Acta Universitatis Apulensis No. 45/2016
ISSN: 1582-5329 pp. 73-86
http://www.uab.ro/auajournal/ doi: 10.17114/j.aua.2016.45.06

NINTH ORDER METHOD FOR NONLINEAR EQUATIONS AND ITS DYNAMIC BEHAVIOUR

M. Rafiullah, D. K. R Babajee and Dur-e-Jabeen

Abstract. The aim of this paper is to construct a new efficient iterative method to solve nonlinear equations and discuss the dynamic behaviour of it. This method is based on "A fifth-order iterative method for solving nonlinear equations, *Numerical Analysis and Applications*, **4** (3) (2011), pp. 239–243". The finite difference and Hermite interpolation are used to improve the convergence order and efficiency index of this method. The new method is of the ninth order of convergence and it is compared with other ninth order methods. Some numerical test problems are given to show the accuracy and fast convergence of the method proposed. The dynamic behaviour of the methods for finding the roots of unity are also studied.

2010 *Mathematics Subject Classification*: 65H05, 65D05, 41A25.

Keywords: Nonlinear equations, Iterative methods, Convergence analysis, Hermite interpolation, Dynamic behaviour, Polynomiograph

1. Introduction

Solving nonlinear equations $f(x) = 0$ is one of the most important and challenging problem in scientific and engineering applications. These equations largely occur in daily life with useful applications such as, to measure the speed of rocket, to find the eigen values of a system, to measure compressibility of gasses, to discuss aging model of a cell's energy producing organelle (mitochondria), to calculate the simple harmonic oscillation, to measure the variation of the local heat transfer, to measure the interior temperature of a material, to the assessment of drug concentration in plasma, to produce the methanol from CO and H_2 by using the equilibrium equation, to measure the velocity of a falling parachutist, etc. Finding the solution of nonlinear equations is not an easy task. Mostly, the analytical methods fail to find their solutions. Ultimately, for this purpose we move towards the numerical methods. There are numerous and well known methods which help us to deal with nonlinear equations such as Bisection, Regula False, Newton-Raphson, Secant, Steffensen, Halley, Jarratt, Ostrowski, King's methods, etc.

Now, consider a nonlinear equation

$$f(x) = 0. \tag{1.1}$$

The Newton-Raphson method is largely used to solve such nonlinear equations (1.1) and it is written as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0. \tag{1.2}$$

This is an important and basic method [8], which converges quadratically.

Recently, a large variety of methods which based on the Newton's method are proposed to solve nonlinear equations. All these modified methods are in the direction of improving the efficiency index and order of convergence by using lower-order derivatives as possible.

2. Preliminaries

Definition 1. *Let $x_0, x_1, x_2, \ldots$ be a sequence which converges to α. Let $e_n = x_n - \alpha$. If there exist a real number p and a positive constant C such that $\frac{|e_{n+1}|}{|e_n|^p} \to C$, as n becomes large, i.e.*

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^p} = C$$

then p is called the order of convergence and C is called asymptotic constant [6].

Definition 2. *According to Ostrowski [6, 8] the efficiency index $\overset{*}{eff}$ is defined as*

$$\overset{*}{eff} = p^{\frac{1}{d}},$$

where d is called the number of function evaluations used in method.

Definition 3. *The computational order of convergence (COC) $\bar{p}$ of a sequence $\{x_n\}_{n\geq 0}$ is defined by*

$$\bar{p}_n = \frac{\ln\left|\frac{e_{n+1}}{e_n}\right|}{\ln\left|\frac{e_n}{e_{n-1}}\right|},$$

where x_{n-1}, x_n and x_{n+1} are three consecutive iterations near the root α and $e_n = x_n - \alpha$ [10].

Definition 4. *Suppose that a set of points $x_0, x_1, x_2, ...x_n$ and respective function values on these points are $y_0, y_1, y_2, ...y_n$ then the first order divided difference of first two points can be define as $f[x_0, x_1] = \frac{y_1-y_0}{x_1-x_0}$ and second order divided difference can be defined as define as $f[x_0, x_1, x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$. the general formula is*

$$f[x_0, x_1, \cdots, x_{k-1}, x_k] = \frac{f[x_1, x_2, \cdots, x_k] - f[x_0, x_1, \cdots, x_{k-1}]}{x_k - x_0}$$

where $k = 2, 3, \cdots, n$ [5].

Theorem 1. *The order of points does not matter such as $f[x_0, x_1] = f[x_1, x_0]$ or $f[x_0, x_1, x_2] = f[x_2, x_1, x_0]$.*

Theorem 2. *For any two points of $x_0, x_1, x_2, \cdots, x_n$ happen to be equal, then the divided difference can be define as follows [10]:*

$$f[x_0, x_0] = \lim_{x_1 \to x_0} \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f'(x_0).$$

Definition 5. *The Newton's polynomial for the data points (x_0, y_0), (x_1, y_1),$\cdots$, (x_n, y_n) using the divided differences definition can be written as*

$$\begin{aligned} P_n(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \\ &\quad \cdots + f[x_0, x_1, \cdots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned}$$

$$P_n(x) = f[x_0] + \sum_{i=1}^{n} f[x_0, \cdots, x_i] \prod_{j=0}^{i-1} (x - x_j).$$

Definition 6. *Hermite interpolation is a method to find out a polynomial of degree $2n + 1$ which satisfy the function values at n known data points and first m derivatives. Hermite Interpolation is likely Newton's divided difference formula but these both differs by the polynomial degree. Lets consider n data points with m derivatives*

$$\begin{array}{cccc} (x_0, f(x_0)) & (x_1, f(x_1)) & \cdots & (x_{n-1}, f(x_{n-1})) \\ (x_0, f'(x_0)) & (x_1, f'(x_1)) & \cdots & (x_{n-1}, f'(x_{n-1})) \\ \vdots & \vdots & \ddots & \vdots \\ (x_0, f^{(m)}(x_0)) & (x_1, f^{(m)}(x_1)) & \cdots & (x_{n-1}, f^{(m)}(x_{n-1})) \end{array}$$

The Hermite interpolation polynomial can be determine by using a modification of Newton's divided difference method.

Modification of Newton's divided differences for Hermite Interpolation

Suppose that the distinct numbers $x_0, \cdots, x_n$ are given together with the values of f and f' at these numbers. Define a new sequence $z_0, z_1, \cdots, z_{2n+1}$ by $z_{2i} = z_{2i+1} = x_i$ for each $i = 0, 1, \cdots, n$, and contract divided difference table. We can not define $f[z_{2i}, z_{2i+1}]$ by the divided difference formula, so the reasonable substitution in this situation is $f[z_{2i}, z_{2i+1}] = f'(x_{2i}) = f'(x_i)$, we can use the entries $f'(x_0),\ f'(x_1), \cdots,\ f'(x_n)$ in the place of the undefined first divided differences $f[z_0, z_1],\ f[z_2, z_3],\ \cdots\ ,\ f[z_{2n}, z_{2n+1}]$. The remaining divided differences are produced as usual, and the appropriate divided differences are employed in Newton's interpolatory divided-difference formula [2, p. 137]. The Hermite polynomial is given by

$$H_{2n+1}(x) = f[z_0] + \sum_{k=0}^{2n+1} f[z_0, z_1, \cdots, z_k](x - z_0)(x - z_1)\cdots(x - z_{k-1}).$$

The divided difference table constructed as for Hermite polynomial

z	$f(z)$	1st DD	2nd DD
$z_0 = x_0$	$f[z_0] = f(x_0)$		
		$f[z_0, z_1] = f'(x_0)$	
$z_1 = x_0$	$f[z_1] = f(x_0)$		$f[z_0, z_1, z_2] = \frac{f[z_1,z_2]-f[z_0,z_1]}{z_2-z_0}$
		$f[z_1, z_2] = \frac{f[z_2]-f[z_1]}{z_2-z_1}$	
$z_2 = x_1$	$f[z_2] = f(x_1)$		$f[z_1, z_2, z_3] = \frac{f[z_2,z_3]-f[z_1,z_2]}{z_3-z_1}$
		$f[z_2, z_3] = f'(x_1)$	
$z_3 = x_1$	$f[z_3] = f(x_1)$		$[z_2, z_3, z_4] = \frac{f[z_3,z_4]-f[z_2,z_3]}{z_4-z_2}$
		$f[z_3, z_4] = \frac{f[z_4]-f[z_3]}{z_4-z_3}$	
$z_4 = x_2$	$f[z_4] = f(x_2)$		$[z_3, z_4, z_5] = \frac{f[z_4,z_5]-f[z_3,z_4]}{z_5-z_3}$
		$f[z_4, z_5] = f'(x_2)$	
$z_5 = x_2$	$f[z_5] = f(x_2)$		

This Table shows the entries that are used for the first three divided-difference columns when determining the Hermite polynomial $H_5(x)$ for x_0, x_1 and x_2". The Hermite polynomial $H_{2n+1}(x)$ can also be written as

$$\begin{aligned} H_{2n+1}(x) \quad &= \quad f[z_0] + f[z_0, z_1](x - x_0) + f[z_0, z_1, z_2](x - x_0)^2 \\ &\quad + f[z_0, z_1, z_2, z_3](x - x_0)^2(x - x_1) + \\ &\quad ... + f[z_0, z_1, ..., z_{2n+1}](x - x_0)^2(x - x_1)^2...(x - x_{n-1}). \end{aligned}$$

3. Derivation of new methods

Consider the following iterative method [7]

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$
$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{1}{2}\left[\frac{f(y_n)}{f'(x_n)}\right]^2 \frac{f''(y_n)}{f'(y_n)}. \tag{3.1}$$

This method is fifth-order and it involves five functions $f(x_n)$, $f'(x_n)$, $f(y_n)$, $f'(y_n)$, $f''(y_n)$ at each step, so the efficiency index of this method is $p^{\frac{1}{d}} = 5^{\frac{1}{5}} = 1.3797$. First of all we eliminate the term $\frac{f''(y_n)}{f'(y_n)}$ from (3.1) to improve the efficiency index of the method (3.1) by approximating $f''(y_n)$ and $f'(y_n)$ using divided differences with known functions and their derivatives as $\frac{f'(y_n)-f'(x_n)}{y_n-x_n}$ and $\frac{f(y_n)-f(x_n)}{y_n-x_n}$ respectively, so we have

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{f(y_n)^2\,(f'(x_n) - f'(y_n))}{2(f(x_n) - f(y_n))f'(x_n)^2}. \tag{3.2}$$

This method involves only four functions evaluation such as $f(x_n)$, $f'(x_n)$, $f(y_n)$, $f'(y_n)$. The order of this method remains same as fifth with error term $\frac{1}{2}(10c_2^4 - 3c_2^2c_3)e_n^5$. But efficiency index of this method is improved as $p^{\frac{1}{d}} = 5^{\frac{1}{4}} = 1.4953$.

Now at the third step, introducing Newton-Raphson method gives

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$
$$z_n = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{f(y_n)^2\,(f'(x_n) - f'(y_n))}{2(f(x_n) - f(y_n))f'(x_n)^2}$$
$$x_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}. \tag{3.3}$$

By suggesting this new step the function evaluations also increases as six. We reduce the number of functions by approximating the $f'(z_n)$ using the Hermite interpolation polynomial with known data points $(x_n, f(x_n))$, $(y_n, f(y_n))$, $(z_n, f(z_n))$, $(x_n, f'(x_n))$,

$(y_n, f'(y_n))$. So we have,

$$\begin{aligned}
H_4(x) &= f[z_0] + f[z_0,z_1](x-x_n) + f[z_0,z_1,z_2](x-x_n)^2 \\
&\quad + f[z_0,z_1,z_2,z_3](x-x_n)^2(x-y_n) + f[z_0,z_1,z_2,z_3,z_4](x-x_n)^2(x-y_n)^2 \\
H_4'(x) &= f[z_0,z_1] + 2f[z_0,z_1,z_2](x-x_n) \\
&\quad + f[z_0,z_1,z_2,z_3]\left((x-x_n)^2 + (x-x_n)(x-y_n)\right) \\
&\quad + f[z_0,z_1,z_2,z_3,z_4]\left((x-x_n)^2(x-y_n) + (x-x_n)(x-y_n)^2\right).
\end{aligned}$$

we approximate $dfz = f'(z_n) = H_4'(z_n)$ as

$$\frac{1}{(x_n-y_n)^3}\left(\begin{array}{c}
\left(\begin{array}{c} f(z_n)(x_n-y_n)^3 + f'(y_n)(x_n-y_n)(x_n-z_n) \\ \left(x_n+2y_n+x_ny_n-(3+x_n+y_n)z_n+z_n^2\right) \\ -f(y_n)(x_n-z_n) \\ (-6y_n+6z_n+(x_n-z_n)(x_n-3y_n+2z_n)) \end{array}\right) + (y_n-z_n) \\
\left(\begin{array}{c} f(x_n)\left(\begin{array}{c} y_n^2 - 3x_n(2+y_n-z_n) + y_nz_n \\ -2(-3+z_n)z_n \end{array}\right) \\ +f'(x_n)(x_n-y_n)\left(\begin{array}{c} 2x_n+y_n+x_ny_n \\ -(3+x_n+y_n)z_n+z_n^2 \end{array}\right) \end{array}\right)
\end{array}\right).$$

So the new method is

$$\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
z_n &= y_n - \frac{f(y_n)}{f'(y_n)} - \frac{f(y_n)^2\left(f'(x_n)-f'(y_n)\right)}{2(f(x_n)-f(y_n))f'(x_n)^2} \\
x_{n+1} &= z_n - \frac{f(z_n)}{dfz}.
\end{aligned} \tag{3.4}$$

This method involves five function evaluations and the order of convergence is improved up to the ninth. The error term is $(10c_2^5c_4 - 3c_2^3c_3c_4)e_n^9$. The efficiency index is $p^{\frac{1}{d}} = 9^{\frac{1}{5}} = 1.5518$ which is better than (3.1) and (3.2).

4. Convergence Analysis

Theorem 3. *Let α be a simple zero of sufficiently differentiable function f: $I \subseteq R \to R$ for an open interval I. If x_0 closed to α, then the method defined by (3.2) is of fifth order and moreover satisfies the following error equation*

$$e_{n+1} = \frac{1}{2}(10c_2^4 - 3c_2^2c_3)e_n^5.$$

where $e_n = x_n - \alpha$, $e_{n+1} = x_{n+1} - \alpha$ and $c_k = \frac{f^{(k)}(\alpha)}{k!f'(\alpha)}$.

Proof. Using Taylor series we have

$$f(x_n) = f'(\alpha) \left[\begin{array}{c} e_n + c_2 {e_n}^2 + c_3 {e_n}^3 + c_4 {e_n}^4 \\ + c_5 {e_n}^5 + c_6 {e_n}^6 + c_7 {e_n}^7 + c_5 {e_n}^8 + \cdots \end{array} \right] \tag{4.1}$$

$$f'(x_n) = f'(\alpha) \left[\begin{array}{c} 1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 \\ +5c_5 e_n^4 + 6c_6 {e_n}^5 + 7c_7 {e_n}^6 + 8c_8 {e_n}^7 + \cdots \end{array} \right] \tag{4.2}$$

$$\begin{aligned} y_n = x_n - \frac{f(x_n)}{f'(x_n)} &= \alpha + c_2 e_n^2 + \left(-2c_2^2 + 2c_3\right) e_n^3 + \left(4c_2^3 - 7c_2 c_3 + 3c_4\right) e_n^4 \\ &- 2\left(4c_2^4 - 10c_2^2 c_3 + 3c_3^2 + 5c_2 c_4 - 2c_5\right) e_n^5 + \cdots \end{aligned} \tag{4.3}$$

Expanding $f(y_n)$ around root and using (4.3), we have

$$\begin{aligned} f(y_n) &= f'(\alpha)[c_2 e_n^2 + \left(-2c_2^2 + 2c_3\right) e_n^3 + \left(5c_2^3 - 7c_2 c_3 + 3c_4\right) e_n^4 \\ &- 2\left(6c_2^4 - 12c_2^2 c_3 + 3c_3^2 + 5c_2 c_4 - 2c_5\right) e_n^5 + ...] \end{aligned} \tag{4.4}$$

$$\begin{aligned} f'(y_n) &= f'(\alpha)[1 + 2c_2^2 e_n^2 + \left(-4c_2^3 + 4c_2 c_3\right) e_n^3 + c_2\left(8c_2^3 - 11c_2 c_3 + 6c_4\right) e_n^4 \\ &- 4\left(c_2\left(4c_2^4 - 7c_2^2 c_3 + 5c_2 c_4 - 2c_5\right)\right) e_n^5 + \cdots] \end{aligned} \tag{4.5}$$

Using (4.1) to (4.5) in (3.2), we have

$$x_{n+1} = \alpha + \frac{1}{2}\left(10c_2^4 - 3c_2^2 c_3\right) e_n^5 + \left(-40c_2^5 + 41c_2^3 c_3 - 6c_2 c_3^2 - 2c_2^2 c_4\right) e_n^6 + \cdots \tag{4.6}$$

so

$$e_{n+1} = \frac{1}{2}\left(10c_2^4 - 3c_2^2 c_3\right) e_n^5 + O(e_n^6).$$

Theorem 4. *Let α be a simple zero of sufficiently differentiable function f: $I \subseteq R \to R$ for an open interval I. If x_0 closed to α, then the method defined by (3.4) is of ninth order and moreover satisfies the following error equation*

$$e_{n+1} = (10c_2^5 c_4 - 3c_2^3 c_3 c_4) e_n^9.$$

where $e_n = x_n - \alpha$, $e_{n+1} = x_{n+1} - \alpha$ and $c_k = \frac{f^{(k)}(\alpha)}{k! f'(\alpha)}$.

Proof. from (3.4) and (4.6)

$$z_n = \alpha + \frac{1}{2}\left(10c_2^4 - 3c_2^2 c_3\right) e_n^5 + \left(-40c_2^5 + 41c_2^3 c_3 - 6c_2 c_3^2 - 2c_2^2 c_4\right) e_n^6 + \cdots \tag{4.7}$$

Expanding $f(z_n)$ around root and using (4.7), we have

$$f(z_n) = f'(\alpha)\left[\begin{array}{c} \left(5c_2^4 - \frac{3}{2}c_2^2c_3\right)e_n^5 \\ -c_2\left(40c_2^4 - 41c_2^2c_3 + 6c_3^2 + 2c_2c_4\right)e_n^6 \\ +\frac{1}{2}\left(\begin{array}{c} 398c_2^6 - 683c_2^4c_3 - 12c_3^3 + 110c_2^3c_4 \\ -34c_2c_3c_4 + c_2^2\left(249c_3^2 - 5c_5\right)\end{array}\right)e_n^7 \\ +\frac{1}{2}\left(\begin{array}{c} -1584c_2^7 + 3777c_2^5c_3 - 926c_2^4c_4 - 52c_3^2c_4 \\ +c_2^3\left(-2319c_3^2 + 142c_5\right) \\ +4c_2\left(83c_3^3 - 6c_4^2 - 11c_3c_5\right) \\ +c_2^2\left(670c_3c_4 - 6c_6\right)\end{array}\right)e_n^8 \\ +\frac{1}{2}\left(\begin{array}{c} 5530c_2^8 - 16789c_2^6c_3 + 5184c_2^5c_4 \\ +c_2^4\left(14759c_3^2 - 1202c_5\right) \\ +c_3\left(164c_3^3 - 75c_4^2 - 68c_3c_5\right) \\ +c_2^3\left(-6242c_3c_4 + 174c_6\right) \\ +2c_2\left(671c_3^2c_4 - 31c_4c_5 - 27c_3c_6\right) \\ +c_2^2\left(-3919c_3^3 + 446c_4^2 + 866c_3c_5 - 7c_7\right)\end{array}\right)e_n^9 + \cdots \end{array}\right] \tag{4.8}$$

Using (4.1) to (4.5) and (4.7) to (4.8) in (3.4), we have

$$x_{n+1} = \alpha + \left(10c_2^5c_4 - 3c_2^3c_3c_4\right)e_n^9 + \left(8382c_2^9 - \frac{61125}{2}c_2^7c_3 + ...\right)e_n^{10} \tag{4.9}$$

so

$$e_{n+1} = \left(10c_2^5c_4 - 3c_2^3c_3c_4\right)e_n^9 + O(e_n^{10}).$$

5. Applications

We have compared our new method (3.4) with the following methods having order nine respectively.

Gradimir *et al.* in 2007 [12] suggested a ninth order method (**GM**) which is given below,

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

Hu *et al.* in 2011 [4] proposed a ninth-order method (**HU**)

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

In 2012, Hafiz and Salwa [11] proposed ninth order method (**HS**) using Halley iterative method and the weight combination of mid-point with Simpson's quadrature formulas and using predictor–corrector technique, as

$$
\begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
w_n &= \frac{x_n + y_n}{2} \\
z_n &= x_n - \frac{12 f(x_n)}{f'(x_n) + 10 f'(w_n) + f'(y_n)} \\
x_{n+1} &= z_n - \frac{2 f(z_n) f'(z_n)}{2\left[f'(z_n)\right]^2 - f(z_n) f''(z_n)}.
\end{aligned}
$$

Now, consider some test problems to illustrate the efficiency of the proposed method, namely R-1 (3.4) which is of ninth order. First we show first three iterations of given examples using method R-1 (3.4), accuracy $|x_3 - x_2|$ and computational order of convergence (COC) of R-1 in table-1. Secondly in table-2, we have compared the results ($|x_3 - x_2|$) of new method R-1 (3.4) with the results of existing methods namely GM, HU and HS which are also ninth order.

Examples:

Function	x_0	α (exact root)
$f_1(x) = 4x^5 - 3x^4 + 2x^3 - 3$	2	1
$f_2(x) = e^x \sin x + log(1 + x^2)$	1.5	0
$f_3(x) = 3\tan x - x$	1	0

Table-1:

f	x_1	x_2	x_3	$\lvert x_3 - x_2\rvert$	COC
$f_1(x)$	$1.1717e+000$	$1.0000e+000$	$1.0000e+000$	$2.1236e-005$	8.92375
$f_2(x)$	$5.0159e-003$	$-1.8309e-019$	$3.6059e-167$	$1.8309e-019$	8.98578
$f_3(x)$	$3.9926e+000$	$4.0781e+000$	$4.0781e+000$	$3.0268e-008$	8.95792

Table-2:

f	GM	HU	HS	$R-1$
$f_1(x)$	$1.8235e-003$	$4.1047e-005$	$6.0517e-009$	$2.1236e-005$
$f_2(x)$	$1.5768e-013$	$6.8481e-009$	$2.4763e-012$	$1.8309e-019$
$f_3(x)$	$1.6744e-006$	$4.5605e-004$	$1.6837e-004$	$3.0268e-008$

Bahman Kalantari [14] coined the term "polynomiography" to be the art and science of visualization in the approximation of roots of polynomial using Iteration Functions. We describe the method to produce the polynomiographs for finding the roots of unity using MATLAB in the following section.

6. Polynomiographs of the ninth order methods for finding the roots of unity

Let $x_0 \in \mathbb{C}$ be the initial point. A square grid of 65536 points, composed of 256 columns and 256 rows corresponding to the pixels of a computer display would represent a region of the complex plane [15]. We consider the square $\mathbb{R} \times \mathbb{R} = [-2, 2] \times [-2, 2]$. Each grid point is used as a starting value x_0 of the sequence generated by the ninth order methods and the number of iterations until convergence is counted for each gridpoint. We consider the polynomial $f(x) = x^r - 1$, $x \in \mathbb{C}$ for finding the r^{th} roots of unity. The r^{th} roots of unity are given by

$$\alpha_j = \cos\left(\frac{2\pi(j-1)}{r}\right) + i \sin\left(\frac{2\pi(j-1)}{r}\right), \ j = 1, 2...r.$$

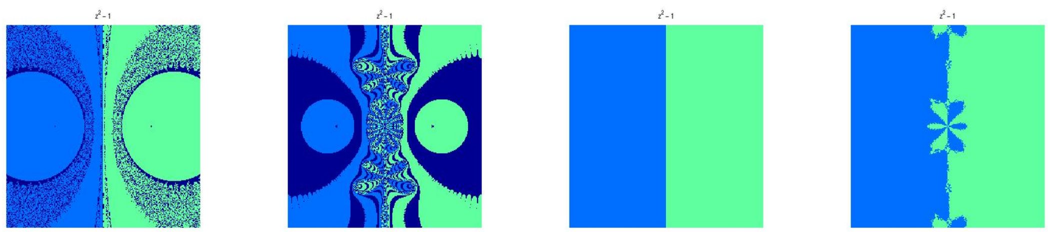
The basin of attraction corresponding to a zero α_j of the polynomial $f(x)$ is the set of all starting points x_0 which are attracted to α_j. If the sequence generated by iterative method attempt a zero α_j of the polynomial with a tolerance $|x_k - \alpha_j| < 1e-4$ and a maximum of 100 iterations, we decide that x_0 is in the basin of attraction of these zero and assign a color to that zero. In this way, the basin of attraction for each root would be assigned a characteristic colour. The common boundaries of these basins of attraction constitute the Julia set of the Iteration Function. If the iterates do not satisfy the above criterion for convergence we assign the dark blue colour (The iterates either diverge or converge to additional fixed points). Let us denote N_d as the number of diverging points and μ as the mean number of iterations for the starting points. We choose $r = 2, 3, 4, 5$ for our numerical experiments.

Table-3:

$Method$	$r=2$		$r=3$		$r=4$		$r=5$	
	N_d	μ	N_d	μ	N_d	μ	N_d	μ
GM	10893	2.68	2976	3.15	2954	4.21	18806	9.39
HU	28360	3.70	8778	10.1	25702	17.5	12360	16.5
HS	0	2.86	0	3.22	430	4.32	0	3.77
$R-1$	0	3.78	0	6.14	358	11.5	4000	13.0

Table 3 show that the new method R-1 performs better than the HU and GM methods because it is globally convergent for the cases $r = 2$ and $r = 3$ and our method is the most efficient method for the case $r = 4$ with the smallest number of diverging points. Our method has many diverging points for the case $r = 5$ compared to the HS with no diverging points. However, the HS method requires the calculation of second derivative.

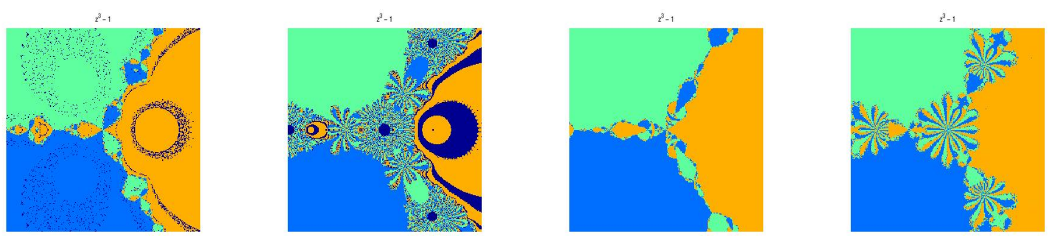
Figure 1: Polynomiographs for $f(x) = x^2 - 1$



(a) GM (b) HU (c) HS (d) R-1

Fig. 1 shows the polynomiographs of the 4 ninth order methods for the quadratic polynomial with roots 1 (green) and -1 (blue). The Julia set for the HS is the imaginary axis. The behaviour of the R-1 method is less dynamic than the HU and GM methods.

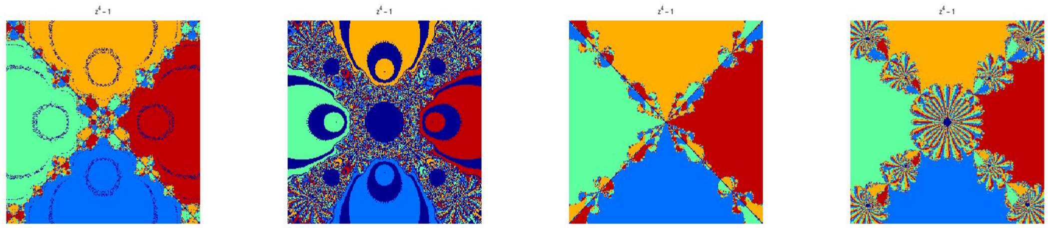
Figure 2: Polynomiographs for $f(x) = x^3 - 1$



(a) GM (b) HU (c) HS (d) R-1

Fig. 2 shows the polynomiographs of the 4 ninth order methods for the cubic polynomial with roots 1 (orange), $-0.5000 - 0.8660i$ (blue) and $-0.5000 + 0.8660i$ (green). The behaviour of the R-1 method is less dynamic than the HU and GM methods.

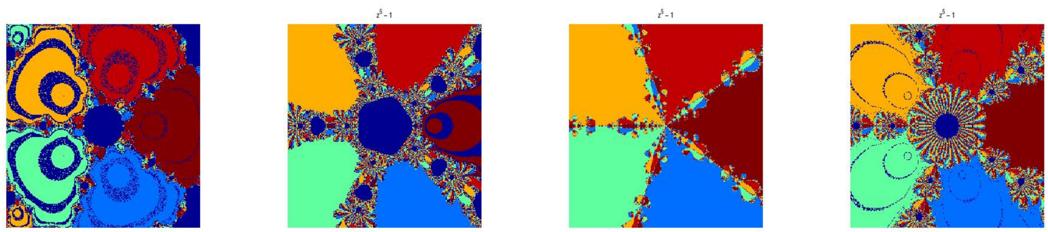
Figure 3: Polynomiographs for $f(x) = x^4 - 1$



(a) GM (b) HU (c) HS (d) R-1

Fig. 3 shows the polynomiographs of the 4 ninth order methods for the quartic polynomial with roots 1 (reddish-brown), -1 (green), i (orange) and $-i$ (blue). The behaviour of the R-1 method is less dynamic than the HU and GM methods.

Figure 4: Polynomiographs for $f(x) = x^5 - 1$



(a) GM (b) HU (c) HS (d) R-1

Fig. 4 shows the polynomiographs of the 4 ninth order methods for the quintic polynomial with roots $0.3090 + 0.9511i$ (reddish-brown), $0.3090 - 0.9511i$ (blue) , $-0.8090 + 0.5878i$ (orange), $-0.8090 - 0.5878i$ (green) and 1 (dark brown). The behaviour of the R-1 method is less dynamic than the HU and GM methods.

7. Conclusions

In this paper we have proposed a new ninth order iterative method. The convergence order of the suggested method is proved, the efficiency is measured and the computational order of convergence (COC) is also calculate. With the help of some numerical test problems, comparison of the obtained results with the existing methods such as the GM, HU and HS is also given and it is observed that the new method is efficient in many cases as compared to the existing methods. The dynamic of the methods for finding the roots of unity are also studied.

References

[1] J. Biazar, A. Amirteimoori, *An improvement to the fixed point iterative method*, Applied mathematics and computation, 182 (2006) pp. 567-571.

[2] R. L. Burden, J. D. Faires , *Numerical analysis 8th ed*, Brooks-Cole, 2004.

[3] L. Hou and X. Li, *Twelfth-order method for nonlinear equation*, International Journal of Research and Reviews in Applied Sciences, 3 (1) (2010), pp. 30-36.

[4] Z. Hu, L. Guocai, L. Tian , *An iterative method with ninth-order convergence for solving nonlinear equations*, Int. J. contemp. Math. Science, 6(1) (2011) 17-23.

[5] M. K. Jain,S. R. K. Iyengar, R. K Jain, *Numerical Methods For Scientic and Engineering Computation, 4th edition*, New age international publishers, 2003.

[6] A.M.Ostrowski *Solutions of equations and system of equations.* New York: Academic Press; 1960.

[7] M. Rafiullah, *A fifth-order iterative method for solving nonlinear equations*, Numerical Analysis and Applications, 4(3) (2011), pp. 239–243.

[8] J.F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, 1964.

[9] F. Ahmed, *Elements of Numerical Analysis*, National book foundation Pakistan, 2001.

[10] S. Weerakoon, T.G.I. Fernando, *A variant of Newton's method with accelerated third-order convergence*, Appl. Math. Lett. 13(2000) 87–93.

[11] M.A. Hafiz, Salwa M. H. Al-Goria, *New ninth and seventh order methods for solving nonlinear eqautions*, European Scientific Journal November edition, 8(27) (2012), PP. 83-95.

[12] G.V. Milovanov'c and A.S. Cvetkovi'c, *A note on three-step iterative methods for nonlinear equations*, Studia Univ. "Babes-Bolyai", Mathematica, Vol. LII, No. 3, September 2007.

[13] Y. I. Kim and C. Chun, *New Twelfth-order modification of Jarratt's method for solving nonlinear equations*, Studies in Nonlinear Sciences, 1(1) 2010, 14-18.

[14] B. Kalantari, *Polynomial root-finding and polynomiography*, World Scientific Publishing Co. Pte. Ltd. Singapore, 2009.

[15] E. R. Vrscay, *Julia Sets and Mandelbrot-Like Sets Associated With Higher Order Schröder Rational Iteration Functions: A Computer Assisted Study*, Math. Comp. 46(1986), 151–169.

Muhammad Rafiullah
Dept. of Mathematics,
COMSATS Institute of Information Technology,

Lahore, Pakistan
email: *mrafiullah@ciitlahore.edu.pk, rafiullaharain@gmail.com*

D. K. R. Babajee
African Network for Policy Research & Actions for Sustainability (ANPRAS)
Mauritius
email: *dkrbabajee@gmail.com*

Dur-e-Jabeen
Faculty of Engineering Science and Technology,
Hamdard Univeristy,
Karachi, Pakistan
email: *durejabeen@hotmail.com*