.

To remedy such shortcomings, only a few approaches such as interval methods (see, e.g., [19] or [20]) have been

FIGURE 1: The colors "`LightTemperatureMap`," "`RedBlueTones`," and "`DarkBands`" (from left to right, respectively).


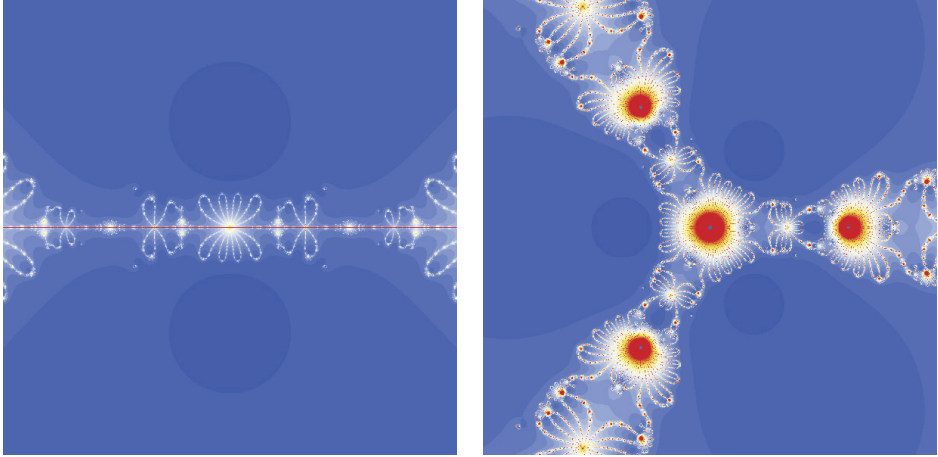
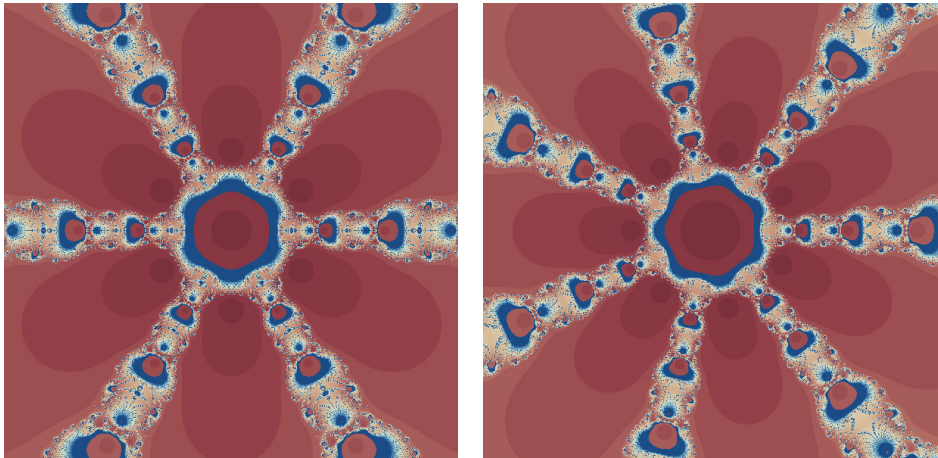FIGURE 2: The fractal pictures obtained by (18) using the polynomials $p(z) = z^2 + 1$ and $p(z) = z^3 + 1$.



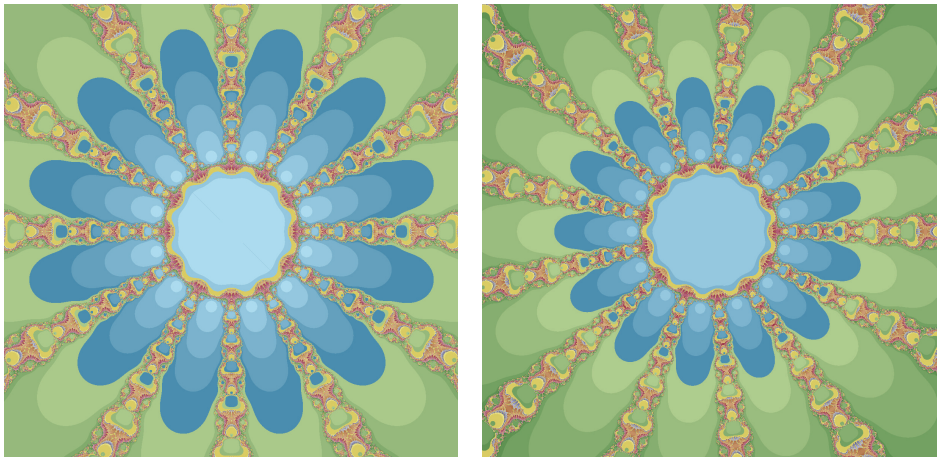FIGURE 3: The fractal pictures obtained by (18) using the polynomials $p(z) = z^5 + 1$ and $p(z) = z^6 + 1$.



FIGURE 4: The fractal pictures obtained by (18) using the polynomials $p(z) = z^{12} + 1$ and $p(z) = z^{13} + 1$.

```
Clear["Global`*"]
f[x_]:= ChebyshevT[20, x] − ChebyshevU[40, x] + Sin[x + 1] − 11/10;
a = −1.; b = +1.;
p = Plot[f[x], {x, a, b}, Mesh -> {{0}}, MeshFunctions -> (f[#] &),
   PlotPoints -> 1000, PerformanceGoal -> "Quality",
   WorkingPrecision -> 128];
```

<div align="center">ALGORITHM 1</div>

```
seeds = Cases[Normal[p], Point[z_]:> z[[1]], Infinity];
setInitial = Sort[seeds]
NumberOfGuesses = Length[setInitial]
```

<div align="center">ALGORITHM 2</div>

given to the literature up to now. For instance, Yun in [21] applied a numerical integration based technique toward this goal. His technique is robust, but it might be time consuming in case of having so many zeros in an interval.

Herein we provide a predictor-corrector algorithm based on Mathematica 8 software consisting of two parts. In the first part, we apply the procedure given by Wagon in [10] for extracting robust initial approximations for all the simple real zeros. And second is to apply any of the presented optimal methods in this paper as a corrector to increase the accuracy of the solution in a short piece of time, when high precision computing is needed. To illustrate the procedure, we start by the fact that we attempt to find all the axis crossings; that is to say, we will not attempt to capture simple zeros at which there is no axis crossing.

Toward this end and as Wagon did in [10], we can use the options of plotting as a function and to find the crossings. Thus, one picks out the points from the normal graphics form of the plot $p$. Note that the setting of MeshFunctions must be a pure function and cannot be just $f$.

To further illustrate, consider the following example in which the whole of the procedure is to extract all the simple real zeros of a nonlinear equation $f(x) =$ ChebyshevT$[20, x]$ − ChebyshevU$[40, x]$ + Sin$[x + 1]$ − $11/10 = 0$ in a given interval $D = [−1., 1.]$(see Algorithm 1).

We used PerformanceGoal-> "Quality" and WorkingPrecision-> 128 in order to obtain high reliable initial approximations for the crossing of the function to the $x$-axis. Note that some of such considerations for some problems could also be changed adaptively for providing better feedbacks. In what follows, we extract the initial guesses, sort them, and obtain the (estimate) number of zeros in the considered interval (see Algorithm 2).

The plot of the function $f$ is given in Figure 5, which clearly reveals the difficulty in obtaining all the real solutions. But the hybrid algorithm descried in this section could be applied for all the real simple solutions. For this test, we have 40 zeros which must be found in high precision accuracy. The initial list of approximations would be {−0.995857, −0.989387, −0.972279, −0.953544, −0.92376, −0.905373,
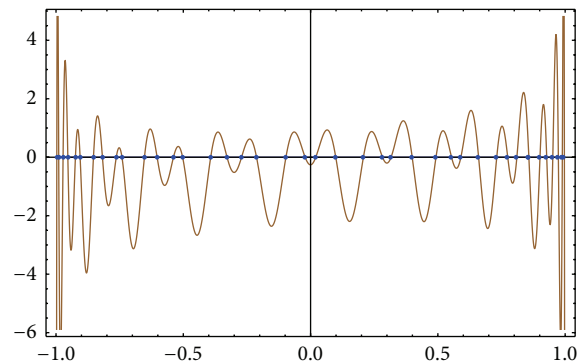


FIGURE 5: The plot of the function $f(x)$ along with its finitely many simple real zeros on the interval $[−1, 1]$.

−0.852924, −0.817623, −0.763259, −0.740905, −0.652616, −0.602838, −0.53862, −0.502713, −0.392391, −0.328099, −0.271123, −0.212284, −0.096817, −0.0213889, 0.0204006, 0.0989106, 0.207534, 0.282518, 0.316156, 0.399215, 0.492675, 0.553758, 0.590388, 0.660032, 0.731315, 0.774491, 0.81035, 0.857463, 0.901219, 0.927616, 0.951482, 0.973356, 0.988923, 0.996117}.

Now the corrector part of our algorithm could be written in what follows for the optimal scheme (18) (see Algorithm 3).

In the above piece of code, each member of the list of initial approximations obtained from the predictor step of our algorithm will be corrected until its residual, that is, $|f(x_n)| \leq 10^{-1000}$, while we work with 2000 number of fixed point arithmetics. Note that the other iterative solvers could be coded in a similar way.

An interesting point is that one may ask that higher optimal order is equal to higher computational load and thus the whole procedures might be inefficient in terms of computational time. We clearly respond that there is no such thing for optimal schemes and especially when we mix them with a convergence guaranteed algorithm; that is to say, the computation of functional evaluations is much more expensive than operational cost when working in high

```
digits = 2000;
For[i = 1, i <= NumberOfGuesses, i++,
    {k = 0; X = SetAccuracy[setInitial[[i]], digits];
    While[Abs[f[X]] > 10^-1000 && k <= 20,
        {k = k + 1; fX = SetAccuracy[f[X], digits];
        f1X = SetAccuracy[f'[X], digits];
        Y = SetAccuracy[X - fX/f1X, digits];
        fY = SetAccuracy[f[Y], digits];
        Z = SetAccuracy[Y - (fX/(fX - 2 fY)) (fY/f1X), digits];
        fZ = SetAccuracy[f[Z], digits]; t = fY/fX; s = fZ/fX; u = fZ/fY;
        Weight = SetAccuracy[1 + 2 t + 4 s + 5 t^2 + 12 t^3 + u/(1 + u^2),
            digits];
        X = SetAccuracy[Z - Weight (fZ/f1X), digits];};];
    Print[Column[{
        "The number of full iterations is:" k,
        "The zero is:" N[X, 64],
        "The residual norm of the approximate zero is:" N[Abs[f[X]],
            5]}, Background -> {{LightRed, LightBrown, LightBlue}},
        Frame -> All]];}]; // AbsoluteTiming
```

ALGORITHM 3

TABLE 1: Comparison of different optimal methods for finding all the zeros of f[x_].

| Iterative methods | Newton | (18) | (A.19) |
|---|---|---|---|
| Computational time (in seconds) | 6.06 | 4.82 | 3.37 |

precision computing. This note alongside the efficiency of our new schemes is totally revealed in Table 1, whereas the total elapsed time for three different correctors has been compared for the above numerical example. One may easily observe that the new optimal 16th order method (by considering $p(t,s) = st + 6t^2s - 15t^3s - 82t^4s$, $q(s,u) = 2su - u^3 - 4su^2$, $g(t,s,v) = v + 8sv + 2tv$, and $h(t,s,v,u) = -2t^3u + 2vu - 3tsu + 12t^3v - 22t^2su + 5t^2v$) beats the other solvers in terms of computational time (in seconds). Note that, due to page limitation, we do not include the approximation of the 40 zeros in the considered interval which are basically correct up to 1500 decimal places.

Note that, the computer specifications in this paper are Intel(R) Core (TM) 2 Quad CPU, Q9550 @ 2.83 GHz with 2.00 GB of RAM. The theoretical number of real solutions in an interval is important for the approach shown in this paper to succeed in finding all the simple real zeros in the desired interval.

The theoretical number of real zeros in the desired interval must be known in advance. Let us consider the above test equation in a given interval $D = [-1, 1]$. We can show that $f$ has only 40 zeros which are all real and contained in the interval $D = [-1, 1]$ by using element properties of Chebyshev polynomials of either kind and Weierstrass approximation theorem. We first of all observe that the smooth function $\sin(x+1) - 11/10$ in $D$ can be approximated by a cubic polynomial: $p_3(x) = -0.258529 + 0.519736x - 0.386822x^2 - 0.065087x^3$, with maximum interpolating absolute error of 0.0151886 at $x = -0.645823$ via Weierstrass
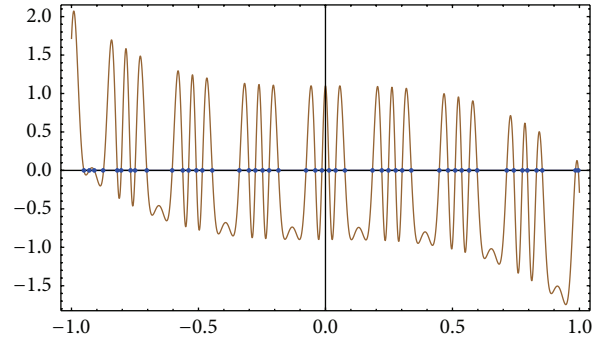


FIGURE 6: The plot of the function $g(x)$ along with its finitely many simple real zeros on the interval $[-1, 1]$.

approximation theorem. Hence, there exist at most 40 real zeros in $D$.

Now, we compose $f(x) = f_1(x) + f_2(x)$, where $f_1(x) = ChebyshevT[20, x] - ChebyshevU[40, x]$ and $f_2(x) = +\operatorname{Sin}[x + 1] - 11/10$. Then, by the properties of Chebyshev polynomials of either kind, $f_1$ has only 40 real zeros in $D$. Indeed, $f_1(x)$ has double zeros at $x = 0$ since ChebyshevT[20, x] - ChebyshevU[40, x] = 0 and ChebyshevT[20, x]' - ChebyshevU[40, x]' = 0, while all the remaining 38 simple real zeros are in $D$. Since $f_2$ moves $f$ downward slightly, the double zero at $x = 0$ splits into two simple zeros near $x = 0$, and $f_2(0) = -0.258529$. But in the remaining locations, the overall downward movement of $f_2(x)$ does not affect the number of real zeros of $f(x)$. Consequently, the theoretical number of real roots for $f(x) = 0$ is 40.

We also compare different optimal methods in finding all the simple real zeros of the nonlinear oscillatory function g[x_] := cos[10Sin[12x]] - x^3 + 1/10 on the interval $D = [-1., 1.]$ with the same stopping criterion as in the above test. The graph of this function is illustrated in Figure 6 and

TABLE 2: Comparison of different optimal methods for finding all the zeros of g[x_].

| Iterative methods | Newton | (18) | (A.19) |
| --- | --- | --- | --- |
| Computational time (in seconds) | 9.26 | 6.10 | 5.04 |

the results of comparison in terms of the elapsed time are reported in Table 2. The list of initial approximations would be {−0.951071, −0.929957, −0.910096, −0.873815, −0.819725, −0.804222, −0.767296, −0.749607, −0.703525, −0.603417, −0.561856, −0.538937, −0.508479, −0.484731, −0.445715, −0.338982, −0.301488, −0.275968, −0.24768, −0.221955, −0.185071, −0.0766472, −0.0399471, −0.0139904, 0.0139903, 0.0399483, 0.0766347, 0.185246, 0.221749, 0.247938, 0.275612, 0.302007, 0.33791, 0.448186, 0.48259, 0.510735, 0.536286, 0.565254, 0.597367, 0.713616, 0.741522, 0.775409, 0.795057, 0.831029, 0.853649, 0.984546, 0.995607}. We have 47 zeros in this interval which shows the difficulty of finding all the simple real zeros. Hopefully the hybrid algorithm given above works well in this case for finding all the simple real zeros in the interval. Again, the new optimal sixteenth-order method beats the other schemes.

Without the theoretical number of real zeros in the desired interval, it would be also difficult to choose the number of initial guesses. It must be noted that this approach must only be taken into account for finding the real simple zeros of nonlinear equations. In fact, if the nonlinear function has zeros with multiplicity, then the fast approach of Wagon must be replaced by a verified method, as the ones discussed in [22] to extract all the simple and multiple real solutions of a nonlinear function as the seeds for the simple and multiple zero-finders developed in this paper.

## 6. Concluding Remarks

This paper has contributed two optimal eighth- and sixteenth-order methods for solving nonlinear equations using generic functions (weight function) in the computer programming package Mathematica 8. It was observed that any derived method from the new optimal methods possesses 1.682 and 1.741 as the optimal computational efficiency indices.

From Tables 1 and 2 and the tested examples, we can conclude that all implemented methods for nonlinear functions converge fast in a shorter piece of time in contrast to the lower order schemes when a list of powerful initial guesses is available for all the simple real zeros in the interval.

The application of one of the new schemes, that is, (18), was given by producing beautiful fractal pictures useful in computer graphics as Kalantari mentioned. We note that keeping tighter conditions will produce fractals with much more quality and reliability. For further application, refer to [4, 23, 24].

We also used the programming package Mathematica 8 in our calculations and gave the necessary cautions and pieces of codes for the users to implement them in their own problems as easily as possible. We have designed a hybrid algorithm to capture all the simple real solutions of nonlinear equations as

rapidly as possible using a similar technique introduced by Stan Wagon. The algorithm worked efficiently for very hard test problems. And thus, the proposed algorithm could be easily used in practical problems.

## Appendix

## Proof of Theorem 2

Consider the following:

$$
\begin{aligned}
&f[e\_] := f[e] = f1a * \left(e + \sum_{i=2}^{16} c_i * e^i\right); \\
&ey = e - \text{Series}\left[\frac{f[e]}{f'[e]}, \{e, 0, 16\}\right]; \quad (*\text{first step}*) \\
&ez = ey - \frac{f[e]}{f[e] - 2 * f[e]} * \frac{f[e]}{f'[e]}; \quad (*\text{Second step}*) \\
&\quad t := \frac{f[ey]}{f[e]}; \qquad s := \frac{f[ez]}{f[e]}; \\
&\quad u := \frac{f[ez]}{f[ey]}, \qquad v := \frac{f[ew]}{f[ez]}; \\
&\mu[t\_, s\_] = 1 + 2 * t + 5 * t^2 + 12 * t^3 + 4 * s; \\
&\rho[u\_] = \frac{u}{1 + u^2}; \\
&ew = ez - h[t, s, u] * \frac{f[ez]}{f[e]}; \quad (*\text{Third step}*) \\
&e1 = ew - \mu[t, S] + \rho[u] + p(t, S) + q(S, u) \\
&\qquad + g(t, S, v) + h(t, S, v, u) \\
&\qquad \times \frac{f[ew]}{f'[e]}; \quad (*\text{Fourth step}*).
\end{aligned}
$$

$$(A.1)$$

The first three steps are optimal according to Section 2. Now, our goal is to add another step and identify some appropriate conditions in such a way that produces an optimal four-step iterative method. Thus, we have

$$
\begin{aligned}
p[t\_, s\_] &= p_{0,0} + p_{1,0}t + p_{0,1}s \\
&\quad + \frac{1}{2!}\left(p_{2,0}t^2 + 2p_{1,1}ts + p_{0,2}s^2\right) + \frac{1}{6}p_{3,0}t^3 \\
&\quad + \frac{1}{6}p_{3,1}t^3 s + \frac{1}{24}p_{4,0}t^4 + \frac{1}{24}p_{4,1}t^4 s, \\
q[s\_, u\_] &= q_{0,0} + q_{1,0}s + q_{0,1}u + q_{1,1}su \\
&\quad + \frac{1}{2}q_{0,2}u^2 + \frac{1}{6}q_{0,3}u^3 + \frac{1}{2}q_{1,2}su^2 + \frac{1}{6}q_{1,3}su^3, \\
g[t\_, s\_, v\_] &= g_{0,0,0} + g_{1,0,0}t + g_{0,1,0}s + g_{0,0,1}v \\
&\quad + g_{1,1,0}ts + g_{1,0,1}tv + g_{0,1,1}sv + g_{1,1,1}tsv,
\end{aligned}
$$

$h\left[t_-, s_-, v_-, u_-\right]$

$$= h_{0,0,0,0} + h_{1,0,0,0}t + h_{0,1,0,0}s + h_{0,0,1,0}v$$

$$+ h_{0,0,0,1}u + h_{1,1,0,0}ts + h_{1,0,1,0}tv + h_{1,0,0,1}tu$$

$$+ h_{0,0,1,1}vu + h_{0,1,0,1}su + h_{0,1,1,0}sv + h_{1,1,1,0}tsv$$

$$+ h_{1,1,0,1}tsu + h_{1,0,1,1}tvu + h_{0,1,1,1}svu + h_{1,1,1,1}tsvu$$

$$+ \frac{1}{2}\left(h_{2,0,0,0}t^2 + h_{2,0,0,1}t^2u + h_{2,0,1,0}t^2v + h_{2,1,0,0}t^2s\right)$$

$$+ \frac{1}{2}\left(h_{2,0,1,1}t^2vu + h_{2,1,0,1}t^2su\right.$$

$$\left. + h_{2,1,1,0}t^2sv + h_{2,1,1,1}t^2svu\right)$$

$$+ \frac{1}{6}\left(h_{3,0,0,0}t^3 + h_{3,1,0,0}t^3s + h_{3,0,1,0}t^3v + h_{3,0,0,1}t^3u\right)$$

$$+ \frac{1}{6}\left(h_{3,1,1,0}t^3sv + h_{3,1,0,1}t^3su\right.$$

$$\left. + h_{3,0,1,1}t^3vu + h_{3,1,1,1}t^3svu\right).$$

(A.2)

The following command produces the coefficient of $e^8$ in the error equation:

$\texttt{a8 = coefficient}\left[\texttt{e1}, e^8\right]\texttt{//FullSimplify}$

$\texttt{Out [a8]} = -c_2\left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2c_3 + c_3^2 + c_2c_4\right)$

$$\times \left(p_{0,0} + q_{0,0} + g_{0,0,0} + h_{0,0,0}\right).$$

(A.3)

To vanish the coefficient of $e^8$ it is sufficient to set

$$\boxed{\begin{array}{ll} p_{0,0} = 0; & q_{0,0} = 0; \\[2mm] g_{0,0,0} = 0; & h_{0,0,0,0} = 0. \end{array}}$$

(A.4)

Similarly, we find the following:

$\texttt{a9 = coefficient}\left[\texttt{e1}, e^9\right]\texttt{//FullSimplify}$

$\texttt{Out [a9]} = -c_2^2\left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2c_3 + c_3^2 + c_2c_4\right)$

$$\times \left(p_{1,0} + g_{1,0,0} + h_{1,0,0,0}\right).$$

(A.5)

Set

$$\boxed{p_{1,0} = 0; \qquad g_{1,0,0} = 0; \qquad h_{1,0,0,0} = 0;}$$

(A.6)

$\texttt{a10 = coefficient}\left[\texttt{e1}, e^{10}\right]\texttt{//FullSimplify}$

$\texttt{Out [a10]} = -\frac{1}{2}c_2\left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2c_3 + c_3^2 + c_2c_4\right)$

$$\times \left(- 2c_3\left(q_{0,1} + h_{0,0,0,1}\right)\right.$$

$$\left. + c_2^2\left(2q_{0,1} + p_{2,0} + 2h_{0,0,0,1} + h_{2,0,0,0}\right)\right).$$

(A.7)

Set

$$\boxed{q_{0,1} = 0; \qquad p_{2,0} = 0; \qquad h_{0,0,0,1} = 0; \qquad h_{2,0,0,0} = 0.}$$

(A.8)

$\texttt{a11 = coefficient}\left[\texttt{e1}, e^{11}\right]\texttt{//FullSimplify}$

$\texttt{Out [a11]}$

$$= -\frac{1}{6}c_2^2\left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2c_3 + c_3^2 + c_2c_4\right)$$

$$\times -6c_3\left(q_{1,0} + p_{0,1} + h_{0,1,0} + h_{0,1,0,0} + h_{1,0,0,1}\right)$$

$$+ c_2^2\left(6q_{1,0} + 6p_{0,1} + p_{3,0}\right.$$

$$\left. + 6\left(g_{0,1,0} + h_{0,1,0,0} + h_{1,0,0,1} + h_{3,0,0,0}\right)\right);$$

(A.9)

Set

$$\boxed{\begin{array}{llll} q_{1,0} = 0; & p_{0,1} = 0; & p_{3,0} = 0; & g_{0,1,0} = 0; \\[3mm] h_{0,1,0,0} = 0; & h_{1,0,0,1} = 0; & h_{3,0,0,0} = 0; & \end{array}}$$

(A.10)

$\texttt{a12 = coefficient}\left[\texttt{e1}, e^{12}\right]\texttt{//fullSimplify}$

$\texttt{Out [a12]}$

$$= -\frac{1}{24}c_2\left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2c_3 + c_3^2 + c_2c_4\right)$$

$$\times \left(24c_2c_4\left(-1 + g_{0,0,1} + h_{0,0,1,0}\right)\right.$$

$$+ 12c_3^2\left(q_{0,2} + 2\left(-1 + g_{0,0,1} + h_{0,0,1,0}\right)\right)$$

$$- 12c_2^2c_3\left(2\left(-16 + q_{0,2} + p_{1,1} + 15g_{0,0,1}\right.\right.$$

$$\left.\left. + g_{1,1,0} + 15h_{0,0,1,0} + h_{1,1,0,0}\right) + h_{2,0,0,1}\right)$$

$$+ c_2^4\left(12q_{0,2} + 24p_{1,1} + p_{4,0}\right.$$

$$+ 12\left(2\left(-42 + 41g_{0,0,1} + g_{1,1,0}\right.\right.$$

$$\left.\left.\left. + 41h_{0,0,1,0} + h_{1,1,0,0}\right) + h_{2,0,0,1}\right)\right).$$

(A.11)

Set

$$
q_{0,2} = 0; \qquad p_{1,1} = 1; \qquad p_{4,0} = 0; \qquad g_{0,0,1} = 1;
$$

$$
g_{1,1,0} = 0; \qquad h_{0,0,1,0} = 0; \qquad h_{1,1,0,0} = 0;
$$

$$
h_{2,0,0,1} = 0;
$$

$$(A.12)$$

```
a13 = coefficient[e1, e^13]//FullSimplify
```

Out[a13]

$$
= -\frac{1}{6}c_2^2 \left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2 c_3 + c_3^2 + c_2 c_4\right)
$$

$$
\times \left(6c_2 c_4 \left(-2 + g_{1,0,1} + h_{1,0,1,0}\right)\right.
$$

$$
+ 6c_3^2 \left(-4 + q_{1,1} + g_{1,0,1}\right.
$$

$$
+ h_{0,1,0,1} + h_{1,0,1,0} - c_2^2 c_3
$$

$$
\times \left(3\left(-76 + 4q_{1,1} + p_{2,1} + 30g_{1,0,1}\right.\right.
$$

$$
+ 4h_{0,1,0,1} + 30h_{1,0,1,0} + h_{2,1,0,0}\right)
$$

$$
\left. + h_{3,0,0,1}\right) + c_2^4
$$

$$
\times \left(6q_{1,1} + 3\left(-176 + p_{2,1} + 82g_{1,0,1}\right.\right.
$$

$$
+ 2h_{0,1,0,1} + 82h_{1,0,1,0} + h_{2,1,0,0}\right)
$$

$$
\left.\left. + h_{3,0,0,1}\right)\right).
$$

$$(A.13)$$

Set

$$
q_{1,1} = 2; \qquad p_{2,1} = 12; \qquad g_{1,0,1} = 2; \qquad h_{0,1,0,1} = 0;
$$

$$
h_{1,0,1,0} = 0; \qquad h_{2,1,0,0} = 0; \qquad h_{3,0,0,1} = -12;
$$

$$(A.14)$$

```
a14 = coefficient[e1, e^14]//FullSimplify
```

Out[a14]

$$
= -\frac{1}{6}c_2 \left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2 c_3 + c_3^2 + c_2 c_4\right)
$$

$$
\times \left(-6c_2 c_3 c_4 \left(-2 + \phi_{0,0,1,1}\right)\right.
$$

$$
- c_3^3 \left(-6 + q_{0,3} + 6h_{0,0,1,1}\right)
$$

$$
+ 3c_2^3 c_4 \left(-14 + 2h_{0,0,1,1} + h_{2,0,1,0}\right)
$$

$$
+ 3c_2^2 c_3^2 \left(-62 + q_{0,3} + 32h_{0,0,1,1}\right.
$$

$$
+ 2h_{1,1,0,1} + h_{2,0,1,0}\right)
$$

$$
- c_2^4 c_3 \left(-978 + 3q_{0,3} + p_{3,1} + 336h_{0,0,1,1}\right.
$$

$$
+ 12h_{1,1,0,1} + 45h_{2,0,1,0} + h_{3,1,0,0}\right)
$$

$$
+ c_2^6 \left(-1608 + q_{0,3} + p_{3,1} + 246h_{0,0,1,1}\right.
$$

$$
\left.\left. + 6h_{1,1,0,1} + 123h_{2,0,1,0} + h_{3,1,0,0}\right)\right).
$$

$$(A.15)$$

Set

$$
q_{0,3} = -6; \qquad p_{3,1} = -90; \qquad h_{0,0,1,1} = 2;
$$

$$
h_{2,0,1,0} = 10; \qquad h_{1,1,0,1} = -3; \qquad h_{3,1,0,0} = 0;
$$

$$(A.16)$$

```
a15 = coefficient[e1, e^15]//FullSimplify
```

Out[a15]

$$
= -\frac{1}{24}c_2^2 \left(c_2^2 - c_3\right)\left(41c_2^4 - 15c_2^2 c_3 + c_3^2 + c_2 c_4\right)
$$

$$
\times \left(-24c_2 c_3 c_4 \left(-8 + g_{0,1,1} + h_{0,1,1,0} + h_{1,0,1,1}\right)\right.
$$

$$
- 12c_3^3 \left(q_{1,2} + 2\left(-4 + g_{0,1,1}\right.\right.
$$

$$
\left.\left. + h_{0,1,1,0} + h_{1,0,1,1}\right)\right)
$$

$$
+ 4c_2^3 c_4 \left(6\left(-20 + g_{0,1,1} + h_{0,1,1,0} + h_{1,0,1,1}\right)\right.
$$

$$
\left. + h_{3,0,1,0}\right) + 4c_2^2 c_3^2
$$

$$
\times \left(-636 + 9q_{1,2} + 96g_{0,1,1} + 96h_{0,1,1,0}\right.
$$

$$
+ 96h_{1,0,1,1} + 3h_{2,1,0,1} + h_{3,0,1,0}\right)
$$

$$
- c_2^4 c_3 \left(36q_{1,2} + p_{4,1} + 24\right.
$$

$$
\times \left(-490 + 56g_{0,1,1} + 56h_{0,1,1,0}\right.
$$

$$
\left. + 56h_{1,0,1,1} + h_{2,1,0,1}\right) + 60h_{3,0,1,0}\right)
$$

$$
+ c_2^6 \left(12q_{1,2} + p_{4,1} + 12\right.
$$

$$
\times \left(-1424 + 82g_{0,1,1} + 82h_{0,1,1,0}\right.
$$

$$
\left.\left.\left. + 82h_{1,0,1,1} + h_{2,1,0,1}\right) + 164h_{3,0,1,0}\right)\right).
$$

$$(A.17)$$

Set

$$
q_{1,2} = -8; \qquad p_{4,1} = -1968; \qquad g_{0,1,1} = 8;
$$

$$
h_{0,1,1,0} = 0; \qquad h_{1,0,1,1} = 0; \qquad h_{3,0,1,0} = 72;
$$

$$
h_{2,1,0,1} = -44.
$$

$$(A.18)$$

By vanishing the coefficient of $e_n^{15}$, we have the following optimal four-step sixteenth-order method:

$$y = x - \frac{f(x)}{f'(x)},$$

$$z = y - \frac{f(x)}{f(x) - 2f(y)} \frac{f(y)}{f'(x)},$$

$$w = z - \varrho \frac{f(z)}{f'(x)}, \tag{A.19}$$

$$\hat{x} = w - \left[\varrho + p(t,s) + q(s,u) + g(t,s,v)\right.$$

$$\left. + h(t,s,v,u)\right] \frac{f(w)}{f'(x)},$$

wherein $\varrho = (1 + 2t + 4s + 5t^2 + 12t^3 + (u/(1 + u^2)))$. The proof is complete.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

[1] A. S. Al-Fhaid, S. Shateyi, M. Z. Ullah, and F. Soleymani, "A matrix iteration for finding Drazin inverse with ninth-order convergence," *Abstract and Applied Analysis*, vol. 2014, Article ID 137486, 7 pages, 2014.

[2] F. Soleymani, S. Shateyi, and G. Ozkum, "An iterative solver in the presence and absence of multiplicity for nonlinear equations," *The Scientific World Journal*, vol. 2013, Article ID 837243, 9 pages, 2013.

[3] F. Soleymani, "A three-step iterative method for non-linear systems with sixth order of convergence," *International Journal of Computing Science and Mathematics*, vol. 4, no. 4, pp. 363–373, 2013.

[4] F. Soleymani, M. Sharifi, S. Shateyi, and F. K. Haghani, "A class of Steffensen-type iterative methods for nonlinear systems," *Journal of Applied Mathematics*, vol. 2014, Article ID 705375, 9 pages, 2014.

[5] B. Ignatova, N. Kyurkchiev, and A. Iliev, "Multipoint algorithms arising from optimal in the sense of Kung-Traub iterative procedures for numerical solution of nonlinear equations," *General Mathematics Notes*, vol. 6, no. 2, pp. 45–79, 2011.

[6] N. Kyurkchiev and A. Iliev, "On some multipoint methods arising from optimal in the sense of Kung-Traub algorithms," *BIOMATH*, vol. 2, no. 1, pp. 1–7, 2013.

[7] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New York, NY, USA, 1964.

[8] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *The Journal of the Association for Computing Machinery*, vol. 21, no. 4, pp. 643–651, 1974.

[9] B. Kalantari, *Polynomial Root-Finding and Polynomiograpghy*, World Scientific, Singapore, 2009.

[10] S. Wagon, *Mathematica in Action*, Springer, New York, NY, USA, 3rd edition, 2010.

[11] S. Wolfram, *The Mathematica Book*, Wolfram Media, 5th edition, 2003.

[12] Y. H. Geum and Y. I. Kim, "A biparametric family of optimally convergent sixteenth-order multipoint methods with their fourth-step weighting function as a sum of a rational and a generic two-variable function," *Journal of Computational and Applied Mathematics*, vol. 235, no. 10, pp. 3178–3188, 2011.

[13] M.-Y. Lee and Y. I. Kim, "A family of fast derivative-free fourth-order multipoint optimal methods for nonlinear equations," *International Journal of Computer Mathematics*, vol. 89, no. 15, pp. 2081–2093, 2012.

[14] Y. H. Geum and Y. I. Kim, "A multi-parameter family of three-step eighth-order iterative methods locating a simple root," *Applied Mathematics and Computation*, vol. 215, no. 9, pp. 3375–3382, 2010.

[15] X. Wang and T. Zhang, "A new family of Newton-type iterative methods with and without memory for solving nonlinear equations," *Calcolo*, vol. 51, no. 1, pp. 1–15, 2014.

[16] R. Hazrat, *Mathematica: A Problem-Centered Approach*, Springer, New York, NY, USA, 2010.

[17] M. McClure, "Newton's method for complex polynomials," A preprint version of a Mathematical graphic column from Mathematica in Education and Research, 1–15, 2005.

[18] H. Susanto and N. Karjanto, "Newton's method's basins of attraction revisited," *Applied Mathematics and Computation*, vol. 215, no. 3, pp. 1084–1090, 2009.

[19] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, Academic Press, New York, NY, USA, 1983.

[20] P. Bakhtiari, T. Lotfi, K. Mahdiani, and F. Soleymani, "Interval Ostrowski-type methods with guaranteed convergence," *Annali dell'Universita'di Ferrara*, vol. 59, no. 2, pp. 221–234, 2013.

[21] B. I. Yun, "Iterative methods for solving nonlinear equations with finitely many roots in an interval," *Journal of Computational and Applied Mathematics*, vol. 236, no. 13, pp. 3308–3318, 2012.

[22] J. B. Keiper, "Interval arithmetic in *Mathematica*," *The Mathematica Journal*, vol. 5, no. 2, pp. 66–71, 1995.

[23] F. Soleymani and F. K. Haghani, "A new high-order stable numerical method for matrix inversion," *The Scientific World Journal*, vol. 2014, Article ID 830564, 10 pages, 2014.

[24] F. Soleymani, M. Sharifi, and S. Shateyi, "Approximatinghe inverse of a square matrix with application in computation of the Moore-Penrose inverse," *Journal of Applied Mathematics*, vol. 2014, Article ID 731562, 8 pages, 2014.