Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online)
Vol.7, No.10, 2017

Some Multiple and Simple Real Root Finding Methods

Tekle Gemechu
Department of mathematics, Adama Science and Technology University, Ethiopia

Abstract
Solving nonlinear equations with root finding is very common in science and engineering models. In particular, one applies it in mathematics, physics, electrical engineering and mechanical engineering. It is a researchable area in numerical analysis. This present work focuses on some iterative methods of higher order for multiple roots. New and existing novel multiple and simple root finding techniques are discussed. Methods independent of a multiplicity m of a root r, which function very well for both simple and multiple roots, are also presented. Error-correction and variatonal technique with some function estimations are used for the constructions. For the analysis of orders of convergence, some basic theorems are applied. Ample test examples are provided (in C++) for test of efficiencies with suitable initial guesses. And convergence of some methods to a root is shown graphically using matlab applications.
Keywords:Iterative algorithms, error-correction, variational methods, multiple roots, applications

1. Introduction

Nonlinear equations appear in most science and engineering problems. For example, in climate models, electric circuit analysis, analysis of state equations for the general gas law, mechanical motions and in many other processes (Alfio et al., 2000). A practical problem of application of multiple roots is finding multiple Eigen values of square matrices (see test equation f_1 in unit-6).

The root finding algorithms are usually constructed by the methods of interpolations (linear, quadratic, cubic, etc) (Tekle, 2017& 2014), perturbation method (Pakdemirli, 2008), variational techniques (Muhammad et al.; 2011), the Newton's method approach, fixed point techniques, etc (Germund and Ake, 2008; Jurgen,1994; Anthony and Philip, 1978). Some iterative methods for multiple roots can be referred to the works of (Muhammad et al.;2011;; YI Jin and Bahman,2010 ;Janak and Rajni, 2012;J.Stoer and Bulirsch, 1993), where the modified Newton method and some higher order convergent methods are accessible. Solving multiple roots with iterative methods may have some difficulties. The basic root finding algorithms such as the Secant method, Bisection method, Regula Falsi method, Newton's method, fixed point iteration, Muller's method and many others are not effective for multiple roots. Some of them need modifications to apply for multiple roots. For example, Richard (1977) used a procedure that involves the divided difference to apply the Secant method for multiple roots.

The Modified Newton method for multiple roots is obtained assuming that $x-r$ is a factor of $f(x)$ with a multiplicity m and then applying linearization or Newton's method for simple root r of $f(x)$ (Alfio et al., 2000; J. Stoer and R. Bulirsch, 1993). Chebyshev's method, Euler's method, Halley's method, Osada's method and Chun-Neta (CNM) method are some well known for multiple roots (Muhammad et al., 2011;Janak and Rajni, 2012; Chungbum&Beny Neta,2009). The variational technique was used to generate novel third order algorithms for multiple roots based on modified Newton's method (Muhammad et al., 2011). In this paper, our present study, we start with linearization technique to find linear convergent methods for multiple roots (Alfio et al.,2000; J.Stoer and R.Bulirsch,1993). And deriving for the [error-correction] $h=x-r$ (for both simple and multiple roots) from the derivative (s) of the assumed expression of $f(x)$, we obtain some new modified methods of order at least two for multiple roots. We also obtain methods that are independent of multiplicity m of a root r of $f(x)$. We develop third order methods for multiple roots based on some ideas in variational methods (Muhammad et al.,2011) with additional concepts. With an appropriate initial input, most of the new algorithms are better competent (in terms of speed of convergence) with some existing methods for multiple roots. The motivation comes from the author's previous work on simple roots (Tekle, 2017). For related concepts refer to (Farooketal., 2016; Farook & Muhammad ,2014; Chungbum&Beny Neta, 2009). For applications of multiple roots, see: https://www.e-education.psu.edu/png520/m10_p2.html (PNG 520: Phase Relations in Reservoir Engineering, in a cubic equation of state).

1.2. Discussions on some Existing Multiple Root Finding

Consider the basic modified Newton's iterative formula

$$x_{k+1}=x_k-m\frac{f(x_k)}{f'(x_k)};k=0,1,2,\dots,n. \tag{1}$$

to numerically approximate a root r of multiplicity m of a nonlinear equation $f(x) = 0$ (Alfio et al., 2000; J. Stoer & R.Bulirsch, 1993). Where $f(x) = (x-r)^m g(x)$ for some g(x). And

$$f' = m(x-r)^m g(x)/(x-r) + (x-r)^m g'(x), g(r) \neq 0. \tag{2}$$

Assuming $g'(x) \approx 0$ and applying Newton's method to solve $f(x) = 0$, we have (1) which is quadratic convergent (Anthon &Philip, 1978). Some basic cubic convergent methods are Chun-Neta method (3), Halley's method (4), Osada's method (5), Euler-Chebyshev's method (5b) and the fourth order (6) are as listed below (Muhammad et al.,2011; YI Jin & Bahman,2010;Janak&Rajni,2012). See also (Tekle, 2014).

$$\phi(x) = x - \frac{2m^2[f(x)]^2 f''(x)}{m(3-m)f(x)f'(x)f''(x) + (m-1)^2[f'(x)]^3}. \tag{3}$$

$$\phi(x_k) = x_{k+1} = x_k - \frac{2mf(x_k)f'(x_k)}{(m+1)[f'(x_k)]^2 - mf(x_k)f''(x_k)}. \tag{4}$$

$$\phi(x_k) = x_{k+1} = x_k - 0.5m(m+1)\frac{f(x_k)}{f'(x_k)} + 0.5(m-1)^2 \frac{f'(x_k)}{f''(x_k)}. \tag{5}$$

$$\phi(x_k) = x_{k+1} = x_k - 0.5m(3-m)\frac{f(x_k)}{f'(x_k)} - 0.5m^2 \frac{[f(x_k)]^2 f''(x_k)}{[f'(x_k)]^3}. \tag{5b}$$

$$\phi(x_k) = x_k - 3mf(x_k)\frac{(1+m)[f'(x_k)]^2 - mf(x_k)f''(x_k)}{(2m^2+3m+1)[f'(x_k)]^3 - 3m(1+m)f(x_k)f'(x_k)f''(x_k) + m^2[f(x_k)]^2 f'''(x_k)} \tag{6}$$

One of the novel third order methods developed using the variational technique by Noor, Farook& Noor is

$$\psi(x) = x - m\frac{f(x)}{f'(x)} - m/2[(1-m) + mf(x)f''(x)/[f'(x)]^2)]\frac{f(x)}{f'(x) - \beta f(x)} \tag{6b}$$

Note: if m = 1, then (3) and (5) become Newton's method for a single root r.
And (4) and (5b) become Halley's method and Chebyshev's method for simple roots respectively. One has to notice that (3) is obtained using

$\frac{ff''}{f'} = \frac{m-1}{m}[2 - \frac{(m-1)f'^2}{mff''}]$ in (4), (Chungbum&Beny Neta, 2009). With m = 1, (6) gives a fourth order (6c) below for simple roots. Methods in (6) & (6c) were presented by YI JIN & BAHMAN (2010).

$$\phi(x_k) = x_k - 3f(x_k)\frac{2[f'(x_k)]^2 - f(x_k)f''(x_k)}{6[f'(x_k)]^3 - 6f(x_k)f'(x_k)f''(x_k) + [f(x_k)]^2 f'''(x_k)} \tag{6c}$$

A quadratic convergent (6d), a cubic one (6e) &4th order methods therein by Munish et al. (2015) are

$$\phi(x_n) = x_{n+1} = x_n - 0.5[m\frac{f(x_n)}{f'(x_n)} - (1-m)\frac{f'(x_n)}{f''(x_n)}] \tag{6d}$$

$$\psi(x_n) = x_{n+1} = x_n - 0.5[m(m+1)\frac{f(x_n)}{f'(x_n)} - (1-m)^2 \frac{f'(x_n)}{f''(x_n)}] \tag{6e}$$

See also higher order methods by Rajinder Thukral (2013).

2. Construction of some Iterative Methods for Multiple Roots

If r is a root of multiplicity m of $f(x) = 0$, then using (2) above, the Newton's formula for multiple roots becomes (7b) as shown below. With linear estimation of f(x) around r,

$$h = (x-r) = f(x)/f'(x) = (x-r)\frac{g(x)}{[(x-r)g'(x) + mg(x)]}, \tag{7}$$

$$\phi(x) = x - (x-r)\frac{g(x)}{[(x-r)g'(x) + mg(x)]} \tag{7b}$$

(7b) is linear convergent with rate $\phi'(r) = 1 - 1/m > 0$ for all m > 1. And if m =1, then $\phi'(r) = 1 - 1/m = 0$ and it is quadratic convergent (Alfio et al., 2000). Some kind of replacements may be used as follows.

Suppose $h = x - r = -\frac{f}{f'}$ and we put $g \approx vf'$, then from (7b) we obtain

$$x_{k+1} = \phi(x_k) = x_k - \frac{f(x_k)f'(x_k)}{m[f'(x_k)]^2 - f''(x_k)f(x_k)}. \quad (8)$$

But if $h = x - r = -m\frac{f}{f'}$ and we use $g \approx vf'$ in (7b), then we can get

$$x_{k+1} = \phi(x_k) = x_k - \frac{f(x_k)f'(x_k)}{[f'(x_k)]^2 - f''(x_k)f(x_k)}. \quad (9)$$

If m = 1 in (8), then we also get (9), an existing method. Eqn. (9) is independent of m and holds well for both simple
and multiple roots. It can be obtained applying Newton's method for solving u(x) = f(x)/f'(x) (Germund&Ake, 2008; Anthony & Philip,1978). Assuming that the order of (7b) may be p >1 and solving the condition $\phi'(x) = 0$ if it holds, we can obtain

$$h = \frac{m(1-m)g}{hg'' + 2mg'}. \quad (9b)$$

Putting $g \approx bf$ and $h = -mf/f'$ in the right part of (9b), we have

$$x_{k+1} = \phi(x_k) = x_k - \frac{(m-1)f(x_k)f'(x_k)}{2[f'(x_k)]^2 - f(x_k)f''(x_k)}. \quad (9c)$$

Putting $g \approx cf$ and $h = -f/f'$ in the right part of (9b), we have

$$x_{k+1} = \phi(x_k) = x_k - \frac{m(m-1)f(x_k)f'(x_k)}{[2m[f'(x_k)]^2 - f(x_k)f''(x_k)]}. \quad (10)$$

If $f'' \approx (f')^2/f$ in (10) and setting m to be m+1 we have

$$x_{k+1} = \phi(x_k) = x_k - \frac{m(m+1)f(x_k)}{[2m+1]f'(x_k)}. \quad (11b)$$

Replacing m(m+1)/(2m+1) by 2m(m+1)/(2m+1) in (11b) gives the second order,

$$x_{k+1} = \phi(x_k) = x_k - \frac{2m(m+1)f(x_k)}{[2m+1]f'(x_k)}. \quad (11c)$$

An interesting aspect of an algorithm in (11c) is that it can be obtained from extended Halley's method in eqn.(4) with $f'' = m/(m+1)(f')^2/f$. And using $f'' = (m-1)/(m+1)(f')^2/f$ in eqn.(4) yields modified Newton's method in eqn.(1). Examples below show that Newton's Method (Nm) for simple roots and formulae (8), (10) and (11b) when applied here for multiple roots are slow convergent. Compare their efficiencies with (9) and (11c) which are relatively much better, based on the number of iterations required for convergence as in table-1 below.
We take

$f_1(x) = 2x^3 - 6x^2 + 6x^2 - 2 = 0$ with x_o =1.5, 2 and root r =1 in [1, 2), m = 3.

$f_2(x) = [3x - \cos x - 1]^2 = 0$ with x_o = 0, 1 and r ≈ 0.607102 in (0, 1), m = 2.

See table of results below.

Table-1: Comparisons of some methods for multiple roots (tol = 10^{-33})

$f(x)$	Nm	(8)	(9)	(10)	(11b)	(11c)
$f_1(x)$	35, 36	90, 95	1, 1	39, 41	35, 36	1, 1
$f_2(x)$	23, 19	39, 33	3, 3	61, 53	22, 19	3, 3

3. Some Second Order Methods for Multiple Roots

By deriving for the error-correction h from eqn.(2) above in different ways, we develop some iterative methods of order two. The aim is to show another way to modify the slow convergent methods in section-3 above and obtain better methods for multiple roots. Consider again the equation,

$$f' = m(x-r)^m g(x)/(x-r) + (x-r)^m g'(x). \tag{12}$$

$$\Rightarrow f' = mf(x)/(x-r) + f(x)g'(x)/g(x). \tag{13}$$

$$\Rightarrow \phi(x) = x - \frac{mfg(x)}{f'g(x) - fg'(x)} \tag{13a}$$

Eqn(13a) can also be obtained taking $F(x) = \varphi(x) + \lambda[f(x)/g(x)]^{1/m}$ and applying variational technique [as in eqn.(33b) below]. If $g' \approx kg$ in (13a) [there can be many cases], then one gets a multiple root finding

$$\phi(x) = x - \frac{mf(x)}{f'(x) - kf(x)} \tag{13b}$$

If $g' \approx cf/f'g$ in (13a), then we get

$$\psi(x) = x - \frac{mf(x)f'(x)}{f'(x)^2 - cf(x)^2} \tag{13c}$$

The method in (13b) was obtained also by (Muhammad et al., 2011). Let us assume that $g'(x) \neq 0$. That is $g(x)$ is not a constant. And suppose

$$f' \approx mf(x)/(x-r) + cf(x)g'(x) \tag{13d}$$

And using,

$$f = h^m g \text{ or } g = f/h^m, h = x - r. \tag{14}$$

$$(13d) \Rightarrow h = x - r = \frac{mf}{f' - cfg'}. \tag{15}$$

$$\text{Or } (13) \Rightarrow h = \frac{mf}{f'} + \frac{hfg'}{f'}. \tag{16}$$

If $g' \approx (k/c)f$ in (15), then we obtain the following iterative function for multiple roots.

$$\phi(x) = x - \frac{mf(x)}{f'(x) - k'[f(x)]^2}. \tag{17}$$

Taking k' = 0 in (17) gives the modified Newton's formula. And if k' = 0 but m = 1, then we get the Newton's formula for simple roots.

And inserting $k' \approx \dfrac{f''}{2ff'}$ and m = 1 in (17) yields the Halley's or Chebyshev's method for single roots.

For k' = (m-1)/(m+1) we obtain the method below for solving a multiple root.

$$x_{k+1} = \phi(x_k) = x_k - \frac{mf(x_k)}{f'(x_k) - \frac{m-1}{m+1}[f(x_k)]^2}. \tag{18}$$

Imposing the condition $\left|\dfrac{f(x_k)}{f'(x_k)}\right| < 1$ in (18), we have

$$x_{k+1} = \phi(x_k) \approx x_k - m\frac{f(x_k)}{f'(x_k)}[1+\frac{m-1}{m+1}\frac{[f(x_k)]^2}{f'(x_k)}]. \quad (19)$$

Using $h = -m\frac{f}{f'}$ and $g' = k\frac{f''}{2f'}$ in (16), we can obtain

$$x_{k+1} = \phi(x_k) = x_k - m\frac{f(x_k)}{f'(x_k)}[1-k\frac{f(x_k)f''(x_k)}{2[f'(x_k)]^2}]. \quad (20)$$

If m = 1= -k, then (20) becomes Chebyshev's method for simple roots. If k = 0, then (20) is modified Newton's method. And if k = 0 and m =1, it is the Newton method for simple roots.

Taking $f'' \approx [f']^2 / f$ and $k = \frac{-2(m-1)}{m^2}$ in (20), we obtain

$$x_{k+1} = \phi(x_k) = x_k - m[1+\frac{m-1}{m^2}]\frac{f(x_k)}{f'(x_k)}. \quad (21)$$

Now with $g' \approx k(1-ff''/[f']^2)g$ and $h = x - r$ used in (13) or in (13d) with c = 1, and $f = h^m g$ or $g = f/h^m$ we can get

$$h = m\frac{f(x)}{f'(x)} + kh[[(f'(x)]^2 - f(x)f''(x)]\frac{f(x)}{[f'(x)]^3}. \quad (22)$$

If $k \approx \frac{m-1}{m+1}\frac{[f'(x)]^4}{f(x)}$, $h = -f/f'$ then we have

$$x_{k+1} = \phi(x_k) = x_k - [m\frac{f(x_k)}{f'(x_k)} - \frac{m-1}{m+1}([f'(x_k)]^2 - f(x_k)f''(x_k))f(x_k)]. \quad (23)$$

Using (22), we can express h as

$$h = \frac{mf(x)[f'(x)]^2}{[[f'(x)]^3 - k([f'(x)]^2 - f(x)f''(x))f(x)]}. \quad (24)$$

From which we derive an iteration function

$$\phi(x) = x - \frac{mf(x)[f'(x)]^2}{[[f'(x)]^3 - k[[f'(x)]^2 - f(x)f''(x)]f(x)]}. \quad (25)$$

If $k \approx \frac{m-1}{m+1}f'$ in (25), then we get the method below.

$$x_{k+1} = \phi(x_k) = x_k - \frac{mf(x_k)f'(x_k)}{[[f'(x_k)]^2 - \frac{m-1}{m}[[f'(x_k)]^2 - f(x_k)f''(x_k)]f(x_k)]}. \quad (26)$$

If $g' \approx \frac{kf'}{f^2}g$ in (13) or in (13d) with c = 1, then putting g ≈ f we get

$$\phi(x) = x - \frac{m[f(x)]^2}{[f(x)f'(x) - kf'(x)]}, \text{ take k=1/m} \quad (27)$$

If $g' \approx \frac{kf}{f'}g$ in (13) then, we obtain

$$\phi(x) = x - \frac{mf(x)f'(x)}{([f'(x)]^2 - k[f(x)]^2)}, \text{ take k =1/m} \quad (28)$$

Also from (12) or (13) assume

$$f'' \approx m\frac{f'}{h} - m\frac{f}{h^2} + mh^m g'/h + h^m g'', \qquad (29)$$

Let $g' = kfg, g'' = (kf' + k^2 f^2)g, h = x - r$, put g = f/h^m in (29) to obtain

$$h = \frac{-mf}{hf'' - mf' - mkf^2 - h(kf' + k^2 f^2)f}, \qquad (30)$$

(Use $h = -f/f'$ in the right part). This gives,

$$\phi(x_k) = x_k - m\frac{f(x_k)f'(x_k)}{[f''(x_k)f(x_k) - k[f'(x_k) + kf(x_k)^2]][f(x_k)]^2 + m[f'(x_k)]^2 + mk[f(x_k)]^2 f'(x_k)}. \qquad (31)$$

If $g'' = 0$ in (29) or in (30) and $h = \dfrac{mf}{f'}$ in the right part of (29) or (30), then with some (sign) changes, we get (32).

$$\phi(x_k) = x_k - \frac{f(x_k)f'(x_k)}{[[f'(x_k)]^2 + k[f(x_k)]^2 f'(x_k) - f(x_k)f''(x_k)}. \qquad (32)$$

For a proper value of k, (32) holds both for single and multiple roots independent of multiplicity m.

If k = 0, then (32) is (9) and for $k \approx \dfrac{1}{4f}$ we obtain a double purpose algorithm (32b) below.

One can also take $k \approx \dfrac{1}{2mf}$ in (32).

$$\phi(x_k) = x_k - \frac{f(x_k)f'(x_k)}{[[f'(x_k)]^2 + 1/4[f(x_k)f'(x_k) - f(x_k)f''(x_k)}. \qquad (32b)$$

Note also that the general class of multiplicity independent methods can be obtained from (7b) and is given in the form

$$\phi(x) = x - \frac{fg(x)}{fg'(x) + f'g(x)},$$

but it is not easily concluded for which g it holds.

4. Construction of Some Third Order Methods for Multiple Roots

We now develop third or higher order methods based on variational technique. The variational technique used by (Muhammad et al., 2011) is that if $\varphi(x)$ is an iterative method of order p >=1 with multiplicity m > 1, then novel multiple root finding methods of order at least p+1 are given by

$$\psi(x) = \varphi(x) - \frac{m\varphi'(x)f(x)g(x)}{p[f'(x)g(x) + f(x)g'(x)]} \qquad (33)$$

for some auxiliary function g. The proof by (Muhammad et al., 2011) was finding the constant λ using the optimal condition of the iteration function $F(x) = \varphi(x) + \lambda[f(x)g(x)]^{p/m}$ to solve f(x) = 0.

Modified Newton's method (1) was used in (33) to generate higher order methods.

Here, similar proof is used but now taking another $F(x) = \varphi(x) + \lambda[f(x)/g(x)]^{p/m}$ to obtain

$$\psi(x) = \varphi(x) - \frac{m\varphi'(x)f(x)g(x)}{p[f'(x)g(x) - f(x)g'(x)]} \qquad (33b)$$

In this paper, if $\phi(x)$ and $\varphi(x)$ are two second order methods for multiple roots, then with little change in (33b) several methods $\psi(x)$ of order p >= 3 can be obtained by

$$\psi(x) = \phi(x) - \frac{m\varphi'(x)f(x)g(x)}{2[f'(x)g(x) - f(x)g'(x)]}. \qquad (33c)$$

Some of the many methods obtained from (33c) are shown below. If we take $\phi(x)$ the modified Newton's

method and
from eqn.(11c), we use

$$\varphi(x) = x - \frac{2m(m+1)f(x)}{(2m+1)f'(x)},$$

and taking $g' = kg$ in (33c),then we get

$$\psi(x) = x - m\frac{f(x)}{f'(x)} - m/2[1 - \frac{2m(m+1)}{2m+1}(1 - f(x)f''(x)/[f'(x)]^2)]\frac{f(x)}{f'(x) - kf(x)}.$$

$$= x - m\frac{f(x)}{f'(x)} - 0.5m\frac{[1 - \frac{2m(m+1)}{2m+1}(1 - f(x)f''(x)/[f'(x)]^2)]f(x)}{f'(x) - kf(x)} \text{...........}(34)$$

And when

$\phi(x) = \varphi(x) = x - \dfrac{2m(m+1)f(x)}{(2m+1)f'(x)}$, with $g' = kg$, we have

$$\psi(x) = x - \frac{2m(m+1)f(x)}{(2m+1)f'(x)} - m/2[1 - \frac{2m(m+1)}{2m+1}(1 - f(x)f''(x)/[f'(x)]^2)]\frac{f(x)}{f'(x) - kf(x)}.$$

$$= x - \frac{2m(m+1)f(x)}{(2m+1)f'(x)} - 0.5m\frac{[1 - \frac{2m(m+1)}{2m+1}(1 - f(x)f''(x)/[f'(x)]^2)]f(x)}{f'(x) - kf(x)} (34b)$$

If $\phi(x) = x - \dfrac{2m(m+1)f(x)}{(2m+1)f'(x)}, \varphi(x) = x - \dfrac{mf(x)}{f'(x)}$,
then with g'= kg we obtain

$$\psi(x) = x - \frac{2m(m+1)f(x)}{(2m+1)f'(x)} - 0.5m\frac{[1 - m(1 - f(x)f''(x)/[f'(x)]^2)]f(x)}{f'(x) - kf(x)} \quad (35)$$

If $\phi(x_n) = x_{n+1} = x_n - 0.5[m\dfrac{f(x_n)}{f'(x_n)} - (1-m)\dfrac{f'(x_n)}{f''(x_n)}], \varphi(x) = x - \dfrac{2m(m+1)f(x)}{(2m+1)f'(x)}$, with $g' = kg$,

one has

$$\psi(x_n) = x_{n+1} = x_n - 0.5[m\frac{f(x_n)}{f'(x_n)} - (1-m)\frac{f'(x_n)}{f''(x_n)}] - m/2[1 - \frac{2m(m+1)}{2m+1}(1 - f(x)f''(x)/[f'(x)]^2)]\frac{f(x)}{f'(x) - kf(x)}.$$

(36)

Note that one can choose k = -1 or k = 1 in (34), (34b) or (35) and (36) but there may be many cases.
It is also possible to use any second order for multiple roots in (33c). For further information on variational techniques for roots, refer to (Farook et al., 2016; Farook & Muhammad, 2014; Muhammad et al., 2011).

5. Convergence Analysis

Theorem 5.1 Let $\phi(x) = \varphi(x)$ be an arbitrary iteration function of order p ≥1 at a multiple root r with multiplicity m of f(x). Suppose also that $\phi(x)$ is continuously differentiable at r, then $\psi(x)$ in (33) has order at least p+1(Muhammad et al., 2011). See also proofs by (Muhammad et al., 2011).

Theorem 5.2 (Order of Convergence) Assume that $\phi(x)$ has sufficiently many derivatives at a root r of $f(x)$. The order of any one-point iteration function $\phi(x)$ is a positive integer p, more especially $\phi(x)$ has order p if and only if $\phi(r) = r$ and $\phi^{(j)}(r) = 0$ for 0 < j < p, $\phi^{(p)}(r) \neq 0$ (Anthony &Philip, 1978; Alfio et al., 2000).
All the algorithms in this paper need an appropriate choice of only one initial guess x_o in an interval Io = [a, b].

And random choices of x_o may lead us to unnecessary works. So we need not do that. From theorem 6.2 above and convergence of fixed point iteration method $x = \phi(x)$, $|\phi'(x)| < 1$ for all x in [a, b]. From which h $|\phi'(x_o)| < 1$. The case $|\phi'(x_o)| > 1$ is divergence. And $|\phi'(x_o)| = 1$ needs especial treatment (reformulations or need for alternative methods; Germund & Ake, 2008).

1) Proof of order of convergence p of (9) and (32b)

One has two cases since both algorithms are used for solving simple and multiple roots.

If m = 1(r is simple root of $f(x)$), then (9) can be written as

$\phi(x) = x - (x - \theta(x))/T(x)$, where $T(x) = 1 - f(x)f''(x)/[f'(x)]^2$

and $\theta(x)$ is Newton's iteration function which has order p = 2.

We see that $\phi(x) = x, \theta'(r) = 0, \theta''(r) \neq 0.$

Applying theorem 6.2, differentiating (9) or $\phi(x)$, we have $\phi'(r) = 0$ and $\phi''(r) \neq 0.$

On the other hand, if p = 2, then we easily show that $\phi'(r) = 0$ but $\phi''(r) \neq 0.$

So p = 2 for (9) when m = 1.

In case of m >1 (r is multiple root), (9) can be written as

$\phi(x) = x - 1/m(x - \theta(x))/T(x)$, where $\theta(x)$ is modified Newton's iteration function of order p = 2 as proved by (Anthony&Philip,1978). Applying the same proof of $\theta(x)$ here for $\phi(x)$ and adding concepts in theorem 5.2, one obtains p = 2.

2) If m = 1, then (32b) can be written as $\phi(x) = x - (x - \theta(x))/B(x)$, with

$B(x) = 1 + 1/4\dfrac{f(x)}{f'(x)} - f(x)f''(x)/[f'(x)]^2$ and $\theta(x)$ is Newton's iteration function.

Applying theorem 6.2 as in **1)** above, we get p = 2 for m = 1. And if m > 1, then (32b) can be expressed as $\phi(x) = x - 1/m(x - \theta(x))/B(x)$, with $\theta(x)$ is modified Newton's method. Applying similar proofs as in the second case of **1),** again we get p = 2.

3) Proof of order of convergence p of (17).

(17) can be expressed as $\phi(x) = x - (x - \theta(x))/H(x)$, where

$H(x) = 1 - k[f(x)]^{2.}/f'(x)$, and $\theta(x)$ is the modified Newton's iteration function of p = 2. Applying the proof for order of $\theta(x)$ now for $\phi(x)$ here, or by theorem 6.2 one obtains p = 2. Hence (17) and (18) are second order convergent. Similar proofs can be used to show that (11c), (23), (26), (28) and (32b) are all of order p = 2. Note also that (11c) holds very well for simple roots. From (11c)

$$\varphi(x) = x - \frac{2m(m+1)f(x)}{(2m+1)f'(x)}.$$

And for simple roots m = 1. So this gives again a quadratic convergent method

$\varphi(x) = x - \dfrac{4f(x)}{3f'(x)}$. Note that both (11c) and (1) could be obtained from (4).

4) By theorem 5.1, p = 2 and then (34b) become of order p+1 = 2+1 = 3. Similarly (34), (35) and (36) are cubic convergent.

Note that theorem 5.1 holds also for (33b) and (33c). The proofs for most of the first order methods in section-2 can be from the iteration function in (7b) which is linear. One can also use other way of proofs in the literatures.

6. Test Equations and Numerical Results

The following equations were selected for test of convergence.

$f_1(x) = x^3 - 3x^2 + 3x - 1 = 0$,

with x_o = 0, 2, 3, and r =1 in [1, 2), m = 3.

$f_2(x) = [3x - \cos x - 1]^2 = 0$

with $x_o = 0, 1, 2$ and $r \approx 0.607102$ in (0,1), m = 2.

$f_3(x) = x^2 - 2xe^{-x} + e^{-2x} = 0$,

with $x_o = 0, 1, 2$ and $r \approx 0.567143$ in (0,1), m = 2.

$f_4(x) = 1 - 2\sqrt{x} + x$, $x_o = 0.5, 1.5, 2$

and root =1.000000 in [1, 2), m = 2.

$f_5(x) = 1 + 4x + 6x^2 + 4x^3 + x^4$,

$x_o = -1, 0, 1$, and root =-1.000000 in [-1, 0), m = 4.

$f_6(x) = [e^{x-1} - 1/x]^2 = 0, x_o = 0.5, \; 1.5, \; 2,3,$

and r = 1.000000 in [0, 1], m = 2.

$f_7(x) = [\log(x+9) - 1]^2 = 0, x_o = 0.5, 1.5, 2$

and r = 1.000000 in [0, 1], m = 2.

Comparisons were made relative to modified Newton method (MN), Halley's method (HM), the algorithms in equations (9), (18), (23), (26), (27), (28), (32b)(34), (34b). C++ implementation was done for each algorithms and the number of iterations taken to converge to a root r was recorded and written in the body of the next **table-2** under each method. The stopping criteria were using the error $E_i = |x_{i+1} - x_i|$ such that $E_i \le \varepsilon$, for chosen $\varepsilon = 10^{-33}$.

Note: The numbers in each cells of table-2 on the next page correspond to the number of iterations needed for convergence at each of the three initial guesses of a root r of $f_i(x) = 0, i = 1,2,...,7$.

From the table Halley's method (HM) for solving $f_3(x) = x^2 - 2xe^{-x} + e^{-2x} = 0$ converges at steps 2, 2, 3 for the initial guesses taken at $x_o = 0.5,1,2$ respectively and being ε given. In the first column, ´´**Functions** (f) ´´ refers to the number of function evaluations up to derivatives and ´´**Efficiency** (e) ´´ represents the computational efficiency index calculated by $p^{1/f}$. An algorithm with the least average number of iterations (**Nar**) to converge to a root r would be ranked the fastest convergent. Taking more initial guesses or more examples gives good average ranking measure. Note that average number of iteration was also used by Changbum Chun & Beny Neta (2009).

In the table, the highest value of efficiency index is 1.442 (for third order methods (34), (34b) and (HM)) and the lowest value is 1.260 (for second order). Any fourth order with four function evaluations [p = 4, f = 4] and any second order with [p = f = 2] have equal computational efficiency indices e = 1.4142. It may not be true that the higher the order is the better the efficiency or the method in general. We can observe that all methods presented in the table are better competent with 2 to 3 average number of iterations to converge from both directions at x_o when an appropriate initial guess x_o is taken, especially near a unit length interval containing the root r.If x_o is not suitably chosen, then one can expect slow convergence and even divergence cases. As an example, we have checked that algorithm (9) and (32b) fail to converge to the desired root when x_o is not properly chosen for solving simple and multiple roots of some equations, but are fast convergent. See the table of results below.

Note: in the table below use the following letters as follows.

Nar = average number of iterations
f = number of function evaluations
p = order of convergence
e = computational efficiency indices

Table-2 Summary of numerical results

The number of iterations needed to converge to a root r for every triplets of initial guesses									
$f(x)$	MN	(9)	(18)	HM	(28)	34	34b	(27)	32b
$f_1(x)$	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1
$f_2(x)$:	3, 3, 3	3, 3, 3	3, 3, 3	2, 2, 3	3, 3, 3	2, 2, 2	2, 2, 2	3, 3, 3	3, 3, 3
$f_3(x)$:	3, 3, 4	3, 3, 4	3, 3, 4	2, 2, 3	3, 3, 4	2, 2, 4	2, 2, 4	3, 3, 4	3, 3, 3
$f_4(x)$	4, 3, 4	4, 3, 4	4, 3, 4	3, 2, 3	4, 3, 4	2, 3, 3	2, 3, 3	4, 3, 4	4, 3, 4
$f_5(x)$:	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1	1, 1, 1
$f_6(x)$	4, 3, 4	4, 3, 4	4, 3, 4	3, 3, 3	4, 3, 4	3, 3, 3	3, 3, 3	4, 3, 4	4, 3, 4
$f_7(x)$:	3,3, 4	3, 3, 3	3, 3, 4	3, 3, 4	3, 3, 4	3, 3, 4	3, 3, 3	3, 3, 4	3, 3, 3
Nar	3	3	3	2	3	2	2	3	3
Fun (f)	2	3	2	3	2	3	3	2	3
Ord(p)	2	2	2	3	2	3	3	2	2
Effi(e)	1.414	1.2560	1.4142	1.442	1.414	1.442	1.442	1.414	1.256

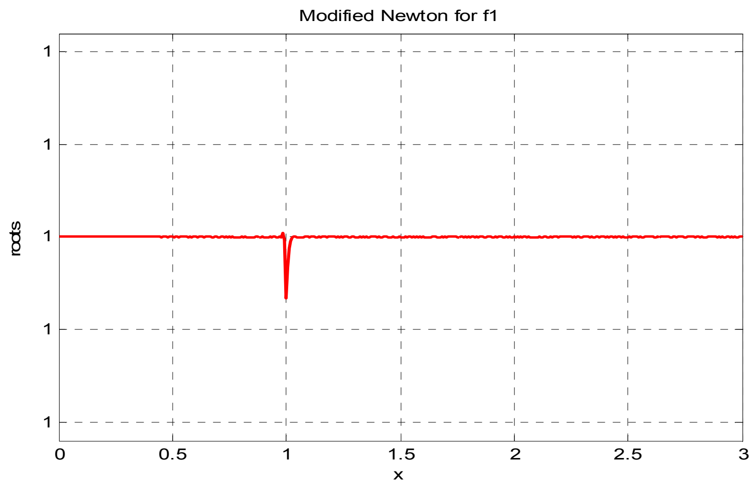

Fig 1: convergence of modified Newton's method for solving $f_1[r = 1]$

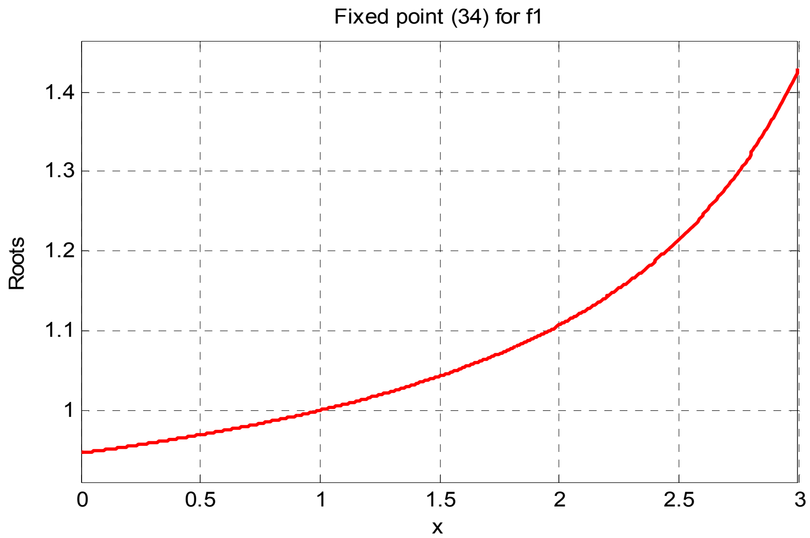

Fig 2: Convergence of method in (34) for solving $f_1[r = 1]$

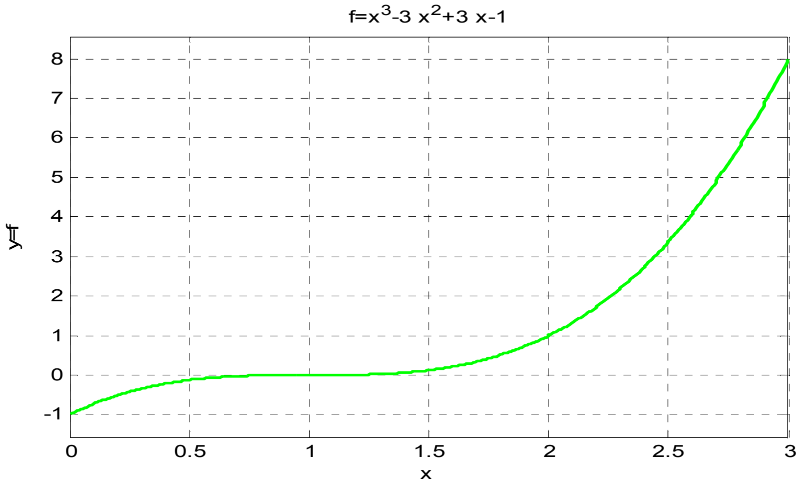

Fig 3 : Graph of f_1 for root location [r = 1]

Note that f1= x^3-3*x^2+3*x-1 is the polynomial of an identity matrix of order 3, with all Eigen values r = 1.

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

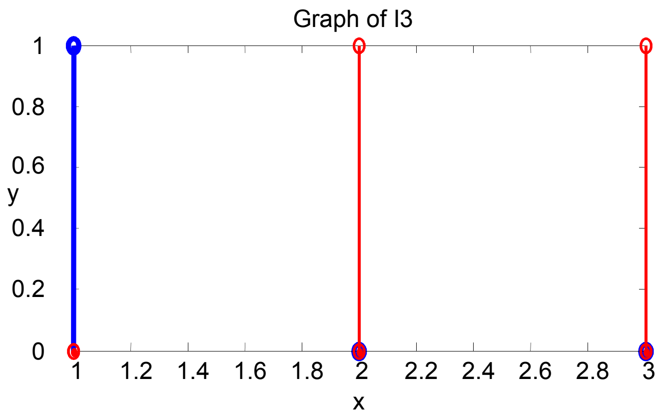

Fig 4: Tops (from above) of the graph show 3 identical Eigen values

7. Conclusions

- In this article, we have presented iterative methods of order 1, 2, 3, 4 both for multiple and simple roots. By deriving for the error-correction h from the derivative (s) of an expression of the equation to be solved, we obtained iterative methods of order two for estimating multiple roots of scalar nonlinear equations. We have developed third order methods based on variational techniques with additional concepts. We have also discussed some basic root finding methods such as the Newton method, Chun-Neta method, Halley's method, Osada's method, Chebyshev's method and (6) for multiple roots. We have seen that there are modified methods in this paper that are second order (see (11c), (18), (23), (26), (27), (28), (32b)) and third order convergent (see (34), (34b), (35), (36)). They are better competent with some existing methods. We showed that there are modified methods of order two with only two function evaluations up to derivatives (see (18), (27), (28)). We presented two second order convergent algorithms which are independent of multiplicity m (see eqn.(9) an existing method and a new one eqn. (32b)).They are fast convergent both for simple and multiple roots. We have also done convergence analysis with proofs and graphs. In the future we will present further analyses of the topic and other higher order iterative algorithms. We hope that this result will be more commendable and commence one to perform further research in the area.

Acknowledgements
I would like to thank the staff of mathematics in ASTU.

REFERENCES

Alfio Quarteroni, Riccardo Sacco, Fausto Saleri (2000), Numerical mathematics (Texts in applied mathematics; 37), Springer-Verlag New York, Inc., USA.

Anthony Ralston, Philip Rabinowitz (1978), A first course in numerical analysis (2nd ed.), McGraw-Hill Book Company, New York.

Changbum Chun and Beny Neta.(2009), A third-order modification of Newton's method for multiple roots, Applied Mathematics and Computation 211, 474–479

Farooq Ahmed Shah, Muhammad Aslam Noor and Muhammad Waseem(2016), Some second-derivative-free sixth-order convergent iterative m ethods for non-linear equations, *Maejo Int. J. Sci. Technol.* , *10*(01), 79-87;

Farooq Ahmed Shah and Muhammad Aslam Noor.(2014), Variational Iteration Technique and Some Methods for the Approximate Solution of Nonlinear Equations, Appl. Math. Inf. Sci. Lett. 2, No. 3, 85-93

Germund Dahlquist, Ake Bjorck(2008), Numerical methods in scientific computing

Volume I, Siam - Society for industrial and applied mathematics. Philadelphia, USA.

Janak Raj Sharma and Rajni Sharma (2012), Modified Chebyshev-Halley type method and its variants for computing multiple roots, Springer science + Business Media, LLC.

Jurgen Gerlach (1994), Accelerated Convergence in Newton's Method, Society for industrial and applied mathematics,Siam Review 36, 272-276.

J.Stoer, R.Bulirsch (1993), Texts in Applied mathematics 12, Introduction to Numerical Analysis (2nd ed.), springer-Verlag New York, Inc., USA

Malvin R.Spencer (1994), PhD Dissertation: Polynomial Real Root Finding in Bernstein Form, Department of engineering Brigham Young University

M.M. Hussein (2009), A note on one-step iteration methods for solving nonlinear equations, World Applied Sciences Journal 7 (special issue for applied math), 90-95, IDOSI Publications.

Muhammad Aslam Noor, Farooq Ahmed Shah, Khalida Inayat Noor and Eisa Al-Said (2011), variation iteration technique for finding multiple roots of nonlinear equations, Scientific Research and Essays Vol.6(6),pp.1344-1350

Munish Kansal, V. Kanwar_, and Saurabh Bhatia (2015), On Some Optimal Multiple Root-Finding Methods and their Dynamics, An International Journal(AAM) : ISSN: 1932-9466, Vol. 10, Issue 1, pp. 349 – 367

Pakdemirli, H. Boyacl and H. A. Yurtsever 92008), A root finding algorithm with fifth order derivatives, Mathematical and Computational Applications,Vol.13, 123-128

Rajinder Thukral (2013), Introduction to Higher-Order Iterative Methods for Finding Multiple Roots of Nonlinear Equations, Hindawi Journal of mathematics, Article ID 404635: http://dx.doi.org/10.1155/2013/404635

Richard F. King (1977), A Secant Method for Multiple Roots, BIT 17, 321-328

Tekle Gemechu (2017), Root Finding With Extensions to Higher Dimension, Mathematical Theory and Modeling (IISTE, U.S.A),Vol. 7 No. 4, ISSN 2225-0522

Tekle Gemechu (Dinka)-(Dissertation 2014), Perturbation effect of scalar polynomial coefficients and real root finding algorithms with applications, CCNU.

YI JIN AND BAHMAN KALANTARI (2010) (communicated by Peter A. Clarkson), A combinatorial construction of high order algorithms for finding polynomial roots of known multiplicity, American Mathematical Society 138, 1897-1906.