

OPTIMAL METHODS FOR FINDING SIMPLE AND MULTIPLE ROOTS OF NONLINEAR EQUATIONS AND THEIR BASINS OF ATTRACTION

PREM BAHADUR CHAND ^{1,2}

¹ *National Academy of Science and Technology, Dhangadhi, Kailali, Nepal*

² *South Asian University, New Delhi, India (Ph.D. Scholar)*

Email: premchand13@gmail.com

Abstract: In this paper, using the variant of Frontini-Sormani method, some higher order methods for finding the roots (simple and multiple) of nonlinear equations are proposed. In particular, we have constructed an optimal fourth order method and a family of sixth order method for finding a simple root. Further, an optimal fourth order method for finding a multiple root of a nonlinear equation is also proposed. We have used different weight functions to a cubically convergent Forntini-Sormani method for the construction of these methods. The proposed methods are tested on numerical examples and compare the results with some existing methods. Further, we have presented the basins of attraction of these methods to understand their dynamics visually.

Key Words: Non-linear equation, Newton's method, Frontini-Sormani method, weight function, multiple root, basin of attraction.

2010 AMS Subject Classification. 65H05.

1. INTRODUCTION

While working in the field of physical science, engineering and social science, non linearity is the integral part of human life. Non linear equations are frequently appeared while modeling any physical problem into mathematical form. Analytical methods generally fail to solve such type of problems and therefore the iterative methods are the best alternate. As a result, solutions of nonlinear equations by iterative methods has been a great area of interest for long time.

One of the most commonly used iterative method to solve nonlinear equation $f(x) = 0$ is the well known Newton's method given by

$$(1.1) \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Method 1.1 converges quadratically for simple roots and linearly for multiple roots. However, for multiple roots with known multiplicity m , the quadratically convergent modified

Newton's method is given by

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)}.$$

Many researchers keep contributing towards method 1.1 to improve the order of convergence in several ways. Among many others, Traub [24] and later the same method with different approach rediscovered by Frontini and Sormani [14] presented the following third order method for finding simple root of a nonlinear equation:

$$(1.2) \quad \begin{aligned} y_n &= x_n - \frac{1}{2} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(y_n)} \end{aligned}$$

We notice that the method (1.2) requires three functions evaluations per iteration to attain third order of convergence, so it is not an optimal in the sense of Kung-Traub [18]. The main goal and motivation in the construction of new method is not only to increase the order of convergence but also to achieve highest possible computational efficiency of the method. There are various techniques for increment of the order of convergence of an iterative method. One of them is using weight functions on a non optimal method. Babajee et al. [2], Chand et al. [7], Chun et al. [10], Sharma et al. [22] and the references therein and many more have worked in this area.

In this direction, Sharifi et al. [21], used weight functions on a third order Heun's method, and proposed a fourth order method given by

$$(1.3) \quad \begin{aligned} y_n &= x_n - \frac{2}{3} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \frac{f(x_n)}{4} \left(\frac{1}{f'(x_n)} + \frac{3}{f'(y_n)} \right) \left(1 + \frac{3}{8} \left(\frac{f'(y_n)}{f'(x_n)} - 1 \right)^2 - \frac{69}{64} \left(\frac{f'(y_n)}{f'(x_n)} - 1 \right)^3 + \left(\frac{f(x_n)}{f'(y_n)} \right)^4 \right). \end{aligned}$$

Babajee [2], presented the following fourth order method using weight function:

$$(1.4) \quad \begin{aligned} y_n &= x_n - \frac{2}{3} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \left(1 + \frac{1}{4}(t-1) + \frac{3}{8}(t-1)^2 \right) \frac{f(x_n)}{f'(y_n)} \end{aligned}$$

where $t = \frac{f'(y_n)}{f'(x_n)}$.

Further, using lower order methods, some higher order methods have been presented in the literature. Among many others, Sharma et al. [23], presented a family of sixth order method given by

$$(1.5) \quad \begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n) - 2f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \frac{f(x_n) + af(y_n)}{f(x_n) + (a-2)f(y_n)}, \end{aligned}$$

where, $a \in \mathbb{R}$.

In [11], Chun and Neta presented following sixth order method obtained from Kung-Traub method [18]

$$(1.6) \quad \begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{1}{\left(1 - \frac{f(y_n)}{f(x_n)}\right)^2}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \frac{1}{\left(1 - \frac{f(y_n)}{f(x_n)} - \frac{f(z_n)}{f(x_n)}\right)^2}. \end{aligned}$$

Again, using the iterative methods for simple roots, authors have developed new methods for finding multiple roots of known multiplicity $m \geq 1$. The first multipoint optimal method for multiple roots was developed by Li et al. [16] in 2009, almost a half century after constructing the first optimal multipoint method for simple roots by Ostrowski [20] in 1960. After that many authors have studied and developed many iterative methods for multiple roots. Interested readers can follow [4], [8], [17], [22] and references therein.

For a known multiplicity m , in 2013, Liu et al. [17] presented a fourth order method given by:

$$(1.7) \quad \begin{aligned} y &= x_n - m \frac{f(x_n)}{f'(x_n)}, \\ w_n &= \left(\frac{f'(y_n)}{f'(x_n)} \right)^{\frac{1}{m-1}}, \\ x_{n+1} &= y_n - m \left(w_n + \left(\frac{2m}{m-1} \right) w_n^2 \right) \frac{f(x_n)}{f'(x_n)}. \end{aligned}$$

In 2010, Sharma et al. [22] presented following fourth order method based on modified Jarrat's method:

$$(1.8) \quad \begin{aligned} y_n &= x_n - \frac{2m}{m+2} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - a_1 w_1 - a_2 w_2 - a_3 \frac{w_2^2}{w_1}, \end{aligned}$$

where $w_1 = \frac{f(x_n)}{f'(x_n)}$, and $w_2 = \frac{f(x_n)}{f'(y_n)}$. The values of the parameters are given by

$$\begin{aligned} a_1 &= \frac{m}{8}(m^3 - 4m + 8), \\ a_2 &= -\frac{m}{4}(m-1)(m+2)^2 \left(\frac{m}{m+2} \right)^m, \\ a_3 &= \frac{m}{8}(m+2)^3 \left(\frac{m}{m+2} \right)^{2m}. \end{aligned}$$

Since, dynamical properties of a rational operator associated with an iterative method, gives important information about the convergence, efficiency and stability of the method,

from the last few decades, study of dynamical behavior of such operators has become a rapidly growing area of research. There is an extensive literature, e.g., [1, 9, 12, 19, 22, 26] and the references therein on the dynamics of the rational function generated by iterative methods.

In this paper, using weight function on a variant of the method (1.2), we propose an optimal fourth order method and a family of sixth order methods to find a simple root of a nonlinear equation. Also, we extend the method (1.2) to find a multiple root of a nonlinear equation and obtain an optimal fourth order method. This is done in Section 2. In section 3, we check performance of the proposed methods with the help of some test functions and the results obtained are compared with some existing methods. Finally, in Section 4, dynamics of these methods are analyzed with the help of their basins of attraction.

2. DEVELOPMENT OF METHODS AND THEIR CONVERGENCE ANALYSIS

2.1. Fourth Order Method for Simple Roots. We propose a fourth order method by using weight function on a variant of Frontini-Sormani method (1.2). In fact, our proposed method is

$$(2.1) \quad \begin{aligned} y_n &= x_n - \frac{2}{3} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - H(t) \frac{f(x_n)}{f'(y_n)}, \end{aligned}$$

where, $H(t) = \frac{a_1 + a_2 t}{a_3 + t}$ is a weight function, a_1 , a_2 and a_3 are constants to be determined and $t = \frac{f'(y_n)}{f'(x_n)}$. We prove the following theorem:

Theorem 2.1. *Let f be a real or complex valued function defined on some interval I having sufficient number of smooth derivatives. Let α be a simple zero of the function f and the initial guess x_0 close enough to α . Then the method (2.1) has fourth order of convergence if $a_1 = -\frac{3}{2}$, $a_2 = \frac{5}{6}$ and $a_3 = -\frac{5}{3}$.*

Proof. Denote $c_j = \frac{f^{(j)}(\alpha)}{j! \cdot f'(\alpha)}$ and $e_n = x_n - \alpha$ is the error in x and $d_n = y_n - \alpha$ is the error in y at n th iteration. Taylor's series expansion of $f(x_n)$, and $f'(x_n)$ about α are given by:

$$(2.2) \quad f(x_n) = f'(\alpha) (e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + O(e_n^5))$$

and

$$(2.3) \quad f'(x_n) = f'(\alpha) (1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + O(e_n^4)).$$

Then, using (2.2) and (2.3) in first equation of (2.1), we obtain

$$d_n = \frac{1}{3} (e_n + 2c_2 e_n^2 - 4(c_2^2 - c_3) e_n^3 + 2(4c_2^3 - 7c_2 c_3 + 3c_4) e_n^4 + O(e_n^5)).$$

Therefore, Taylor's series expansion of $f'(y_n)$ about α is given by

$$\begin{aligned}
f'(y_n) &= f'(d_n + \alpha) \\
&= f'(\alpha) \left(1 + \frac{2}{3}c_2e_n + \frac{1}{3}(4c_2^2 + c_3)e_n^2 + \frac{4}{27}(-18c_2^3 + 27c_2c_3 + c_4)e_n^3 \right. \\
(2.4) \quad &\left. + \frac{1}{81}(432c_2^4 - 864c_2^2c_3 + 396c_2c_4 + 216c_3^2 + 5c_5)e_n^4 + O(e_n^5) \right).
\end{aligned}$$

Now, from (2.3) and (2.4), we get

$$\begin{aligned}
t &= \frac{f'(y_n)}{f'(x_n)}, \\
&= 1 - \frac{4}{3}c_2e_n + \frac{4}{3}(3c_2^2 - 2c_3)e_n^2 - \frac{8}{27}(36c_2^3 - 45c_2c_3 + 13c_4)e_n^3 \\
(2.5) \quad &+ \frac{4}{81}(540c_2^4 - 999c_2^2c_3 + 363c_2c_4 + 216c_3^2 - 100c_5)e_n^4 + O(e_n^5).
\end{aligned}$$

Using (2.5) on the weight function of (2.1), we obtained

$$\begin{aligned}
H(t) &= \frac{a_1 + a_2t}{a_3 + t} \\
&= \frac{a_1 + a_2}{1 + a_3} + \frac{4c_2(a_1 - a_2a_3)}{3(1 + a_3)^2}e_n - \frac{4(a_1 - a_2a_3)}{9(1 + a_3)^3}(c_2^2(5 + 9a_3) - 6c_3(1 + a_3))e_n^2 \\
&\quad + \frac{8(a_1 - a_2a_3)}{27(1 + a_3)^4}(4c_2^3(2 + 9a_3 + 9a_3^2) - 3c_3c_2(7 + 22a_3 + 15a_3^2) + 13c_4(1 + a_3)^2)e_n^3 \\
&\quad - \left(\frac{1 - a_2a_3}{81(1 + a_3)^5} \right) \left(16c_2^4(2 + 63a_3 + 180a_3^2 + 135a_3^3) - 36c_2^2c_3(15 + 109a_3 + 205a_3^2 + 111a_3^3) \right. \\
(2.6) \quad &\left. + 4c_2c_4(155 + 363a_3)(1 + a_3)^2 + (1 + a_3)^2(288c_3^2(1 + 3a_3) - 401c_5(1 + a_3)) \right) e_n^4 + O(e_n^5).
\end{aligned}$$

Therefore, in view of (2.2), (2.4) and (2.6), from (2.1), we obtain

$$\begin{aligned}
e_{n+1} &= \left(1 - \frac{a_1 + a_2}{1 + a_3} \right) e_n + \frac{((5a_1 + a_2) + a_3(a_1 - 3a_2))c_2}{3(1 + a_3)^2}e_n^2 \\
&\quad + \frac{2}{9(1 + a_3)^3} \left(c_2^2(15a_1 + 7a_2 + 6a_3(5a_1 + a_2) + a_3^2(7a_1 - 9a_2)) \right. \\
&\quad \left. - 3c_3(1 + a_3)(5a_1 + a_2 + a_3(a_1 - 3a_2)) \right) e_n^3 - \frac{1}{27(1 + a_3)^4} \left(4c_2^3(19a_1 + 22a_2 \right. \\
&\quad \left. + 96a_1a_3 + 69a_2a_3 + a_3^2(115a_1 + 36a_2) + a_3^3(22a_1 - 27a_2)) - 3c_2c_3(1 + a_3)(81a_1 + 41a_2 \right. \\
&\quad \left. + 6a_3(31a_1 + 7a_2) + a_3^2(41a_1 - 63a_2)) + c_4(1 + a_3)^2(127a_1 + 23a_2 + 23a_1a_3 - 81a_2a_3) \right) e_n^4 \\
(2.7) \quad &+ O(e_n^5).
\end{aligned}$$

In order to get fourth order of convergence, we must have

$$\begin{aligned}
1 + a_3 - a_1 - a_2 &= 0, \\
5a_1 + a_2 + a_3(a_1 - 3a_2) &= 0, \\
c_2^2(15a_1 + 7a_2 + 6a_3(5a_1 + a_2) + a_3^2(7a_1 - 9a_2)) - 3c_3(1 + a_3)(5a_1 + a_2 + a_3(a_1 - 3a_2)) &= 0,
\end{aligned}$$

which on solving gives

$$a_1 = -\frac{3}{2}, \quad a_2 = \frac{5}{6}, \quad a_3 = -\frac{5}{3}.$$

Substituting these values of the parameters in (2.7), the error equation of (2.1) is obtained as:

$$e_{n+1} = \frac{1}{9}(33c_2^3 - 9c_2c_3 + c_4)e_n^4 + O(e_n^5).$$

Hence, the assertion is proved. \square

2.2. Sixth Order Method for Simple Roots. In view of Theorem 2.1, our proposed fourth order method for finding simple roots of a nonlinear equation is given by

$$(2.8) \quad \begin{aligned} y_n &= x_n - \frac{2}{3} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \left(\frac{9 - 5 \frac{f'(y_n)}{f'(x_n)}}{10 - 6 \frac{f'(y_n)}{f'(x_n)}} \right) \frac{f(x_n)}{f'(y_n)}. \end{aligned}$$

It is an optimal method in the sense of Kung-Traub conjecture [18] and having the efficiency index 1.5874. Higher order methods can be obtained by using (2.8) in the Newton's component as below:

$$(2.9) \quad \begin{aligned} y_n &= x_n - \frac{2}{3} \frac{f(x_n)}{f'(x_n)}, \\ z_n &= x_n - \left(\frac{9 - 5 \frac{f'(y_n)}{f'(x_n)}}{10 - 6 \frac{f'(y_n)}{f'(x_n)}} \right) \frac{f(x_n)}{f'(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(z_n)}. \end{aligned}$$

It is eighth order method at the cost of five functions evaluation per iteration and its efficiency index is 1.5157. If we approximate $f'(z_n)$ in (2.9) by already known value(s) and apply some weight function, we obtain a new method. Precisely, we propose the following method:

$$(2.10) \quad \begin{aligned} y_n &= x_n - \frac{2}{3} \frac{f(x_n)}{f'(x_n)}, \\ z_n &= x_n - \left(\frac{9 - 5 \frac{f'(y_n)}{f'(x_n)}}{10 - 6 \frac{f'(y_n)}{f'(x_n)}} \right) \frac{f(x_n)}{f'(y_n)}, \\ x_{n+1} &= z_n - G(t) \frac{f(z_n)}{f'(x_n)} \end{aligned}$$

where, $G(t)$ is a weight function and t is as given in (2.1). We prove the following theorem for the convergence of (2.10):

Theorem 2.2. *Let f be a real or complex valued function defined on some interval I having sufficient number of smooth derivatives. Let α be a simple root of the equation $f(x) = 0$ and the initial guess x_0 is close enough to α . Then, the method (2.10) has sixth order of convergence if $G(0) = \frac{5}{2} + \frac{1}{2}G''(0)$ and $G'(0) = -\frac{3}{2} - G''(0)$ and $G''(0) \in \mathbb{R}$.*

Proof. Let, $\theta_n = z_n - \alpha$, then, in view of (2.2), (2.3) and (2.1), from second equation of (2.10), we obtain

$$\theta_n = \frac{1}{9} (33c_2^3 - 9c_2c_3 + c_4) e_n^4 + \left(-\frac{1}{9} 244c_2^4 + 24c_3c_2^2 - \frac{20c_4c_2}{9} - 2c_3^2 + \frac{31c_5}{108} \right) e_n^5 + O(e_n^6).$$

Therefore,

$$\begin{aligned} f(z_n) &= f(\theta_n + \alpha) \\ &= f'(\alpha) \left(\frac{1}{9} (33c_2^3 - 9c_3c_2 + c_4) e_n^4 - \frac{1}{108} (2928c_2^4 - 2592c_3c_2^2 + 240c_4c_2 + 216c_3^2 - 31c_5) e_n^5 \right. \\ (2.11) \quad &\left. + O(e_n^6) \right) \end{aligned}$$

The weight function $G(t)$ can be expand in Taylor's series about 0 as

$$\begin{aligned} G(t) &= G(0) + tG'(0) + \frac{t^2}{2}G''(0) + \dots, \\ (2.12) \quad &= b_1 + b_2t + b_3t^2 + \dots, \end{aligned}$$

where, for simplicity, $b_1 = G(0)$, $b_2 = G'(0)$ and $b_3 = \frac{1}{2}G''(0)$.

In view of (2.3),(2.5), (2.11) and (2.12), from (2.10), we obtain

$$\begin{aligned} e_{n+1} &= -\frac{1}{9} (33c_2^3 - 9c_3c_2 + c_4) (b_1 + b_2 + b_3 - 1) e_n^4 \\ &+ \frac{1}{108} \left(24c_2^4 (155b_1 + 177b_2 + 199b_3 - 122) - 72c_2^2c_3 (39b_1 + 41b_2 + 43b_3 - 36) \right. \\ &+ 8c_2c_4 (33b_1 + 35b_2 + 37b_3 - 30) + (216c_3^2 - 31c_5) (b_1 + b_2 + b_3 - 1) \left. \right) e_n^5 \\ &+ \frac{1}{328} \left(-24c_2^5 (2881b_1 + 3699b_2 + 4605b_3 - 1951) \right. \\ &+ 36c_2^3c_3 (2357b_1 + 2793b_2 + 3245b_3 - 1790) - 8c_2^2c_4 (1629b_1 + 1779b_2 + 1937b_3 - 1431) \\ &+ c_2 \left(c_5 (1293b_1 + 1417b_2 + 1541b_3 - 1107) - 108c_3^2 (171b_1 + 187b_2 + 203b_3 - 150) \right) \\ (2.13) \quad &\left. + 2 \left(6c_3c_4 (207b_1 + 215b_2 + 223b_3 - 198) - 83c_6 (b_1 + b_2 + b_3 - 1) \right) \right) e_n^6 + O(e_n^7) \end{aligned}$$

In order to get sixth order of convergence, we must have

$$b_1 + b_2 + b_3 - 1 = 0, \quad \text{and}$$

$$\begin{aligned} 24c_2^4 (155b_1 + 177b_2 + 199b_3 - 122) - 72c_2^2c_3 (39b_1 + 41b_2 + 43b_3 - 36) \\ + 8c_2c_4 (33b_1 + 35b_2 + 37b_3 - 30) + (216c_3^2 - 31c_5) = 0. \end{aligned}$$

On solving these equations, we obtain $b_1 = \frac{5}{2} + b_3$ and $b_2 = -\frac{3}{2} - 2b_3$. This implies that $G(0) = \frac{5}{2} + \frac{1}{2}G''(0)$ and $G'(0) = -\frac{3}{2} - G''(0)$ and $G''(0) \in \mathbb{R}$. With these values of b_1 and b_2 , from (2.13), the error equation of the method (2.10) is obtained as

$$e_{n+1} = -\frac{1}{81} (33c_2^3 - 9c_3c_2 + c_4) ((16b_3 - 54)c_2^2 + 9c_3) e_n^6 + O(e_n^7).$$

This completes the proof of the theorem. \square

In (2.10), if we take the weight function

$$G(t) = \left(\frac{5}{2} + \beta\right) - \left(\frac{3}{2} + 2\beta\right)t + \beta t^2,$$

where $t = \frac{f'(y)}{f'(x)}$ and $\beta = G''(0)$. Then, the family of the sixth order method is obtained as

$$(2.14) \quad \begin{aligned} y_n &= x_n - \frac{2}{3} \frac{f(x_n)}{f'(x_n)}, \\ z_n &= x_n - \left(\frac{9 - 5 \frac{f'(y_n)}{f'(x_n)}}{10 - 6 \frac{f'(y_n)}{f'(x_n)}} \right) \frac{f(x_n)}{f'(y_n)}, \\ x_{n+1} &= z_n - \left(\left(\frac{5}{2} + \beta \right) - \left(\frac{3}{2} + 2\beta \right) t + \beta t^2 \right) \frac{f(z_n)}{f'(x_n)}. \end{aligned}$$

Method (2.14) involves four functions evaluation per iteration. So, the efficiency index of the method is 1.5651, which is better than eighth order method (2.9).

2.3. Modification for multiple roots. Methods proposed in the previous subsection converge accordingly whenever α is a simple root of the equation $f(x) = 0$. On the other hand, if α is a multiple root, then these methods do not retain their order of convergence. For example, if we apply (2.8) to find the multiple root of the equation $f(x) = 0$ having multiplicity m , then, the error equation of the method is obtained as

$$e_{n+1} = \left(1 - \frac{10\mu + m(-9 + 5\mu)}{2m(6\mu + m(-5 + 3\mu))} \right) e_n + O(e_n^2), \quad \mu = \left(\frac{m}{m+2} \right)^m$$

which shows the linear convergence of the method (2.8) for multiple roots.

In this subsection, we modify the method (2.8) to find the multiple roots of any equation with known multiplicity m . In fact, we propose the following:

$$(2.15) \quad \begin{aligned} y_n &= x_n - \frac{2m}{m+2} \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \left(\frac{p_1 - p_2 \frac{f'(y_n)}{f'(x_n)}}{p_3 - \frac{f'(y_n)}{f'(x_n)}} \right) \frac{f(x_n)}{f'(y_n)}. \end{aligned}$$

where, p_1 , p_2 and p_3 are constants to be determined. We prove the following theorem:

Theorem 2.3. *Let f be a real or complex valued function defined on some interval I having sufficient number of smooth derivatives. Let α be a multiple root with multiplicity $m \geq 1$ of the equation $f = 0$ and the initial guess x_0 is close enough to α . Then, the method (2.15) has fourth order of convergence if*

$$\begin{aligned} p_1 &= -\frac{\mu^2(m-2)(m+2)^3}{2m^2}, \\ p_2 &= -\frac{\mu(16 - 16m - 8m^2 + 2m^3 + m^4)}{2m^2}, \\ p_3 &= \frac{\mu(8 - 4m + m^3)}{m^3}. \end{aligned}$$

Proof. If α is a multiple root with multiplicity m of the equation $f(x) = 0$, then

$$f(\alpha) = f'(\alpha) = \dots = f^{m-1}(\alpha) = 0 \quad \text{and} \quad f^m(\alpha) \neq 0.$$

Denote $k_j = \frac{m!}{(m+j)!} \frac{f^{(m+j)}(\alpha)}{f^{(m)}(\alpha)}$, $j = 1, 2, \dots$ and expand $f(x_n)$ and $f'(x_n)$ in Taylor's series about α , we obtain:

$$(2.16) \quad f(x_n) = \frac{f^m(\alpha)}{m!} e_n^m (1 + k_1 e_n + k_2 e_n^2 + k_3 e_n^3 + k_4 e_n^4 + O(e_n^5)).$$

and

$$(2.17) \quad f'(x_n) = \frac{f^m(\alpha)}{m!} e_n^{m-1} (m + (m+1)k_1 e_n + (m+2)k_2 e_n^2 + (m+3)k_3 e_n^3 + (m+4)k_4 e_n^4 + O(e_n^5)).$$

Using (2.16) and (2.17) in first equation of (2.15), we obtain:

$$\begin{aligned} d_n = & \frac{m}{m+2} e_n + \frac{2k_1}{m(m+2)} e_n^2 - \frac{2(k_1^2(m+1) - 2k_2 m)}{m^2(m+2)} e_n^3 \\ & + \frac{2(3k_3 m^2 + k_1^3(m+1)^2 - k_2 k_1 m(3m+4))}{m^3(m+2)} e_n^4 + O(e_n^5). \end{aligned}$$

Therefore, Taylor's series expansion of $f'(y_n)$ about α is given by

$$\begin{aligned} f'(y_n) = & f'(d_n + \alpha) \\ = & f'(\alpha) \mu \left[\frac{m+2}{m! e_n} + \frac{k_1(m^3 + 3m^2 + 2m - 4)}{m^2 m!} + \frac{1}{m^4 m!} (k_2 m^2 (m^3 + 4m^2 + 4m - 8) \right. \\ & - 4k_1^2 (m-2) e_n + \frac{1}{3m^6 (m+2)^2 m!} \left(-12k_1 k_2 (m+2)^2 (m^2 + 4m - 8) m^2 \right. \\ & + 4k_1^3 (m+2)^2 (m^4 + 5m^3 - 4m^2 + 4m - 12) \\ & \left. \left. + 3m^4 k_3 (m^5 + 9m^4 + 30m^3 + 36m^2 - 24m - 48) \right) e_n^2 + O(e_n^3) \right]. \end{aligned} \quad (2.18)$$

In view of (2.16), (2.17) and (2.18), we obtain from (2.15)

$$(2.19) \quad e_{n+1} = \frac{1}{\mu(m+2)(\mu(m+2) - p_3 m)} \left[A_1 e_n + A_2 e_n^2 + A_3 e_n^3 + A_4 e_n^4 \right] + O(e_n^5)$$

where

$$\begin{aligned}
A_1 &= p_1 m + (m+2)\mu((m+2)\mu - p_2 - p_3 m) \\
A_2 &= \frac{k_1}{m^2(m+2)(\mu(m+2) - p_3 m)} \left(p_2(m+2)^2 \mu(\mu(m^2 + 2m - 4) - p_3 m^2) \right. \\
&\quad \left. + p_1 m(p_3 m(m^2 + 2m - 4) - \mu(m^3 + 4m^2 - 4m - 16)) \right) \\
A_3 &= \frac{1}{m^3((m+2)(\mu(m+2) - p_3 m))^2} \left[\left(-p_2 \mu(m+2)^3 (p_3^2 m^3(m+1) - 2p_3 \mu(m^4 + 3m^3 - 2m - 4) \right. \right. \\
&\quad \left. \left. + \mu^2(m^4 + 5m^3 + 4m^2 - 8m - 16)) + p_1(p_3^2 m^3(m^4 + 5m^3 + 4m^2 - 8m - 16) \right. \right. \\
&\quad \left. \left. + \mu^2(m+2)^2(m^5 + 5m^4 - 20m^2 - 32m + 16) \right. \right. \\
&\quad \left. \left. - 2p_3 m^2 \mu(m^5 + 7m^4 + 12m^3 - 10m^2 - 52m - 48) \right) \right] k_1^2 \\
&\quad - 2m(m+2)(p_3 m - \mu(m+2))(p_2 \mu(m+2)^2(\mu(m^2 + 2m - 4) - p_3 m^2) \\
&\quad \left. + p_1 m(p_3 m(m^2 + 2m - 4) - \mu(m^3 + 4m^2 - 4m - 16)) \right) k_2 \Big]
\end{aligned}$$

and

$$\begin{aligned}
A_4 &= \frac{1}{3m^5((m+2)(\mu(m+2) - p_3 m))^3} \left[\left(-p_2 \mu(m+2)^4 (3p_3^3 m^5(m+1)^2 \right. \right. \\
&\quad \left. \left. - p_3^2 m \mu(9m^6 + 36m^5 + 37m^4 + 2m^3 - 52m^2 - 32m - 48) \right. \right. \\
&\quad \left. \left. + p_3 m \mu^2(9m^6 + 54m^5 + 101m^4 + 44m^3 - 132m^2 - 176m - 176) \right. \right. \\
&\quad \left. \left. - \mu^3(3m^7 + 24m^6 + 67m^5 + 66m^4 - 64m^3 - 184m^2 - 144m - 32) \right) \right. \\
&\quad \left. + p_1 m(p_3^3 m^3(3m^7 + 24m^6 + 67m^5 + 66m^4 - 64m^3 - 184m^2 - 144m - 32) \right. \\
&\quad \left. - \mu^3(m+2)^3(3m^7 + 24m^6 + 59m^5 + 18m^4 - 212m^3 - 296m^2 - 48m + 224) \right. \\
&\quad \left. + p_3 \mu^2(m+2)^2(9m^8 + 72m^7 + 185m^6 + 102m^5 - 488m^4 - 872m^3 - 480m^2 + 128m + 192) \right. \\
&\quad \left. - p_3^2 m^2 \mu(9m^8 + 90m^7 + 337m^6 + 536m^5 - 40m^4 - 1440m^3 - 2096m^2 - 1280m - 256) \right) \Big] k_1^3 \\
&\quad + 3m^2(m+2)(p_3 m - \mu(m+2)) \left(p_2 \mu(m+2)^3 (p_3^2 m^3(3m+4) \right. \\
&\quad \left. - 2p_3 \mu(3m^4 + 10m^3 + 4m^2 - 8m - 16) + \mu^2(3m^4 + 16m^3 + 20m^2 - 16m - 64) \right) \\
&\quad - p_1 \left(p_3^2 m^3(3m^4 + 16m^3 + 20m^2 - 16m - 64) + \mu^2(m+2)^2(3m^5 + 16m^4 + 12m^3 \right. \\
&\quad \left. - 48m^2 - 128m + 64) - 2p_3 m^2 \mu(3m^5 + 22m^4 + 48m^3 - 160m - 192) \right) \Big] k_1 k_2 \\
&\quad - 3m^3(p_3 m - \mu(m+2))^2 \left(-p_2 \mu(m+2)^2 (\mu(3m^4 + 18m^3 + 28m^2 - 24m - 48) \right. \\
&\quad \left. - 3p_3 m^2(m+2)^2) - p_1 m(p_3 m(3m^4 + 18m^3 + 28m^2 - 24m - 48) \right. \\
&\quad \left. - \mu(3m^5 + 24m^4 + 56m^3 - 32m^2 - 240m - 192) \right) \Big] k_3.
\end{aligned}$$

The method (2.15) is of fourth order if in (2.19)

$$A_1 = 0, \quad A_2 = 0 \quad \text{and} \quad A_3 = 0$$

which on solving, gives the desired values of p_1 , p_2 and p_3 . Also, with these values of $A_i, i = 1, 2, 3$, (2.19) gives the following error equation of the method (2.15):

$$e_{n+1} = \frac{1}{3m^4(m+2)^2(m^2+2m-4)} \left(k_1^3 (m^7 + 8m^6 + 22m^5 + 18m^4 - 28m^3 - 40m^2 - 16m - 64) \right. \\ \left. - 3k_1k_2m^3(m+2)^2(m^2+2m-4) + 3k_3m^5(m^2+2m-4) \right) e_n^4 + O(e_n^5).$$

Hence, the assertion is proved. \square

3. NUMERICAL EXAMPLES

In this section, theoretical results proved in the previous section are verified practically. We check the performance of all the proposed methods on some test functions and compare the various results with some known methods.

We test numerically the order of convergence of the proposed methods by using *Computational Order of Convergence* (or COC) defined by Grau-Sanchez et al. [15]. They define COC of a sequence $\{x_n\}$ by

$$(3.1) \quad \text{COC} = \frac{\log \left| \frac{\tilde{e}_n}{\tilde{e}_{n-1}} \right|}{\log \left| \frac{\tilde{e}_{n-1}}{\tilde{e}_{n-2}} \right|}, \quad n = 2, 3, \dots$$

where $\tilde{e}_n = x_n - x_{n-1}$. The use of COC, as given by (3.1) serves as a practical check on the theoretical error calculations.

For simple roots, we take following test functions $f_i(x), i = 1, 2$ with initial guesses x_0 :

$$f_1(x) = \sin^2 x - x^2 + 1, \quad x_0 = 2, \\ f_2(x) = x^2 - e^x - 3x + 2, \quad x_0 = 1$$

For multiple roots, the test functions $g_i(x), i = 1, 2$, multiplicity m and initial guesses x_0 are taken as

$$g_1(x) = (\sqrt{x} + \log x - 5)^4, \quad m = 4, \quad x_0 = 1, \\ g_2(x) = \left(e^{x^2+7x-30} - 1 \right)^6, \quad m = 6, \quad x_0 = 3.2$$

For simple roots, we compare the results obtained from fourth order method (2.8), which is denoted by M_{41} , with the two known methods (1.3) and (1.4), which are denoted by M_{42} and M_{43} respectively. Table 1 shows the results obtained from these methods. Similarly, the results obtained from the sixth order method (2.14) with $\beta = 0$, denoted by M_{61} , are compared with the two known methods (1.5) denoted by M_{62} for $a = 2$ and (1.6) which is denoted by and M_{63} . Table 2 shows the results obtained from sixth order methods. Finally, in Table 3, the results for multiple roots are presented. For this, we take our proposed fourth order method (2.15), which is denoted by M_1 , and the two known methods (1.7) and (1.8), which are denoted by M_2 and M_3 respectively. In each case, we

compare the results obtained from first four iterations. From the tables, we observe that the results obtained from proposed methods are competitive with other known methods of the same order. But the results obtained from M_1 are much better than its two counterparts, viz., M_2 and M_3 for all the three examples.

In each of the table, $aE(-b)$ stands for $a \times 10^{-b}$. We have done our calculations with several number of significant digits (upto 1500 significant digits) to minimize the round off error using Mathematica 9. Further, due to limitations, we show the results up to 14 decimal places for the approximate root x_n .

Methods	Iterations	Test functions ($f_i(x)$)							
		f_1				f_2			
		x_n	$ f(x_n) $	$d = x_n - x_{n-1} $	COC	x_n	$ f(x_n) $	$d = x_n - x_{n-1} $	COC
M_{41}	1	1.42854602264018	0.0608	0.5715		0.257721519154245	0.0007	0.7423	
	2	1.40449214651858	1.237 E(-6)	0.0241		0.257530285439860	1.984 E(-17)	0.0002	
	3	1.40449164821534	2.573 E(-25)	4.983 E(-7)	3.99	0.257530285439860	1.128 E(-71)	5.251 E(-18)	4
	4	1.40449164821534	4.818 E(-100)	1.037 E(-25)		0.257530285439860	1.177 E(-288)	2.985 E(-72)	
M_{42}	1	1.34651146339548	0.1374	0.6535		0.028008046701890	0.8884	0.972	
	2	1.40448821376536	8.526 E(-6)	0.058		0.258206488835457	0.0026	0.2302	
	3	1.40449164821534	2.86 E(-23)	3.43 E(-6)	4.13	0.257530285439862	4.931 E(-15)	0.0007	4.63
	4	1.40449164821534	3.624 E(-93)	1.152 E(-23)		0.257530285439860	7.578 E(-62)	1.305 E(-15)	
M_{43}	1	1.42531031266043	0.0525	0.5747		0.257718325301160	0.0007	0.7423	
	2	1.40449182932916	4.496 E(-7)	0.0208		0.257530285439860	2.369 E(-17)	0.0002	
	3	1.40449164821534	2.777 E(-27)	1.811 E(-7)	3.99	0.257530285439860	2.926 E(-71)	6.269 E(-18)	4
	4	1.40449164821534	4.044 E(-108)	1.119 E(-27)		0.257530285439860	6.81 E(-287)	7.743 E(-72)	

TABLE 1. Numerical Results of fourth order methods for simple roots

Methods	Iterations	Test functions ($f_i(x)$)							
		f_1				f_2			
		x_n	$ f(x_n) $	$d = x_n - x_{n-1} $	COC	x_n	$ f(x_n) $	$d = x_n - x_{n-1} $	COC
M_{61}	1	1.42854602264018	0.0608	0.5888		0.257721519154245	0.0007	0.7425	
	2	1.40449165158474	8.364 E(-9)	0.0067		0.257530285439860	9.277E(-23)	8.893E(-6)	
	3	1.40449164821534	3.548 E(-49)	5.4E(-13)	5.99	0.257530285439860	9.93E(-143)	9.046E(-36)	5.99
	4	1.40449164821534	2.107E(-291)	1.499E(-73)		0.257530285439860	1.499E(-862)	1.003E(-215)	
M_{62}	1	1.40872501636896	0.0105	0.5913		0.257557067100725	0.0001	0.7424	
	2	1.40449164821535	2.071E(-14)	0.0042		0.257530285439860	2.937E(-32)	0.00002	
	3	1.40449164821534	1.241E(-84)	8.343E(-15)	5.99	0.257530285439860	1.753E(-197)	7.772E(-33)	6
	4	1.40449164821534	5.76E(-506)	5.001E(-85)		0.257530285439860	7.934E(-1189)	4.64E(-198)	
M_{63}	1	1.40801383147755	0.0088	0.592		0.257556453236163	0.0001	0.7424	
	2	1.40449164821534	7.278E(-15)	0.0035		0.257530285439860	1.386E(-31)	0.00002	
	3	1.40449164821534	2.470E(-87)	2.932E(-15)	5.99	0.257530285439860	1.049E(-192)	3.667E(-32)	6
	4	1.40449164821534	3.779E(-522)	9.952E(-88)		0.257530285439860	1.976E(-1159)	2.776E(-193)	

TABLE 2. Numerical Results of sixth order methods for simple roots

Methods	Iterations	Test functions ($g_i(x)$)							
		g_1				g_2			
		x_n	$ g(x_n) $	$d = x_n - x_{n-1} $	COC	x_n	$ g(x_n) $	$d = x_n - x_{n-1} $	COC
M_1	1	8.34283806280616	9.226E(-9)	7.343		3.072501442580200	15.556	0.1275	
	2	8.30943269423444	5.078E(-49)	0.0334		3.004662021752300	5.963E(-8)	0.0678	
	3	8.30943269423157	4.631E(-210)	2.873E(-12)	4	3.000000187660250	2.108E(-34)	0.0047	3.78
	4	8.30943269423157	3.202E(-854)	1.579E(-52)		3.000000000000000	1.020E(-139)	1.877E(-7)	
M_2	1	5.22224727313623	1.271	4.222		3.096465184959970	266.804	0.1035	
	2	8.25518500344720	6.512E(-8)	3.033		3.020404720495330	0.0008	0.0761	
	3	8.30943268977097	2.95E(-36)	0.0542	4.05	3.000184593232500	1.924E(-16)	0.0202	3.54
	4	8.30943269423157	1.28E(-149)	4.461E(-9)		3.000000000002110	4.297E(-64)	0.0002	
M_3	1	7.91111508454778	2.009E(-4)	6.911		3.073793036880630	18.355	0.1262	
	2	8.30943259155894	8.28E(-31)	0.3983		3.005102578449770	1.043E(-7)	0.0687	
	3	8.30943269423157	1.943E(-136)	1.027	4.01	3.000000276592830	2.161E(-33)	0.0051	3.78
	4	8.30943269423157	5.89E(-559)	4.019		3.000000000000000	1.36E(-135)	2.766E(-7)	

TABLE 3. Numerical Results of fourth order method for multiple roots

4. BASINS OF ATTRACTION

The basins of attraction of a rational function is the union of sets of initial values which on successive iteration converge to the attracting fixed points of the function. Generally, if we are close enough to one of the fixed point, then we should be in the basin of attraction of that point. But, near the boundary of the basin of attraction of different fixed points, we may have the chaotic region depending upon the function. Whenever an iterative method is applied to a polynomial, a rational operator R is obtained. The fixed points of such an operator are the roots of the polynomial as well as others. The fixed points which are not the roots of the polynomial, are called extraneous or strange fixed points. The roots of the involved polynomial are always the (super) attracting fixed points where as extraneous fixed points may be attractive, repulsive or neutral. The existence of extraneous fixed points causes the chaotic basin of attraction of any rational operator [26].

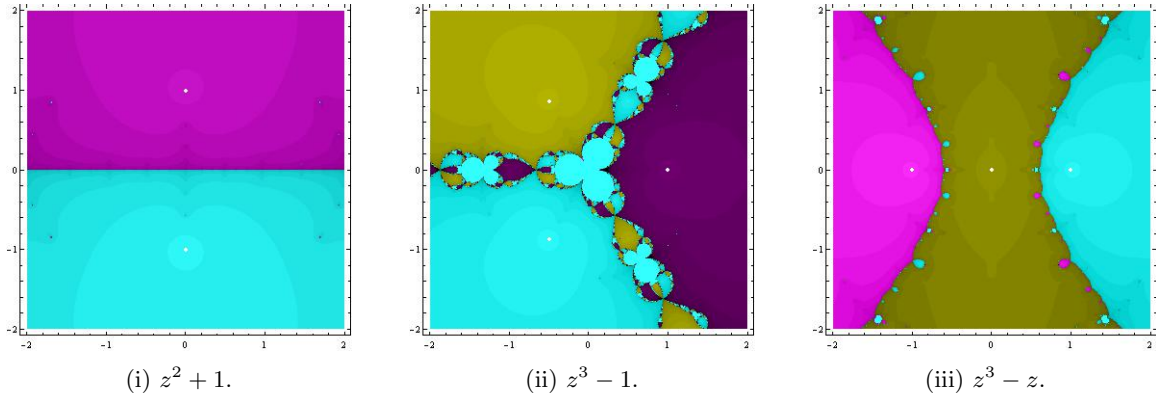
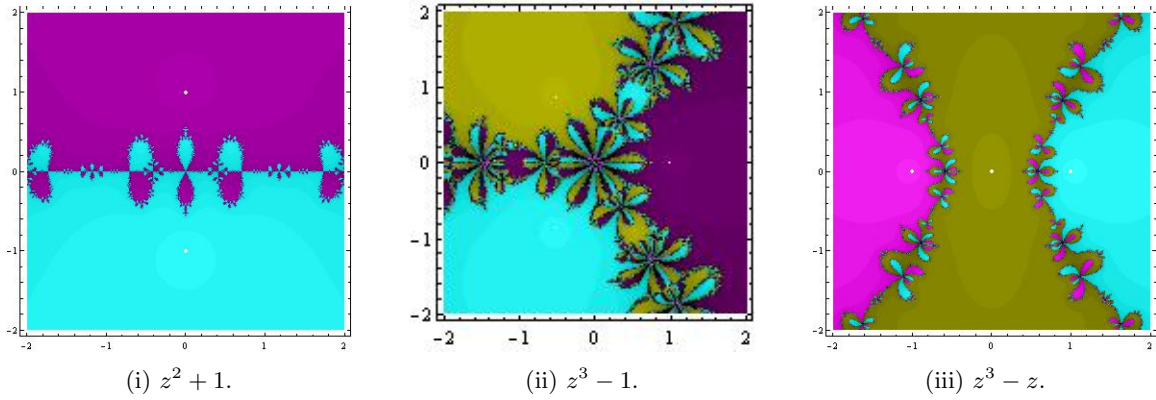
Formally, if z_0 is an attracting fixed point of the rational operator R , then the basin of attraction of z_0 is the set

$$\mathbb{B}(z_0) = \{z \in \hat{\mathbb{C}} : R^n(z) \rightarrow z_0 \text{ as } n \rightarrow \infty\},$$

where, $\hat{\mathbb{C}}$ is the extended complex plane, *i.e.* $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$.

The basins of attraction of an iterative method gives information about the stability of the method. Wider and smoother the basins, more stable is the method. On the other hand, a method having basins with chaotic dynamics with greater irregularity in it, is the less stable. For a comprehensive study of dynamics of rational functions, the interested readers can follow [3], [5], [6], [13] and references therein.

The basins of attraction are generated by Mathematica 9 using the idea of Varona [25]. We have used a mesh of 400×400 points in the region $[-2, 2] \times [-2, 2]$ of the complex plane. Different colors represent the basins of attraction of different roots of the polynomial involved. The roots are represented by white spots. Black region shows the no convergence

FIGURE 1. Basins of attraction of various fixed points of M_{41} .FIGURE 2. Basins of attraction of various fixed points of M_{61} .

of the initial guess within the maximum number of iterations or whose orbit converges to another thing (strange fixed points, cycles, etc.) or diverges. We work with a tolerance of 10^3 and a maximum number of 40 iterations.

To understand the dynamical behavior of the methods visually, the basins of attraction of the methods $M_{41}(z)$ and $M_{61}(z)$ for the polynomials $p(z) = z^2 + 1$, $p(z) = z^3 - 1$ and $p(z) = z^3 - z$ are presented in Figure 1 and 2. From the figures, it can be observed that the basin of attraction becomes more chaotic as the order of the method increases for any polynomial. In other words, the basins of attraction of the method M_{61} is more chaotic than that of M_{41} for any particular polynomial. Consequently, M_{61} is less stable than the method M_{41} .

For multiple roots, we apply the method $M_1(z)$ to the polynomials $p(z) = (z^2 + 1)^2$, $p(z) = (z^3 - 1)^2$, $p(z) = (z^2 + 1)^4$ and $p(z) = (z^3 - 1)^4$. The corresponding basins of attraction are presented in Figure 3. From the figure, it is clear that as the multiplicity of the polynomial increases, the basins of attraction becomes more chaotic. Therefore, the method (2.15) is more appropriate for finding the zeros of the functions having lower multiplicity.

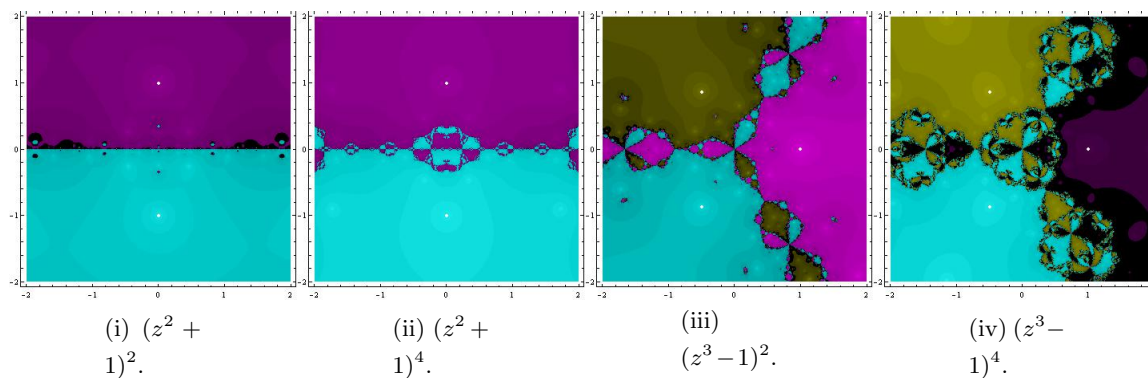


FIGURE 3. Basins of attraction of various fixed points of M_1 .

5. CONCLUSION

We have developed three higher order iterative methods for solving nonlinear equations using weight functions on the variants of cubically convergent Frontini-Sormani method. Two methods having order of convergence four and six are for computing simple root and a fourth order method to compute multiple root of a nonlinear equation. Both the fourth order methods are optimal and the sixth order method is non-optimal as it requires four functions evaluation per iteration. The numerical results obtained from the proposed methods are found to be competitive with existing methods. The dynamics of the proposed methods is discussed with the help of basins of attraction of the fixed points and stability of the methods is analyzed.

Further extension: All the methods developed in this paper can be extended for multivariate function in which first order derivative is replaced by an equivalent Jacobian matrix.

Acknowledgement: The author would like to express deep gratitude to Prof. Pankaj Jain for his patient guidance, enthusiastic encouragement and useful critiques for this research work.

REFERENCES

- [1] S. Amat, S. Busquier, S. Plaza, Review of some iterative root finding methods from a dynamical point of view, *Scientia*, 10, 3-35, 2004.
- [2] D.K.R. Babajee, Several improvements of the 2-point third order midpoint iterative method using weight functions, *Appl. Math. Comput.*, 218, 7958-7966, 2012.
- [3] A.F. Beardon, Iteration of Rational Functions, *Graduate Text in Mathematics*, Springer-Verlag, New York, 1991.
- [4] R. Behl, A. Cordero, S.S. Motsa, J.R. Torregrosa, V. Kanwar, An optimal fourth-order family of methods for multiple roots and its dynamics, *Numer. Algor.*, 71(4), 775-796, 2016.
- [5] P. Blanchard, The dynamics of Newton's method, *Proc. Sympos. Appl. Math.*, 49, 139-154, 1994.
- [6] L. Carleson, T. W. Gamelin, Complex Dynamics, *Universitext*, Springer Verlag, 1992.
- [7] P.B. Chand, F.I. Chicharro, N. Garrido, P. Jain, Design and complex dynamics of Potra-Pták type optimal methods for solving nonlinear equations and its applications, *Mathematics*, 7(10), 942, 2019.

- [8] P.B. Chand, F.I. Chicharro, P. Jain, K Sethi, Optimal fourth order Weerakoon-Fernando type methods for multiple roots and their dynamics, *Mediterr. J. Math.*, 16(3), 67, 2019.
- [9] F.I. Chicharro, A. Cordero, J.M. Gutiérrez, J.R. Torregrosa, Complex dynamics of derivative free methods for nonlinear equations. *Appl. Math. Comput.*, 219, 7023-7035, 2013.
- [10] C. Chun, B. Neta, J. Kozdon, M. Scott, Choosing weight functions in iterative methods for simple roots, *Appl. Math. Comput.*, 227, 788-800, 2014.
- [11] C. Chun, B. Neta, A new sixth-order scheme for nonlinear equations, *Appl. Math. Letters*, 25, 185-189, 2012.
- [12] A. Cordero, L. Feng, A.A. Magrenan, J.R. Torregrosa, A new fourth order family for solving nonlinear problems and its dynamics, *J. Math. Chem.*, 53, 893-910, 2015.
- [13] R. L. Devaney, An Introduction to Chaotic Dynamical Systems, *Addison-Wesley publishing Company*, New York, 1989.
- [14] M. Frontini, E. Sormani, Some variant of Newton's method with third order convergence, *Appl. Math. Comp.*, 140, 419-426. 2003.
- [15] M. Grau-Sánchez, M. Noguera, J.M. Gutiérrez, On some computational orders of convergence, *Appl. Math. Lett.*, 23(4), 472-478, 2010.
- [16] S. Li, X. Liao, L. Cheng, A new fourth-order iterative method for finding multiple roots of nonlinear equations, *Appl. Math. Comp.*, 215(3), 1288-1292, 2009.
- [17] B. Liu, X. Zhou, A new family of fourth order methods for multiple roots of nonlinear equations, *Nonlinear Anal. Model. Control*, 18(2), 143-152, 2013.
- [18] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iterations, *J. Assoc. Comput. Mach.*, 21, 643-651, 1974.
- [19] B. Neta, C. Chun, Basins of attraction for several methods to find the simple roots of nonlinear equations, *Appl. Math. Comput.*, 218(21), 10548-10556, 2012.
- [20] A.M. Ostrowski, Solutions of Equations and System of Equations, *Academic Press*, New York, 1960.
- [21] M. Sharifi, D.K.R. Babajee, F. Soleymani, Finding solutions of nonlinear equations by a class of optimal methods, *Comput. Math. Appl.*, 63, 764-774, 2012.
- [22] J.R. Sharma, R. Sharma, Modified Jarratt method for computing multiple roots, *Appl. Math. Comp.*, 217, 878-881, 2010.
- [23] J. R. Sharma, R.K. Guha, A family of modified Ostrowski's methods with accelerated sixth order convergence, *Appl. Math. Comput.*, 190, 111-115, 2007.
- [24] J.F. Traub, Iterative Methods for the Solution of Equations, *Chelsea Publishing company*, New York, 1977.
- [25] J.L. Varona, Graphic and numerical comparison between iterative methods, *Math. Intell.*, 24, 37-46, 2002.
- [26] E.R. Vrscay, W.J. Gilbert, Extraneous fixed points, basin boundaries and chaotic dynamics for Schroder and Konig rational iteration functions, *Numer. Math.*, 52, 1-16, 1988.