SOME NOVEL SCHEMES BY USING MULTIPLICATIVE CALCULUS FOR NONLINEAR EQUATIONS

F. A. SHAH[1*], E. UL-HAQ[2], M. A. NOOR[1], M. WASEEM[1]

ABSTRACT. In this paper, we suggest and analyze a new family of two-step predictor corrector type iterative schemes for solving nonlinear equations in the framework of multiplicative calculus. We also discuss the convergence criteria of these newly developed iterative methods. Some numerical examples will be given to illustrate the efficiency and performance of derived methods.

Keywords: Iterative methods, Nonlinear equations, Convergence, multiplicative calculus.

AMS Subject Classification: 83-02, 99A00

1. INTRODUCTION

Differential and integral calculus were created independently by Sir Isaac Newton and Gottfried Wilhelm Leibnitz. Differentiation and integration are the basic operation in calculus and analysis. From 1967 to 1970 Michael Grossman and Robert Katz introduced the so called Bi-geometric calculus , where they defined a new kind of derivative and integral, moving the roles of subtraction and addition to division and multiplication, and thus established a new calculus, called Multiplicative Calculus. The theoretical background of multiplicative was given by A.E. Bashirov [2,3] . Sometimes, it is called an alternative or non-Newtonian calculus as well. As multiplicative calculus is the Taylor-made calculus for growth related problem, that are modeled in science and engineering using the exponential function. In recent years multiplicative calculus is applying in various fields see detail $[1-16]$. It is more than self-evident to use multiplicative calculus also for numerical approximations The nonlinear equation $f(x) = 0$ have important and significant applications in various field of science. Various techniques have been developed in simple calculus to solving this problem. In this work, we will develop Householder-type iterative methods for

---

[1] Comsats University, Islamabad, Attock Campus, Pakistan.
e-mail: farooqhamdani@gmail.com; ORCID: https://orcid.org/0000-0001-7419-423X.
* Corresponding author.
e-mail:noormaslam@hotmail.com; ORCID: https://orcid.org/0000-0001-6105-2435.
e-mail:sattarmwaseem@gmail.com; ORCID: https://orcid.org/0000-0001-5903-0122.
[2] University of Wah, Pakistan.
e-mail: ehsan.shah61@gmail.com; ORCID: https://orcid.org/0000-0001-8266-449X .

solving nonlinear equation in the framework of multiplicative calculus. To construct these methods we will use auxiliary function, which will help to implement properly the iterative schemes. At the end of the article, some examples are given to clarify the application of this work and also to check the performance of these new methods and their comparison with other existing methods.

Derivation of Multiplicative Iterative Methods

In this section, we describe some basics results in the field of multiplicative calculus. Using the idea of multiplicative calculus, we construct the iterative schemes for nonlinear equation in this framework.

Multiplicative derivative.

The function $f : D \subset \mathbb{R} \to \mathbb{R}$ is said to be $*$ differentiable at x or on D if it is positive and differentiable, respectively, at x or on D, and the multiplicative derivative satisfies the following limit:

$$f^*(x) = \frac{d^*f}{dx} = \lim_{h \to 0} \left[\frac{f(x+h)}{f(x)} \right]^{\frac{1}{h}} \tag{1}$$

Using (1), we can calculate the multiplicative derivative of function as:

$$f^*(x) = e^{\frac{f'(x)}{f(x)}} = e^{(\ln of)'x}. \tag{2}$$

Where $(\ln of)(x) = \ln f(x)$

$$f^{**}(x) = e^{(\ln of^*)'x} = e^{(\ln of)''x}. \tag{3}$$

In general, it can be described as:

$$f^{*(n)}(x) = e^{(\ln of)^n x}, n = 1, 2, 3, \dots \tag{4}$$

Note that in formula (4), the case $n = 0$, means no multiplicative derivative and it represents the original function.

Some rules of * differentiation.

Let c be a positive constant, f, g are $*$ differentiation and h is differentiable, then the following results hold:

(1) (a) $(c)^* = 1$,
 (b) $(cf)^*(x) = f^*(x)$,
 (c) $(fg)^*(x) = f^*(x)g^*(x)$,
 (d) $\left(\frac{f}{g} \right)^*(x) = \frac{f^*(x)}{g^*(x)}$,
 (e) $(f^h)^*(x) = f^*(x)^h . f(x)^{h'(x)}$,
 (f) $(foh)^*(x) = f^*(h(x))^{h'(x)}$, where foh is defined

Now, we develop the iterative methods using above results

A function $g : D \subset \mathbb{R} \to \mathbb{R}^+$ be a positive nonlinear function.

$$g(x) = 1 \tag{5}$$

We remark that equation (5) is a nonlinear equation in the framework of multiplicative calculus.

we can use the following multiplicative fixed point relation for (5), we have

$$e^{\psi(x)} e^{[lng(x)]^p \lambda \phi(x)} = e^x, \tag{6}$$

where $p \geq 1$ is order of convergence of iteration function $\psi(x)$. $\phi(x)$ be an auxiliary arbitrary function. We consider the following relation

$$e^x = G(x) \tag{7}$$

By using the multiplicative optimal criteria as $\frac{d*G(x)}{dx} = 1$, then we have

$$e^{\psi'(x)} e^{[lng(x)]^p \lambda \phi'(x)} e^{plng*(x)[lng(x)]^{(p-1)}\lambda \phi'(x)} = 1, \tag{8}$$

where λ is a parameter which is usually called the Lagrange multiplier. Taking natural log of (8), we obtain λ as:

$$\lambda = \frac{-\psi'(x)}{p \ln(g^*(x))[\ln(g(x))]^{p-1}\phi(x)\ln(g(x)) + \phi'(x)\ln(g(x))]} \tag{9}$$

Taking natural log of (6) and using (9), we can define following new fixed point formulation.

$$x = \psi(x) - \frac{\psi'(x)[\ln(g(x))\phi(x)]}{p\phi(x)\ln(g^*(x)) + \phi'(x)\ln(g(x))} \tag{10}$$

We would like to emphasize the fact that $\psi(x)$ can be selected as a predictor suggesting the iterative method.

For simplicity, we consider auxiliary function as;

$$\psi(x) = x - \frac{\ln(g(x))}{\ln(g^*(x))} \tag{11}$$

Then

$$\psi'(x) = \frac{\ln(g(x))\ln(g^{**}(x))}{[\ln(g^*(x))]^2} \tag{12}$$

By using (11) and (12) in (10), we obtain the following iterative method in the framework of multiplicative calculus.

Algorithm 2.1. For a given initial value x_0, approximate solution x_{n+1} of $g(x)$ by the following iterative scheme:

$$x_{n+1} = x_n - \frac{\ln(g(x_n))}{\ln(g^*(x_n))} - \frac{[\ln(g(x_n))]^2 \ln(g^{**}(x_n))\phi(x_n)}{[\ln(g^*(x_n))]^2[2\phi(x_n)\ln(g^*(x_n)) + \phi'(x_n)\ln(g(x_n))]}$$

For different values of the auxiliary function $\phi(x_n)$, we can obtain several Householder type iterative methods for solving nonlinear equations. Fotr the implement we consider have only two cases.
1. Let $\phi(x_n) = e^{\beta(x_n)}$. Then from Algorithm 2.1, we obtain the following iterative method for solving the nonlinear equation (5).

Algorithm 2.2. For a given initial value x_0, approximate solution x_{n+1} of $g(x)$ by the following iterative scheme:

$$x_{n+1} = x_n - \frac{\ln(g(x_n))}{\ln(g^*(x_n))} - \frac{[\ln(g(x_n))]^2 \ln(g^{**}(x_n))}{[\ln(g^*(x_n))]^2[2\ln(g^*(x_n)) + \beta \ln(g(x_n))]}$$

If $\beta = 0$, then Algorithm 2.2 reduces into given Algorithm .

Algorithm 2.3. For a given initial value x_0, find the approximate solution x_{n+1} of $g(x)$ by the following iterative scheme:

$$x_{n+1} = x_n - \frac{\ln(g(x_n))}{\ln(g^*(x_n))} - \frac{[\ln(g(x_n))]^2 \ln(g^{**}(x_n))}{2[\ln(g^*(x_n))]^3}$$

2. Let $\phi(x_n) = e^{\frac{-\beta}{\ln(g(x_n))}}$ Then from Algorithm 2.1, we obtain the following iterative method for solving nonlinear equations.

Algorithm 2.4. For a given initial value x_0, approximate solution x_{n+1} of $g(x)$ by the following iterative scheme:

$$x_{n+1} = x_n - \frac{\ln(g(x_n))}{\ln(g^*(x_n))} - \frac{[\ln(g(x_n))]^2 \ln(g^{**}(x_n))}{2[\ln(g^*(x_n))]^3 - \beta \ln(g(x_n)) \ln(g**(x_n))}$$

We replace the approximation of second derivative in Algorithms 2.2 and 2.4 by a suitable substitution involving only the first derivative and obtain predictor-corrector type iterative methods.

we apply multiplicative Taylor theorem to remove second $*$ derivative, we have

$$g(y) \approx (g(x))(g^*(x))^{(y-x)}(g^{**}(x))^{\frac{(y-x)^2}{2}} \tag{13}$$

From (13), we have

$$\ln(g(y)) = \frac{1}{2} \frac{[\ln(g(x))]^2 \ln(g^{**}(x))}{[\ln(g^*(x))]^2} \tag{14}$$

Using (14) we have the following iterative multiplicative iterative method.

Algorithm 2.5. For a given initial value x_0, find the approximate solution x_{n+1} of $g(x)$ by the following iterative schemes:

$$y_n = x_n - \frac{\ln(g(x_n))}{\ln(g^*(x_n))}$$

$$x_{n+1} = y_n - \frac{2\ln(g(y_n))}{[2\ln(g^*(x_n)) + \beta \ln(g(x_n))]}, n = 0, 1, 2, ...$$

Algorithm 2.6. For a given initial value x_0, find the approximate solution x_{n+1} of $g(x)$ by the following iterative schemes:

$$y_n = x_n - \frac{\ln(g(x_n))}{\ln(g^*(x_n))}$$

$$x_{n+1} = y_n - \frac{\ln(g(x_n)) \ln(g(y_n))}{[\ln(g^*(x_n)) \ln(g(x_n)) + \beta \ln(g(y_n))]}, n = 0, 1, 2, ...$$

CONVERGENCE ANALYSIS

In this section, we study the multiplicative convergence criteria of methods developed in section 2. For study the convergence we use Multiplicative Taylor's theorem [2].

Theorem 1.1. *Consider $g : D \subset \mathbb{R} \to \mathbb{R}^+$ be a positive function. Let g be sufficiently $*$ differentiable and differentiable function at all $x \in D$. Assume that there exists a solution $\alpha \in [a, b]$ of nonlinear equation such that $g(\alpha) = 1$. Then there exists a $\delta > 0$, such that the iteration scheme defined by Algorithm 2.5 converges to the solution α for any initial approximation $x_0 \in [\alpha - \delta, \alpha + \delta]$ and its rate of convergence is three.*

Proof. Consider the iteration defined by Algorithm 2.5

$$x_{n+1} = y_n - \frac{2\ln(g(y_n))}{[2\ln(g^*(x_n)) + \beta \ln(g(x_n))]}. \tag{15}$$

Where

$$y_n = x_n - \frac{\ln(g(x_n))}{\ln(g^*(x_n))} \tag{16}$$

From (16) by taking $x_n - \alpha = e_n, y_n - \alpha = e_n^*$, we have

$$e_n^* - e_n = -\frac{\ln(g(x_n))}{\ln(g^*(x_n))}. \tag{17}$$

From (17) we can define

$$\ln[(g^*(x_n))^{e_n^* - e_n}(g(x_n))] = 0, \tag{18}$$

By applying natural exponential on both sides of (18)

$$(g^*(x_n))^{e_n^* - e_n}(g(x_n)) = 1, \tag{19}$$

From (19) we obtain

$$(g^*(x_n))^{e_n^*} = \frac{(g^*(x_n))^{e_n}}{(g(x_n))} \tag{20}$$

We expand $g(x_n)$, using the following Taylor's series for multiplicative calculus:

$$g(x_n) \approx g(\alpha)(g^*(\alpha))^{x_n - \alpha}(g^{**}(\alpha))^{\frac{(x_n-\alpha)^2}{2!}}(g^{***}(\alpha))^{\frac{(x_n-\alpha)^2}{3!}} \tag{21}$$

Using $g(\alpha) = 1$ in (21), we have

$$g(x_n) \approx (g^*(\alpha))^{e_n}(g^{**}(\alpha))^{\frac{e_n^2}{2!}}(g^{***}(\alpha))^{\frac{e_n^3}{3!}}. \tag{22}$$

Similarly, by expanding $g^*(x_n)$ about α, we have

$$g*(x_n) \approx g^*(\alpha)(g^{**}(\alpha))^{e_n}(g^{***}(\alpha))^{\frac{e_n^2}{2!}}(g^{****}(\alpha))^{\frac{e_n^3}{3!}}. \tag{23}$$

Using (22) and (23) in (20), we get

$$\left[g^*(\alpha)(g^{**}(\alpha))^{e_n}(g^{***}(\alpha))^{\frac{e_n^2}{2!}}\right]^{e_n^*} = \frac{\left[g^*(\alpha)(g^{**}(\alpha))^{e_n}(g^{***}(\alpha))^{\frac{e_n^2}{2!}}\right]^{e_n^*}}{(g^*(\alpha))^{e_n}(g^{**}(\alpha))^{\frac{e_n^2}{2!}}(g^{***}(\alpha))^{\frac{e_n^3}{3!}}} \tag{24}$$

From (24) we obtain

$$e_n^* = e_n^2 \frac{\ln\left[(g^{**}(\alpha))^{\left(\frac{1}{2}\right)}(g^{***}(\alpha))^{\frac{e_n^2}{3!}}\right]}{\ln\left[(g^*(\alpha))^{e_n}(g^{**}(\alpha))^{e_n}(g^{***}(\alpha))^{\frac{e_n^2}{2!}}\right]} \tag{25}$$

From (25), we have following results

$$e_n^* = T e_n^2 \tag{26}$$

Where $T > 0$, is given by

$$T = \frac{\ln\left[(g^{**}(\alpha))^{\left(\frac{1}{2}\right)}(g^{***}(\alpha))^{\frac{e_n^2}{3!}}\right]}{\ln\left[(g^*(\alpha))^{e_n}(g^{**}(\alpha))^{e_n}(g^{***}(\alpha))^{\frac{e_n^2}{2!}}\right]}$$

Now we implies (15)

$$x_{n+1} = y_n - \frac{2\ln(g(y_n))}{[2\ln(g^*(x_n)) + \beta \ln(g(x_n))]}. \tag{27}$$

First we expand using multiplicative $g(y_n)$ Taylor series as;

$$g(y_n) \approx g(\alpha)(g^*(\alpha))^{y_n - \alpha}(g^{**}(\alpha))^{\frac{(y_n - \alpha)^2}{2!}}(g^{***}(\alpha))^{\frac{(y_n - \alpha)^2}{3!}} \qquad (28)$$

Using $g(\alpha) = 1$ in (25), we have

$$g(y_n) \approx (g^*(\alpha))^{e_n}(g^{**}(\alpha))^{\frac{e_n^2}{2!}}(g^{***}(\alpha))^{\frac{e_n^3}{3!}}. \qquad (29)$$

From (27) by taking $x_n - \alpha = e_n, y_n - \alpha = e_n^*$ and $x_{n+1} - \alpha = e_{n+1}$, we have

$$e_{n+1} - e_n^* = -\frac{\ln(g(y_n^2))}{[\ln(g^*(x_n))^2) + \beta \ln(g(x_n))]}, \qquad (30)$$

We can write (30) in the following form;

$$\ln[(g^*(x_n))^2(g(x_n))^\beta)^{e_{n+1} - e_n^*}(g(y_n))^2] = 0, \qquad (31)$$

By applying natural exponential on both sides (31), we have

$$((g^*(x_n))^2(g(x_n))^\beta)^{e_{n+1} - e_n^*}(g(y_n))^2 = 1 \qquad (32)$$

Then

$$((g^*(x_n)^2)(g(x_n))^\beta)^{e_{n+1}} = \frac{((g*(x_n))^2(g(x_n))^\beta)^{e_n^*}}{(g(y_n))^2} \qquad (33)$$

Using (19), (22),(23) in (30), we get

$$\left[(g^*(\alpha))^{(1+ke_n)}(g^{**}(\alpha))^{\left(e_n + \frac{ke_n^2}{2!}\right)}(g^{***}(\alpha))^{\left(\frac{e_n^2}{2!} + \frac{ke_n^3}{3!}\right)}\right]^{e_{n+1}} \qquad (34)$$

$$= \frac{\left[(g^*(\alpha))^{(1+ke_n)}(g^{**}(\alpha))^{\left(e_n + \frac{ke_n^2}{2!}\right)}(g^{***}(\alpha))^{\left(\frac{e_n^2}{2!} + \frac{ke_n^3}{3!}\right)}\right]^{e_n^*}}{(g^*(\alpha))^{e_n^*}(g^{**}(\alpha))^{\frac{(e_n^*)^2}{2!}}(g^{***}(\alpha))^{\frac{(e_n^*)^3}{3!}}}$$

Simplifying (34), we can write

$$\left[(g^*(\alpha))^{(1+ke_n)}(g^{**}(\alpha))^{\left(e_n + \frac{ke_n^2}{2!}\right)}(g^{***}(\alpha))^{\left(\frac{e_n^2}{2!} + \frac{ke_n^3}{3!}\right)}\right]^{e_{n+1}} \qquad (35)$$

$$= \left[(g^*(\alpha))^{(ke_n)}(g^{**}(\alpha))^{\left(e_n + \frac{ke_n^2}{2!} - \frac{(e_n^*)^2}{2!}\right)}(g^{***}(\alpha))^{\left(\frac{e_n^2}{2!} + \frac{ke_n^3}{3!} + \frac{(e_n^*)^3}{3!}\right)}\right]^{e_n^*}$$

Since $(k = -\beta)$, we use the value of e_n^* given in (35). Use these all values in (35). Then also applying natural log on both sides, we have

$$e_{n+1} = Te_n^3 \frac{\ln\left[(g^*(\alpha))^{(-\beta)}(g^{**}(\alpha))^{\left(1 - \frac{\beta e_n}{2!} - \frac{Te_n}{2!}\right)}(g^{***}(\alpha))^{\left(\frac{e_n}{2!} - \frac{\beta e_n^3}{3!} - \frac{T^2 e_n^3}{3!}\right)}\right]}{\ln\left[(g^*(\alpha))^{(1 - \beta e_n)}(g^{**}(\alpha))^{\left(e_n - \frac{\beta e_n^2}{2!}\right)}(g^{***}(\alpha))^{\left(\frac{e_n^2}{2!} - \frac{\beta e_n^3}{3!}\right)}\right]} \qquad (36)$$

where

$$Q = T \frac{\ln\left[(g^*(\alpha))^{(-\beta)} (g^{**}(\alpha))^{\left(1 - \frac{\beta e_n}{2!} - \frac{T e_n}{2!}\right)} (g^{***}(\alpha))^{\left(\frac{e_n}{2!} - \frac{\beta e_n^3}{3!} - \frac{T^2 e_n^3}{3!}\right)} \right]}{\ln\left[(g^*(\alpha))^{(1 - \beta e_n)} (g^{**}(\alpha))^{\left(e_n - \frac{\beta e_n^2}{2!}\right)} (g^{***}(\alpha))^{\left(\frac{e_n^2}{2!} - \frac{\beta e_n^3}{3!}\right)} \right]}$$

From (36), final error equation will be

$$e_{n+1} = Q e_n^3 \tag{37}$$

Relation (37) show that Algorithm 2.5 is third order convergent iterative method. Similarly, we can observe the convergence criteria for other methods. □

NUMERICAL RESULTS

In this section, we will show the performance of newly derived methods developed in section 2. To observe the performance these new methods we will compare it with some already existing methods in literature as following.

(1) Existing method described in [5,7] by using Variational iteration method.

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

(2) Classical Newton method[4]:

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

(3) Halley method[6] :

$$x_{n+1} = x_n - \frac{2f'(x_n)f(x_n)}{2(f'(x_n))^2 + f(x_n)f''(x_n)}$$

We use Intel core i3 Due CPU 2 GHz computer (window 8.1 pro 64-bit). We use Maple for all computational work with 150 digits floating point arithmetic (digits: =150). For the computer programs, we use stopping criteria $|g(x_n) - 1| < \epsilon$ or multiplicative methods and $|f(x_n - 0)| < \epsilon$ for the ordinary methods, where $\epsilon = 10^{-20}$, we approximate the computational order of convergence (COC)

$$COC \approx \frac{\ln(|x_{n+1} - x_n|/|x_n - x_{n-1}|)}{\ln(|x_n - x_{n-1}|/|x_{n-1} - x_{n-2}|)},$$

Example 1.1. *(Application to population growth model). Consider the nonlinear equation which arises in mathematical modeling of the growth of population over short periods of time:*

$$f(x) = 1,000,000e^x + \frac{435,000}{x}(e^x - 1) - 1564,000 = 0$$

Where x denotes the constant birth rate of the population and whose value is needed to find. One can rewrite the nonlinear equation in the following equivalent form of multiplicative nonlinear equation:
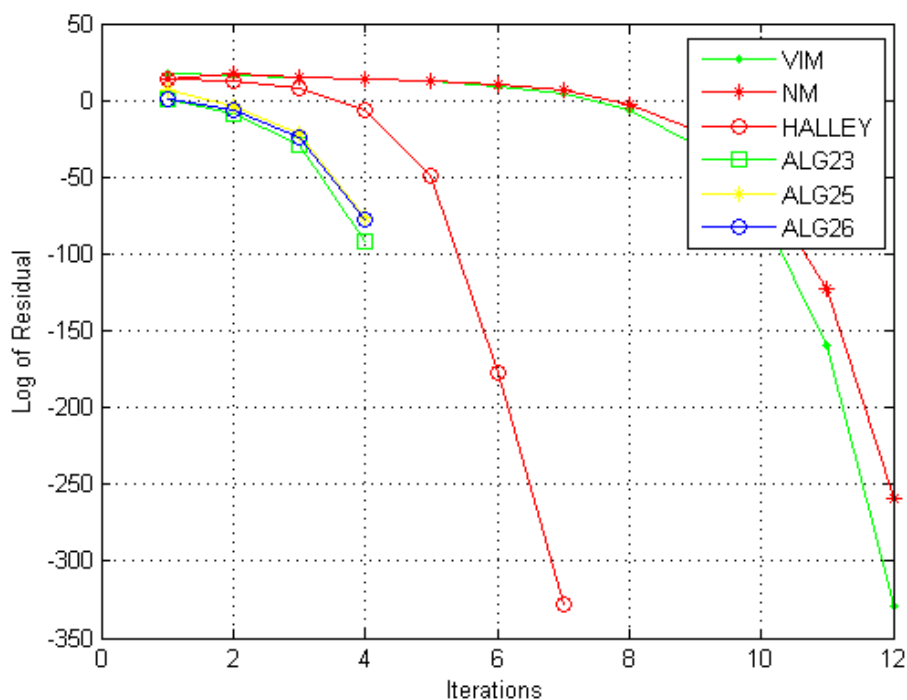
$$g(x) = \frac{1000}{1564}e^x + \frac{435}{1564}(e^x - 1) = 1$$

We use initial guess is $x_0 = 1$. We also chose $\beta = 0.01$ for ordinary VIM, Algorithm 2.3 ,Algorithm 2.5 and Algorithm 2.6. Performance of these methods is shown in below table

Table 4.1

Method	IT	x_{n+1}	Rel.Err	COC
VIM	11	0.10099792	$5.32e^{-77}$	2.00000
NM	11	0.10099792	$2.66e^{-44}$	2.00000
Halley	7	0.10099792	$2.00e^{-143}$	2.99974
Alg2.3	4	0.10099792	$3.22e^{-54}$	2.99989
Alg2.5	3	0.10099792	$6.85e^{-41}$	3.00000
Alg2.6	3	0.10099792	$5.44e^{-45}$	3.00000

Now we check efficiency of these methods graphically. Here we need log of residual for every iteration. Graphically comparisons is Shown is figure 4.1.

Figure 4.1 Log of Residuals for Example 4.1

Example 1.2. *Consider the nonlinear equation:*

$$f(x) = \left(e^{x^2+7x-30} - 1 \right) = 0$$

One can rewrite the nonlinear equation in the following equivalent as multiplicative nonlinear equation:
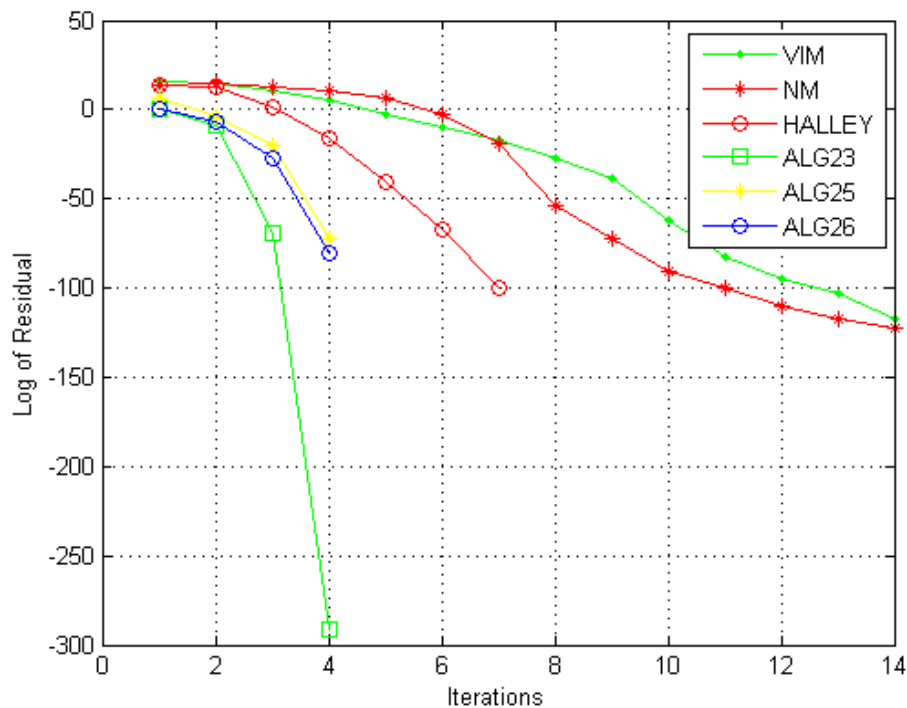
$$g(x) = e^{x^2+7x-30} = 1$$

We use initial guess as 3.5 . We also chose $\beta = 0.01$ for ordinary VIM, Algorithm 2.3 ,Algorithm 2.5 and Algorithm 2.6. Performance of these methods is shown in below table

Table 4.2

Method	IT	x_{n+1}	Rel.Err	COC
VIM	13	3.00000	$1.47e^{-49}$	2.00000
NM	13	3.00000	$1.52e^{-47}$	2.00000
Halley	7	3.00000	$2.56e^{-44}$	2.00032
Alg2.3	4	3.00000	$1.94e^{-49}$	2.99989
Alg2.5	3	3.00000	$4.34e^{-33}$	2.97315
Alg2.6	3	3.00000	$1.94e^{-114}$	2.99999

Figure 4.2 Log of Residuals for Example 4.2

Example 1.3. *You are working for a start-up computer assembly company and have been asked to determine the minimum number of computers that the shop will have to sell to make a profit. The equation that gives the minimum number of computers x to be sold after considering the costs and the total sales is:*

$$f(x) = 40x^{1.5} - 875x + 35000 = 0$$

One can rewrite the nonlinear equation in the following equivalent as multiplicative nonlinear equation

$$g(x) = \frac{4}{3500}x^{1.5} - \frac{875}{35000}x = 1$$

We use initial guess is 60. We also chose $\beta = 0.01$ for ordinary VIM, Algorithm 2.3 ,Algorithm 2.5 and Algorithm 2.6. Performance of these methods is shown in below table

Table 4.3

Method	IT	x_{n+1}	Rel.Err	COC
VIM	11	62.6916715	$5.32e^{-77}$	2.00000
NM	9	62.6916715	$3.37e^{-117}$	2.00000
Halley	7	62.6916715	$2.56e^{-76}$	2.99999
Alg2.3	4	62.6916715	$1.39e^{-21}$	2.69174
Alg2.5	3	62.6916715	$1.38e^{-26}$	2.97690
Alg2.6	3	62.6916715	$1.15e^{-42}$	2.99650

62.6916715 is the minimum number of computers to be sold
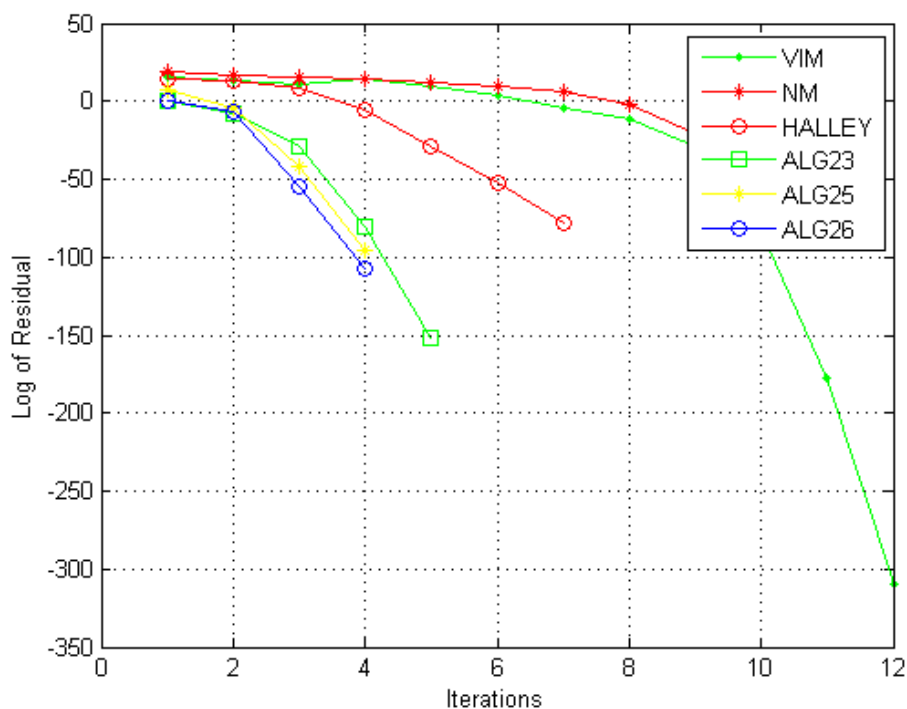
Figure 4.3 Log of Residuals for Example 4.3

Results and discussion

In this article we developed some novel techniques by using multiplicative calculus. These techniques are much efficient rather then existing methods and much effective for some problems. Convergence criteria of developed methods is studied. Numerical results are also calculated and comparison is also exhibited. Graphical and numerical results shows the performance of methods. All numerical and graphical computation is made by using MAPLE and MATLAB.

References

[1] Aniszewska, D., (2007), Multiplicative runge–kutta methods. Nonlinear Dynamics, 50(1-2), 265-272.
[2] Bashirov, A. E., Kurpınar, E. M., Ozyapıcı,A., (2008), Multiplicative calculus and its applications, Journal of Mathematical Analysis and Applications, 337(1), 36-48.
[3] Bashirov, A. E., Mısırlı, E., Tandogdu, Y., Ozyapıcı, A., (2011), On modeling with multiplicative differential equations, Applied Mathematics-A Journal of Chinese Universities, 26(4), 425-438.

[4] Burden, R. L., Faires, J.D., (2001), Numerical Analysis, PWS Publishing Company, Bostan, USA.
[5] Cordova-Lepe, F., (2006), The multiplicative derivative as a measure of elasticity in economics, TMAT
     Revista Latinoamericana de Ciencias e Ingenieria, 2(3).
[6] Chen, T., Hongyu L He., Church, G. M., (1999), Modeling gene expression with differential equations,
     In Pacific symposium on biocomputing, 29-40.
[7] Daftardar-Gejji, V., Jafari, H., (2006), An iterative method for solving nonlinear functional equations,
     J. Math. Anal. Appl., 316, 753–763.
[8] Riza, M., Ozyapici, A., Misirli, E., (2009), Multiplicative finite difference methods, Quart. Appl. Math.,
     67(4), 745-754.
[9] Grossman, M., and Katz, R., (1972), Non-Newtonian calculus, Lee Press, Pigeon Cove, Mass.
[10] Grossman, M., (1983), Bigeometric calculus, Archimedes Foundation, Rockport, Mass, A system with
     a scale-free derivative .
[11] He, J. H., (2007), Variational iteration method-some recent result sandne winter pretations, J. Comp.
     Appl. Math., 207, 3-11.
[12] Johnson, L. W., Riess, R. D., Numerical Analysis, Reading, MA:Addison-Wesley, (1977).
[13] Noor, M. A., Shah, F. A., Noor, E. Al-said, K. I., (2011), Variational iteration technique for finding
     multiple roots of nonlinear equations, Sci. Res. Essays., 6(6), 1344-1350.
[14] Noor, M. A., Shah, F. A., Noor, K. I., Al-said, E., (2011), Variational iteration technique for finding
     multiple roots of nonlinear equations, Sci. Res. Essays., 6(6), 1344-1350.
[15] Noor, K. I., Noor, M. A.,(2007), Predictor-corrector Halley method for nonlinear equations, Appl.
     Math. Comput., 188, 1587-1591.

Dr. Farooq Ahmed Shah is an assistant professor in the Department of Mathematics (CUI ATTOCK) He earned his PhD degree from Comsats Institute of Information Technology in the field of Numerical Analysis and Optimization (Applied Mathematics). His area of interest in Numerical Analysis is iterative methods for solving linear and nonlinear equations.

Ehsan Ul Haq obtained his MSc degree from Comsats University in the field of Numerical Analysis.The area of his research is analysis of nonlinear equations arising in physical sciences.

Muhammad Aslam Noor for the photography and short autobiography, see TWMS J. App. and Eng. Math. V.11, N.1.

Muhammad Waseem received his Ph.D. degree from CIIT Islamabad and the Ph.D. degree in applied mathematics (numerical analysis). He is currently an assistant professor in the Department of Mathematics, Comsats University Islamabad. His current research interests include numerical analysis to construct iterative schemes for linear and nonlinear equations.