Proceedings of the Pakistan Academy of Sciences:
A. Physical and Computational Sciences: 57 (4): 77-91 (2020)

ISSN: 2518-4245 (print), 2518-4253 (online)

Pakistan Academy of Sciences

Research Article

Some Efficient Algorithms to Raise Order of Convergence of Iterative Methods for Nonlinear Equations with Applications

Hameer Akhtar Abro[1], and Muhammad Mujtaba Shaikh[1,2*]

[1]Department of Basic Sciences and Related Studies, Mehran University of Engineering and Technology, Jamshoro
[2]Supply Chain and Operations Management Research Group, Mehran University of Engineering and Technology, Jamshoro

Abstract: Two algorithms to raise the order of convergence of nonlinear solvers for scalar nonlinear equations are proposed. The suggested algorithms comprise of three steps: first two being any existing iterative method for nonlinear equations with nth order convergence, and the proposed third step being free from any new derivative. The third step in the proposed algorithms uses two different divided difference approximations to replace new derivatives. The order of convergence is raised to n + 3 and n + 4, respectively, when the third steps are combined with any two-step order convergent method. The extension in orders of convergence of methods is proved theoretically. As an application of proposed algorithms, proposed third steps are combined with some well-known existing third, fourth and fifth order two-step methods from literature. The consequent improvement in order of convergence is justified for five new methods derived using proposed algorithms. The computational performance of the proposed methods and some other similar order methods from literature are examined on several nonlinear equations of different nature including engineering problems and real mechanical system. All the proposed methods exhibit encouraging performance for test examples, and also for some applied nonlinear equations, like NASA's launched satellite, real mechanical system, catenary cable and thermodynamics application.

Keywords: Nonlinear Equations, Efficient Algorithms, Iterative Methods, CPU Time, Order of Convergence, Computational Efficiency.

1. INTRODUCTION

To find the solutions of nonlinear scalar equations of the form is an important and challenging task in mathematical and engineering problems. Many real life problems are mathematically modeled in the form of these nonlinear equations. For instance, finding temperature of saturation concentration, measuring depth of water in a trapezoidal channel, solving Michaelis-Menten equation used in kinetics of enzyme mediated reactions, finding number of moles of chemical in a reversible chemical reaction, computing point of maximum and minimum deflection in a uniform beam, computing time required for the growth of population by transportation engineers [1]. Likewise, determining doping density of doped silicon, to find the distance between charge and center of ring, to find the Fanning friction factor in pipes by using Karman equation, to compute the velocity of rocket, to find the friction factor in rough pipes for turbulent flow by using Colebrook equation, to find the mole fraction of water in a chemical engineering process, to determine the amount of fuel in a tank by using Redlich-Kwong equation, to determine the depth of fluid in a cylindrical tank, to determine the angular frequency in a circuit by using Kirchhoff's law, to compute the trajectories of rockets in aerospace engineering, to find the depth of fluid in a spherical tank, to compute the complex frequency in control systems analysis, to find the depth of flow in a rectangular open channel by using Manning equation, to find the gas compressibility in [2], to define the cell's energy aging model that produce organelle in [3], to find the oscillation of simple harmonic motion in [4], to

Received: December 2019; Accepted: December 2020
* Corresponding Author: Muhammad Mujtaba Shaikh <mujtaba.shaikh@faculty.muet.edu.pk>

assess the heat transfer variation in [5], to measure the material's temperature in [2], to find the drug concentration in plasma in [6], and to compute the falling parachutist's speed in [7], etc.

The direct methods fail in finding the solutions to such applied nonlinear equations. Thus, the numerical methods may be used for the solution. Classical Newton method (NM) is the most basic numerical method for solving nonlinear equations of type . The NM finds the approximate solution with quadratic convergence. Many efficient modifications to NM have been proposed by researchers to accelerate the order of convergence and to get the desired solution in less number of iterations. Rafiullah et al. [8] proposed a ninth order iterative method in 2016, Hou and Li [9] presented twelfth order method in 2010, Hu et al. [10] developed ninth order method in 2011, Zafar and Bibi [11] proposed a fourteenth order method in 2014 for solving nonlinear equations.

The higher order iterative methods definitely require extra evaluations of function and derivative per iteration. This increases the computational load and CPU time required by computer to achieve the pre-specified tolerance. Due to this purpose, the researchers have used divided differences instead of first order derivatives in the iterative methods, for example Zafar and Bibi [11] used divided difference in place of new derivative in third step which raised the convergence of method to fourteenth order, Rafiullah and Jabeen [12] used divided differences with the help of Lagrange interpolation and accelerated the convergence order to eight and sixteen, Cordero et al. [13] proposed sixth and seventh order methods by replacing new derivatives by divided difference approximations. The use of divided differences reduces the derivative evaluation per iteration and increases the efficiency index of the method.

The main objective of this research is to propose some higher order methods with enhanced efficiency indices by incorporating the divided differences in place of new derivatives. We propose two three step algorithms for raising the order of convergence by using the divided differences in the third step. The third step, when combined with any two-step nth order convergent method, raises the convergence order to n+3 and n+4 respectively. The proposed algorithms are tested by combining the third step with some existing two-step methods, like the third order method of Darrvishi and Baraati [14], fourth order method of Khattri and Agrawal [15] and fifth order method of Rafiullah et al. [12]. Five new efficient iterative methods to solve nonlinear equations are presented using the proposed algorithms. Several numerical examples are solved including some applications such as NASA's launched satellite's nonlinear equation, nonlinear spring equation in a real mechanical system, tension force appeared in catenary cable and thermodynamics temperature computed by engineers.

The paper is organized into five sections. Section 2 contains basic definitions. Section 3 contains the proposed algorithms and the analytical proof of the improvement in orders of convergence. Five particular methods are also presented in Section 3, and the efficiency indices of proposed methods and other well-known existing iterative methods are compared their-in. The numerical examples including some applications are listed in Section 4 with details of the numerical setup to verify computational properties of proposed and other methods. The results and consequent discussions of the numerical experiments are presented in Section 5. Finally, we outline main contributions in the Conclusion.

2. SOME BASIC DEFINITIONS

Definition 1. Error equation

For scalar equations, if $e_i = x_i - \alpha$ be the error at i[th] iteration, then the error equation is defined as,

$$e_{i+1} = C\left(e_i\right)^p + O\left(e_i\right)^{p+1} \tag{1}$$

Where, p is the convergence order and C is the asymptotic error coefficient [16].

Definition 2. Order of convergence

Let $\{x_i\},\ \ i \geq 1$ be a sequence in $\mathbf{R}$ that converges to α found by an iterative method. Then, the method is said to be of converging order p, $p > 1$, as defined in [16], such that

$$\left\|e_{i+1}\right\| \leq M \left\|e_i\right\|^p \quad \forall i \geq 1 \tag{2}$$

Definition 3. Computational order of convergence

If γ_{i+1}, γ_i and γ_{i-1} be the absolute errors of three consecutive iterations, then the computational order of convergence [17] denoted by 'q' can be computed as:

$$q \approx \frac{\log\left|(\gamma_{i+1})/(\gamma_i)\right|}{\log\left|(\gamma_i)/(\gamma_{i-1})\right|} \tag{3}$$

Definition 4. Efficiency index

The efficiency index [18] is calculated by the formula:

$$E = p^{1/d} \tag{4}$$

where, d is the number of evaluations of function and derivative per iteration in an iterative method.

Conjecture 1. Kung-Traub conjecture of optimality

According to Kung and Traub conjecture [19], an iterative method is said to be optimal if its order of convergence is 2^{d-1}.

Definition 5.

Let $x_0, x_1, x_2, \ldots, x_{n-1}, x_n$ and $f(x_0), f(x_1), f(x_2), \ldots, f(x_{n-1}), f(x_n)$ be the points and their corresponding functional values respectively, then the first order divided difference [20] denoted by $f[x_0, x_1]$ is given by:

$$f[x_0, x_1] = \frac{f(x_1)-f(x_0)}{x_1-x_0} \tag{5}$$

and the n[th] order divided difference [21] represented by $f[x_0, x_1, \ldots, x_{n-1}, x_n]$ is given by

$$f[x_0, x_1, \ldots, x_{n-1}, x_n] = \frac{f[x_1, \ldots, x_{n-1}, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0} \tag{6}$$

3. PROPOSED ALGORITHMS AND METHODS

To accelerate the order of convergence, we propose two three-step algorithms by using the divided differences in place of derivatives in the following proposed scheme, equation (7).

$$\left.\begin{aligned} y_i &= g_2(x_i) \\ z_i &= h_n(x_i, y_i) \quad n \ge 3 \\ x_{i+1} &= z_i - \frac{f(z_i)}{f'(z_i)} \end{aligned}\right\} \tag{7}$$

where, $g_2(x_i)$ is any second order convergent nonlinear solver which uses information at x_i only, $h_n(x_i, y_i)$, $n \ge 3$, is an n^{th} order two-step method utilizing information at both x_i and y_i. The third step is the usual Newton step.

Following two different divided difference approximations for the replacement of $\boldsymbol{f'(z_i)}$, as suggested in [11].

$$\begin{aligned} f'(z_i) &\approx f[z_i, y_i] + f[z_i, x_i, x_i](z_i - y_i) \\ &= \frac{f(z_i) - f(y_i)}{(z_i - y_i)} \\ &\quad + \frac{[f(z_i) - f(x_i) - f'(x_i)(z_i - x_i)](z_i - y_i)}{(z_i - x_i)^2} \end{aligned} \tag{8}$$

And

$$\begin{aligned} f'(z_i) &\approx 2f[z_i, y_i] - f'(y_i) \\ &= \frac{2f(z_i) - 2f(y_i) - f'(y_i)(z_i - y_i)}{(z_i - y_i)} \end{aligned} \tag{9}$$

Using (8) and (9) in the third step of general scheme, we propose the following two algorithms:

Algorithm 1

$$\left.\begin{aligned} y_i &= g_2(x_i) \\ z_i &= h_n(x_i, y_i) \quad n \ge 3 \\ x_{i+1} &= z_i - \frac{f(z_i)(z_i-y_i)(z_i-x_i)^2}{(z_i-x_i)^2[f(z_i)-f(y_i)]+[f(z_i)-f(x_i)-f'(x_i)(z_i-x_i)](z_i-y_i)^2} \end{aligned}\right\} \tag{10}$$

Algorithm 2

$$\left.\begin{aligned} y_i &= g_2(x_i) \\ z_i &= h_n(x_i, y_i) \quad n \ge 4 \\ x_{i+1} &= z_i - \frac{(z_i-y_i)f(z_i)}{2f(z_i)-2f(y_i)-f'(y_i)(z_i-y_i)} \end{aligned}\right\} \tag{11}$$

The proposed algorithms (10) and (11) do not use any new derivative evaluation, thus, the cost of derivative evaluation is saved, and the algorithms extend the convergence to orders $n + 3$ and $n + 4$, respectively.

3.1. Order of Convergence

To prove the orders of convergence theoretically of above defined algorithms, we use the Taylor's expansion in the form of following two theorems, Theorems 1 and 2.

Theorem 1. *Let $\alpha \in Q$ be the root of differentiable function $f: Q \subset R \to R$for an open interval$Q$. Then, the general iterative algorithm 1 described in (10) has $(n+3)^{rd}$ order of convergence and satisfy the following error term,*

$$e_{i+1} = -2A_3bce_i^{n+3} + O\left(e_i^{n+4}\right), n \geq 3 \quad (12)$$

Where, $A_i = \frac{f^i(\alpha)}{i!f^{'}(\alpha)}, \quad i = 2,3,4, \ldots$

Proof of Theorem 1.

Let α be the root of $f(x)$ and$\boldsymbol{e_i} = \boldsymbol{x_i} - \boldsymbol{\alpha}$ Also let y_i be the approximate solution using any second order convergent method, $g_2(x_i)$, then:

$$\hat{e}_i = y_i - \alpha = be_i^2 + O\left(e_i^3\right) \quad (13)$$

Where b is asymptotic error coefficient of first step in (10) and it depends on A_i's.

Letz_i be the approximation to the solution by an n^{th} order convergent method, $h_n(x_i, y_i)$, $n \geq 3$ then:

$$\tilde{e}_i = z_i - \alpha = ce_i^n + O\left(e_i^{n+1}\right) \quad n \geq 3 \quad (14)$$

Where c is asymptotic error coefficient of approximation by second step, and it also depends on some A_i's.

Using Taylor's expansion for$f(z_i)$

$$f(z_i) = f^{'}(\alpha)\left[\tilde{e}_i + O\left(\tilde{e}_i^2\right)\right] \quad (15)$$

Equation (15) leads to:

$$f(z_i)(z_i - y_i)(z_i - x_i)^2 = -e_i^2\hat{e}_if^{'}(\alpha)\left[\tilde{e}_i - \frac{\tilde{e}_i^2}{\hat{e}_i} - 2\frac{\tilde{e}_i^2}{e_i}\right] \quad (16)$$

and

$$\frac{1}{\left[(z_i - x_i)^2[f(z_i) - f(y_i)] + [f(z_i) - f(x_i) - f^{'}(x_i)(z_i - x_i)](z_i - y_i)^2\right]} = -\frac{1}{e_i^2\hat{e}_if^{'}(\alpha)}\left[1 + 2A_3e_i\hat{e}_i + 2\frac{\tilde{e}_i}{e_i} + \frac{\tilde{e}_i}{\hat{e}_i}\right] \quad (17)$$

Using (16) and (17) in third step of (10), we get

$$e_{i+1} = -2A_3e_i\hat{e}_i\tilde{e}_i \quad (18)$$

Using (12) and (13) in (18), we have:

$$e_{i+1} = -2A_3bce_i^{n+3} + O\left(e_i^{n+4}\right), n \geq 3 \quad (19)$$

Equation (19) shows $(n+3)^{rd}$ order of convergence of proposed general Algorithm 1, i.e. equation (10).

Theorem 2. *Let $\alpha \in Q$ be the root of differentiable function $f: Q \subset R \to R$for an open interval$Q$. Then, the proposed iterative algorithm 2 defined in (11) has $(n+4)^{th}$ order convergence and satisfy the following error term,*

$$e_{i+1} = (A_2c^2 - A_3b^2c)e_i^{n+4} + O\left(e_i^{n+5}\right), n \geq 4 \quad (20)$$

where, $A_i = \frac{f^i(\alpha)}{i!f^{'}(\alpha)}, \quad i = 2,3,4, \ldots$

Proof of Theorem 2.

Let αbe the root of $f(x)$ and$\boldsymbol{e_i} = \boldsymbol{x_i} - \boldsymbol{\alpha}$ Also let y_i be the second order convergent method then the expression of the corresponding error term can be defined by equation (13) as before.

Letz_i be an n^{th} order convergent scheme with $n \geq 4$, then

$$\tilde{e}_i = z_i - \alpha = ce_i^n + O\left(e_i^{n+1}\right) \quad n \geq 4 \quad (21)$$

Using Taylor's expansion for$f(z_i)$

$$f(z_i) = f^{'}(\alpha)\left[\tilde{e} + \tilde{e}_i^2 + O\left(\tilde{e}_i^3\right)\right] \quad (22)$$

Also calculating

$$f(z_i)(z_i - y_i) = -\hat{e}_if^{'}(\alpha)\left[\tilde{e}_i + A_2\tilde{e}_i^2 - \frac{\tilde{e}_i^2}{\hat{e}_i}\right] \quad (23)$$

and

$$\frac{1}{2f(z_i) - 2f(y_i) - f^{'}(y_i)(z_i - y_i)} = -\frac{1}{\hat{e}_if^{'}(\alpha)}\left[1 + \frac{\tilde{e}_i}{\hat{e}_i} - 2A_2\tilde{e}_i + 3A_3\hat{e}_i^2 + \frac{\tilde{e}_i^2}{\hat{e}_i^2}\right] \quad (24)$$

Using (23) and (24) in third step of (11), we get

$$e_{i+1} = A_2\tilde{e}_i^2 - A_3\hat{e}_i^2\tilde{e}_i \tag{25}$$

Or, $e_{i+1} = (A_2c^2 - A_3b^2c)e_i^{n+4} + O(e_i^{n+5}), n \geq 4$ (26)

which shows that the Algorithm 2, i.e. equation (11) has order of convergence of $n+4$.

3.2. Proposed Iterative Methods

For verification of the proposed algorithms 1 and 2, we present five new iterative methods as an application of the proposed algorithms. These new algorithms will verify the extension in order of convergence to $n+3$ and $n+4$.

3.2.1. Proposed Methods using Algorithm 1

Using the two-step modified Newton's method [14] which is third order convergent (n = 3) as first two steps in equation (10) with the third step same as proposed, a new three-step iterative method of sixth order convergence (n +3) denoted by M1 takes the following form:

Method 1 (M1)

$$\left.\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)} \\ z_i &= y_i - \frac{f(y_i)}{f'(x_i)} \\ x_{i+1} &= z_i - \frac{f(z_i)(z_i-y_i)(z_i-x_i)^2}{[(z_i-x_i)^2[f(z_i)-f(y_i)]+[f(z_i)-f(x_i)-f'(x_i)(z_i-x_i)](z_i-y_i)^2]} \end{aligned}\right\} \tag{27}$$

Replacing the first two steps of algorithm 1, i.e. equation (10) with two-step optimal fourth order convergent (n = 4) method from [15], we get a new three-step iterative method of seventh order convergence (n + 3) denoted by M2 takes the following form:

Method 2 (M2)

$$\left.\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)} \\ z_i &= y_i - \frac{f(y_i)}{2\left(\frac{f(y_i)-f(x_i)}{y_i-x_i}\right)-f'(x_i)} \\ x_{i+1} &= z_i - \frac{f(z_i)(z_i-y_i)(z_i-x_i)^2}{(z_i-x_i)^2[f(z_i)-f(y_i)]+[f(z_i)-f(x_i)-f'(x_i)(z_i-x_i)](z_i-y_i)^2} \end{aligned}\right\} \tag{28}$$

In algorithms 1, using the fifth order convergent (n=5) method from [12] as first two steps, the new three-step iterative method of eighth order convergence (n + 3) denoted by M3 takes the following form

Method 3 (M3)

$$\left.\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)} \\ z_i &= y_i - \frac{f(y_i)}{f'(y_i)} - \frac{f(y_i)^2[f'(x_i)-f'(y_i)]}{2f'(x_i)^2[f(x_i)-f(y_i)]} \\ x_{i+1} &= z_i - \frac{f(z_i)(z_i-y_i)(z_i-x_i)^2}{(z_i-x_i)^2[f(z_i)-f(y_i)]+[f(z_i)-f(x_i)-f'(x_i)(z_i-x_i)](z_i-y_i)^2} \end{aligned}\right\} \tag{29}$$

3.2.2. Proposed Methods using Algorithm 2

Using a two-step optimal fourth order convergent (n=4) method from [15] as first two steps in equation (11), the new three-step iterative method of eighth order convergence (n + 4) using algorithm 2, denoted by M4, takes the following form

Method 4 (M4)

$$\left.\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)} \\ z_i &= y_i - \frac{f(y_i)}{2\left(\frac{f(y_i)-f(x_i)}{y_i-x_i}\right)-f'(x_i)} \\ x_{i+1} &= z_i - \frac{f(z_i)(z_i-y_i)}{2f(z_i)-2f(y_i)-f'(y_i)(z_i-y_i)} \end{aligned}\right\} \tag{30}$$

Using the two-step fifth order convergent (n = 5) method from [12] with the third step of equation (11), the new three-step iterative method of ninth order convergence (n + 4) using the algorithm 2, denoted by M5, takes the following form

Method 5 (M5)

$$\left.\begin{aligned} y_i &= x_i - \frac{f(x_i)}{f'(x_i)} \\ z_i &= y_i - \frac{f(y_i)}{f'(y_i)} - \frac{f(y_i)^2[f'(x_i)-f'(y_i)]}{2f'(x_i)^2[f(x_i)-f(y_i)]} \\ x_{i+1} &= z_i - \frac{f(z_i)(z_i-y_i)(z_i-x_i)^2}{2f(z_i)-2f(y_i)-f'(y_i)(z_i-y_i)} \end{aligned}\right\} \tag{31}$$

Applying theorems 1 and 2, the asymptotic error terms of the proposed methods: M1 (sixth order), M2 (seventh order), M3 (eighth order), M4 (eighth order) and M5 (ninth order), respectively, can be expressed in the following form:

$$M1: e_{i+1} = -4A_2^3 A_3 e_i^6 + O\left(e_i^7\right) \quad (32)$$

$$M2: e_{i+1} = 2(A_2^2 A_3^2 - A_2^4 A_3) e_i^7 + O\left(e_i^8\right) \quad (33)$$

$$M3: e_{i+1} = -\left(10A_2^5 A_3 - 3A_2^3 A_3^2\right) e_i^8 + O\left(e_i^9\right) \quad (34)$$

$$M4: e_{i+1} = \left(A_2^7 - 3A_2^5 A_3 + 2A_2^3 A_3^2\right) e_i^8 + O\left(e_i^9\right) \quad (35)$$

$$M5: e_{i+1} = -0.5A_2^2 A_3 (10A_2^4 - 3A_2^2 A_3) e_i^9 + O\left(e_i^{10}\right) \quad (36)$$

Remark 1. Five new methods: M1-M5 have been proposed by applying proposed algorithms 1 and 2 on some existing two step methods from [12] and [14-15]. The maximum order of convergence for a two-step method in literature, according to our survey, has been fifth.

Remark 2. While the algorithms 1 (2), is defined for extending any n^{th} order method, with $n \geq 3$ ($n \geq 4$), to order $n + 3$ ($n + 4$), the application of the algorithms are discussed only for $3 \leq n \leq 5$. However, the expected extension in the order of convergence using algorithms 1 and 2 has been established generally for any $n \geq 3$ and $n \geq 4$, respectively, to the orders $n + 3$ and $n + 4$ in theorems 1 and 2.

3.3. Efficiency Indices of the Proposed Methods

Here, we compare the efficiency indices of the proposed method M1-M5 with some existing methods in literature of the same order of convergence. The method M1 and M2 are compared with the sixth and seventh order methods of Alicia [13], denoted here as MK6 and MK7 respectively. The M3 and M4 methods are compared with the eighth order convergent method in [22], denoted by MZM here. The M5 method is compared with the ninth order iterative method of Noor [23], denoted by NRM. The efficiency indices of proposed methods (EM1, EM2, EM3, EM4, EM5) and discussed methods (EMK6, EMK7, EMZM , ENRM) are given in Table 1.

Table 1. Efficiency indices of proposed and discussed methods.

Order of method	Method	Efficiency index
Sixth order	M1	1.565084580073287
methods	MK6	1.565084580073287
Seventh order	M2	1.626576561697786
methods	MK7	1.626576561697786
Eighth order	M3	1.515716566510398
methods	M4	1.515716566510398
	MZM	1.515716566510398
Ninth order	M5	1.551845573915360
methods	NRM	1.551845573915360

It can be seen that the new methods M1-M5 proposed using the algorithms 1 and 2 do not compromise on the efficiency in comparison to some of the similar order methods in literature. The computational properties of the proposed methods are verified in the next section.

4. NUMERICAL EXPERIMENTS

The performance of the proposed method are highlighted and compared with other methods from literature in this section on some test problems. Following is the list of some numerical examples to be considered for comparison, most of which are taken from [11] and [24-25].

Example 1
$f_1(x) = (x-1)^3 - 1$
Example 2
$f_2(x) = \cos x - x$
Example 3
$f_3(x) = 4\sin x - x + 1$
Example 4
$f_4(x) = x + \sqrt{x^2 + 1.54e^{20}} - 2.47e^{10}$
Example 5
$f_5(x) = x^5 + x - 10000$

Example 6

$f_6(x) = cos^2 x - {}^{x}/_{5}$

Example 7

$f_7(x) = \sqrt{x} - {}^{1}/_{x} - 3$

Example 8

$f_8 = x^3 - 20$

Example 9

$f_9 = x^3 + 4x^2 - 10$

Example 10

$f_{10} = sin^2 x - x^2 + 1$

Example 11

Another nonlinear scalar equation, taken from [26], in which we compute the distance 'r' of satellite "Wind" launched by NASA from earth is given as:

$$f_{11}(r) = G\frac{M_S m}{r^2} - G\frac{M_e m}{(R-r)^2} - mr\omega^2 = 0$$

where, $G = 6.67 \times 10^{-11}$, $M_S = 1.98 \times 10^{30}[kg], M_e = 5.98 \times 10^{24}[kg]$, m=the mass of satellite $[kg]$, $R = 1.49 \times 10^{11}[m], \omega = 2\pi/T, T = 3.15576 \times 10^7[sec]$

Example 12

A nonlinear equation from [1] to find the distance 'd' above a nonlinear spring in a real mechanical system. The equation is

$$f_{12}(d) = \frac{2k_2 d^{5/2}}{5} + \frac{1}{2}k_1 d^2 - mgd - mgh = 0$$

with the parameters $k_1 = 50{,}000g/s^2, k_2 = 40g/(s^2m^5), m = 90g, g = 9.81m/s^2, h = 0.45m.$

Example 13

The tension force T_A at one end of a catenary cable [1] can be described by a nonlinear relationship as:

$$f_{13}(T_A) = \frac{T_A}{w}cosh\left(\frac{w}{T_A}x\right) + y_0 - \frac{T_A}{w} - y$$

with the parameters $w = 12, y_0 = 6, y = 15, x = 50.$

Example 14

The temperature [1] corresponding to specific heat 1.1 kJ/(kg K) used in thermodynamics by engineers. The nonlinear equation is

$$f_{14}(T) = 0.99403 + 1.671 \times 10^{-4}T + 9.7215 \times 10^{-8}T^2 - 9.5838 \times 10^{-11}T^3 + 1.9520 \times 10^{-14}T^4 - c_p$$

where, c_p is the specific heat of dry air.

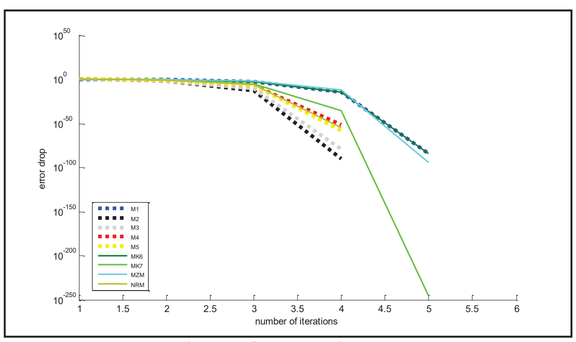

Fig. 1. Error drop of Example 1

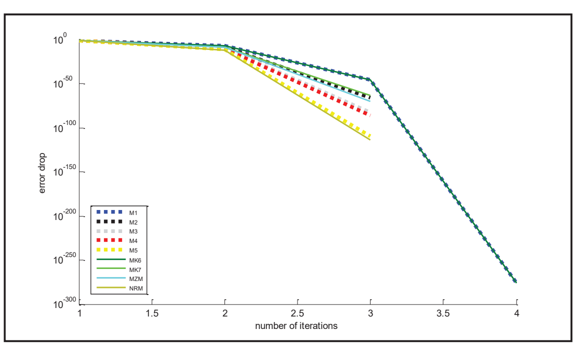

Fig. 2. Error drop of Example 2

5. RESULTS AND DISCUSSION

The computational properties of proposed methods are compared with other discussed methods from literature (Table 2) for Examples 1-10 and Examples 11-14 in Table 3. The used initial guesses are also shown in Tables 2 and 3. The results in Tables 2 and 3 are compared on the basis of number of iteration 'I', observed computational order of convergence, and absolute error γat last iteration to achieve a pre-specified error tolerance. The absolute error, γ, is given by the following formula γ=|x_(i+1)-x_i |.

We have used γ≤e^(-299)as stopping criterion to note the numerical results. MATLAB R2013a is used for calculations, installed in Intel(R) Core (TM) i3 hp laptop with RAM of 4GB operating at a processing speed of 2.4GHz with 12000 digits precision.

Figures 1-14 shows the absolute error distributions for all iterations incurred in approximations by all methods for examples 1-14, respectively. Due to the graphical limit of MATLAB, the errors lower than 10-299 were assumed zero by the software, and could not be marked in figures. It is clear from Table 2 that the theoretical orders of convergence of all proposed methods M1-M5, as established in equations (32)-(36), and other methods have been verified as expected.

It appears from Table 2 that the number of iterations required by the proposed methods M1-M5 are either equal or smaller than those in accompanying methods from literature. The same is also evident from Figures 1-10. Particularly, M2 method takes fewer iterations than MK7 in Example1, the M3 and M4 methods take fewer iterations than MZM method for Example 1 and Examples 7-9 (Table 2).

In most of the cases the proposed methods and the other accompanying methods with similar order of convergence and efficiency index take similar number of iterations, which shows that the proposed methods are comparable to other existing methods. However, in the same problems, some of the proposed methods take ascendency over other methods with respect to the computational cost and execution time, also called CPU time (in seconds) to achieve pre-specified error tolerance. It can be seen from Figure. 15 and 16 that the proposed method M1 and the other method MK6 require same number of evaluations to achieve the pre-specified error; which is referred here as the total computational cost (COC). However, with regards to CPU times as shown in Figure. 17 and 18, the proposed M1 method is better than MK6 for most of the examples. For example 1, the proposed M2 exhibits lower COC value and lesser CPU time than the accompanying MK7 method as shown in Figures 15 and 17. While the computational cost of the proposed M2 method is same as the MK7 method for Examples 2-10, 12 and 14 as shown in Figures 15 and 16, the former takes lesser CPU time in the same situations as seen from Figure 17 and 18. For example 11, as appears from Table 3, the MK7 method loses order of convergence from 7 (expected) to 1(observed), whereas the M2 method maintains seventh order convergence.

The COC and CPU times of the proposed M2 are much lower than those by MK7 as shown in Figures 16 -18. For example 13, as in Table 3 and in Figures 16-18, the MK7 method diverges with the same initial guess for which the proposed method works efficiently fine. The CPU time of the M2 is substantially smaller than that by MK7 for example 11 as in Figure 18. From Figures 15-16, we observe that the COC value for the proposed M3 and M4 methods, for both or atleast one, are smaller than those in MZM method for examples 1, 7-9, 12 and 14. For example 11 in Figure 16, some methods M3 and MZM diverge, whereas in the same situation the proposed M4 method is still applicable and maintains it theoretical order of convergence.

Figures 17 and 18 show that among the discussed eighth order methods in Tables 2 and 3, the CPU time for the proposed M4 method are always and much lower than the MZM and M3 methods, whereas the CPU times of the M3 and MZM methods are comparable and slightly take edge over each other in some examples. As described in Tables 2 and 3, both the discussed ninth order methods: proposed M5 method and NRM method, show comparable performance with regards to number of iterations and observed order of convergence. Particularly, for the case for example 11, the proposed M5 method takes much lower number of iterations, i.e. 18 as compared to those in NRM method, i.e. 49. The COC value of the proposed M5 and NRM

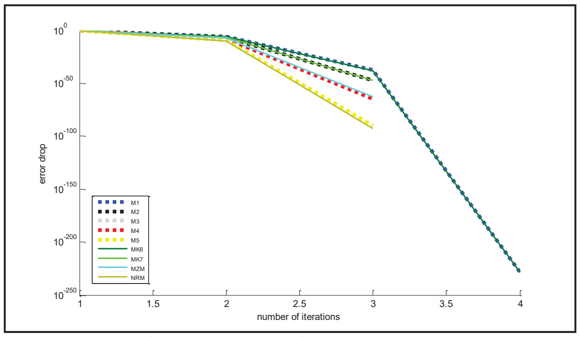

Fig. 3. Error drop of Example 3

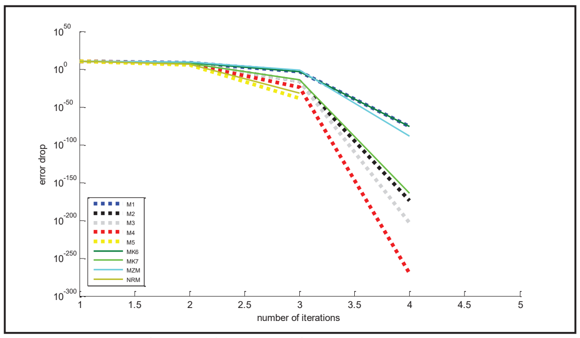

Fig. 4. Error drop of Example 4

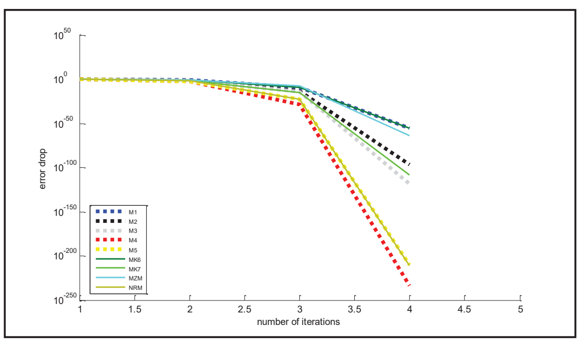

Fig. 5. Error drop of Example 5

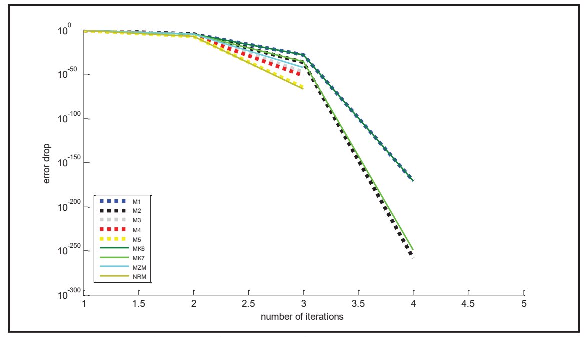

Fig. 6. Error drop of Example 6

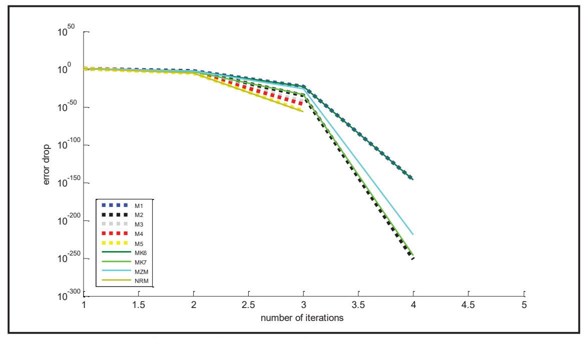

Fig. 7. Error drop of Example 7

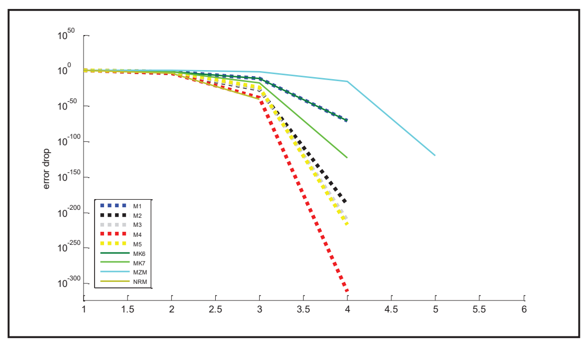

Fig. 8. Error drop of Example 8

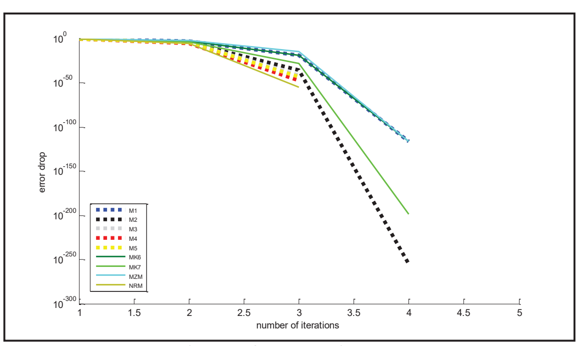

Fig. 9. Error drop of Example 9

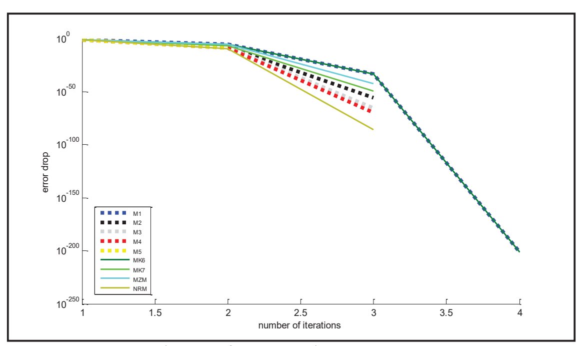

Fig. 10. Error drop of Example 10

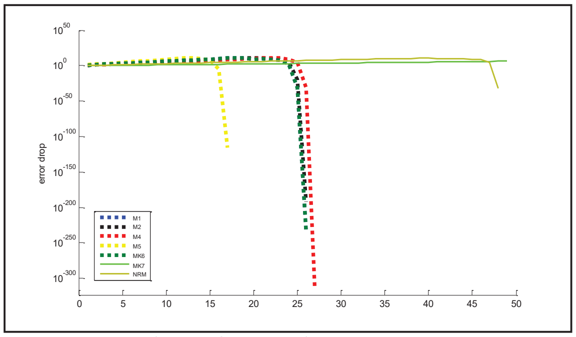

Fig.11. Error drop of Example 11

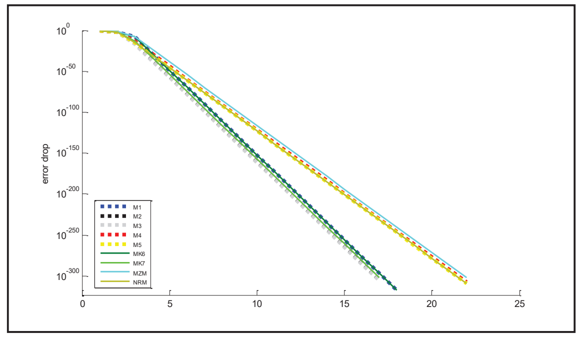

Fig. 12. Error drop of Example 12

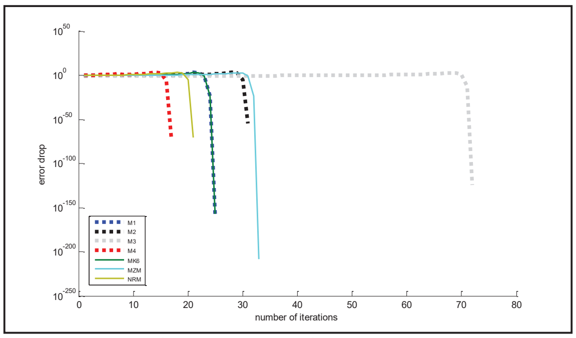

Fig. 13. Error drop of Example 13

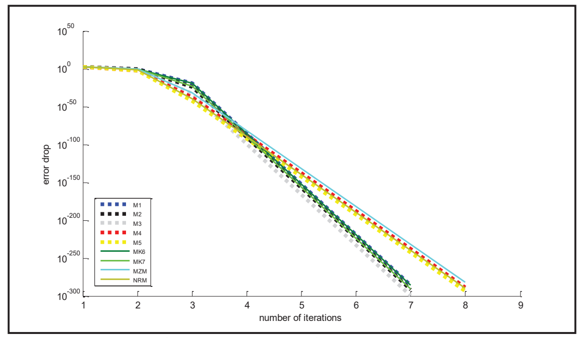

Fig. 14. Error drop of Example 14

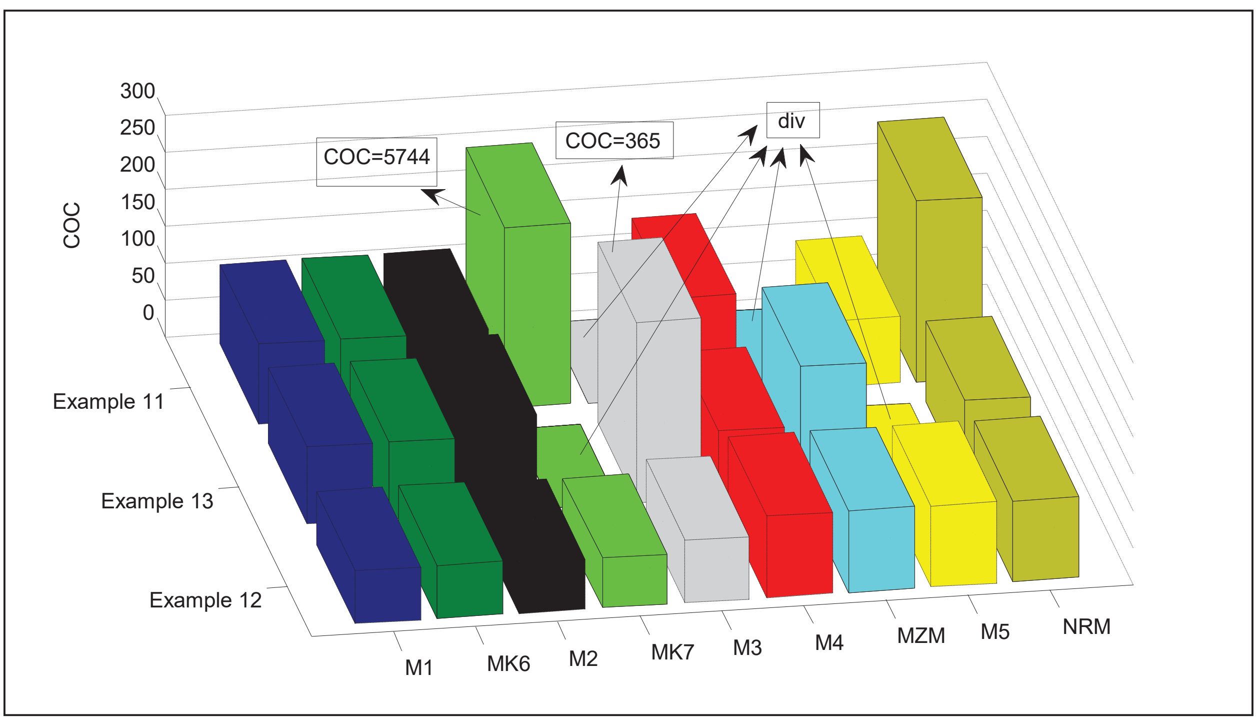

Fig. 15. COC for Examples 1-10 and Example 14

Table 2. Numerical results of Examples 1 to 10

Examples with guesses	Parameters	Sixth order methods		Seventh order methods		Eighth order methods			Ninth order methods	
		M1	MK6	M2	MK7	M3	M4	MZM	M5	NRM
f_1 $x_0 = 4.9$	I	6	6	5	6	5	5	6	5	5
	p	5.9999	5.9880	6.9999	6.9756	7.9375	7.9998	7.9368	8.9655	8.9272
	T	0.518298	0.489411	0.453423	0.614767	0.526551	0.407300	0.545045	0.478824	0.505494
	C	24	24	20	24	25	25	30	25	25
	γ	8.54e-503	8.54e-503	3.36e-633	1.24e-1716	2.0e-635	4.18e-414	2.68e-754	5.61e-520	2.43e-491
f_2 $x_0 = 0.6$	I	5	5	4	4	4	4	4	4	4
	p	6.0000	5.9927	7.0000	7.0156	8.0120	8.0000	8.0285	9.0363	8.9826
	T	0.328464	0.320908	0.285847	0.299021	0.316806	0.263534	0.300967	0.283048	0.312002
	C	20	20	16	16	20	20	20	20	20
	γ	8.63e-1660	8.63e-1660	8.90e-467	1.10e-449	6.84e-665	2.38e-693	1.80e-562	6.48e-994	1.33e-1033
f_3 $x_0 = 0$	I	5	5	4	4	4	4	4	4	4
	p	6.0001	6.0087	7.0230	6.9166	8.0307	7.9988	7.9841	8.9560	9.0430
	T	0.386319	0.383286	0.328576	0.355196	0.367492	0.286443	0.326835	0.327400	0.355268
	C	20	20	16	16	20	20	20	20	20
	γ	1.94e-1376	1.94e-1376	3.61e-334	4.41e-332	5.62e-522	1.35e-519	1.94e-503	3.90e-815	9.79e-841
f_4 $x_0 = 0$	I	5	5	5	5	5	5	5	5	5
	p	6.0000	6.6842	6.9999	7.3841	8.3578	8.0000	8.7865	9.1793	9.2154
	T	0.479607	0.487545	0.539146	0.599879	0.640161	0.479288	0.598171	0.584084	0.666031
	C	20	20	20	20	25	25	25	25	25
	γ	5.99e-508	5.99e-508	4.69e-1281	1.94e-1211	6.05e-1705	1.38e-2238	1.10e-782	3.84e-3993	3.98e-3465
f_5 $x_0 = 8.8$	I	5	5	5	5	5	5	5	5	5
	p	6.0001	5.9821	6.9997	6.9724	7.9915	7.9999	8	8.9952	9.0047
	T	0.417029	0.409521	0.464512	0.505426	0.528104	0.407066	0.483803	0.474441	0.507422
	C	20	20	20	20	25	25	25	25	25
	γ	3.31e-335	3.31e-335	7.48e-683	5.58e-760	3.95e-951	1.04e-1878	3.17e-514	1.60e-1898	3.42e-1900
f_6 $x_0 = 2.1$	I	5	5	5	5	4	4	4	4	4
	p	5.9999	6.0058	7.0000	6.992	7.9166	8.0001	8	8.9848	9.0149
	T	0.403450	0.420983	0.449957	0.502017	0.412889	0.305590	0.382907	0.367628	0.412467
	C	20	20	20	20	20	20	20	20	20
	γ	3.45e-1027	3.45e-1027	1.74e-1809	2.36e-1748	1.78e-380	4.11e-420	1.96e-344	1.65e-593	2.23e-604
f_7 $x_0 = 15.5$	I	5	5	5	5	4	4	5	4	4
	p	5.9999	6.0408	7.0000	7.0121	8.0930	8.0001	8.0319	9.1785	9.0701
	T	0.509794	0.518412	0.558535	0.636652	0.583896	0.378508	0.667903	0.515029	0.597125
	C	20	20	20	20	20	20	25	20	20
	γ	6.03e-888	6.03e-888	7.77e-1767	5.36e-1732	2.03e-348	5.91e-381	1.62e-1759	1.15e-514	2.38e-517
	I	5	5	5	5	5	4	6	5	4
	p	6.0001	5.9722	6.9999	6.9677	7.9811	7.9999	8.0083	9.0092	9.0243

Examples with guesses	Parameters	Sixth order methods		Seventh order methods		Eighth order methods			Ninth order methods	
		M1	MK6	M2	MK7	M3	M4	MZM	M5	NRM
f_8 $x_0 = 2$	T	0.369843	0.369631	0.417620	0.457535	0.513456	0.279273	0.493725	0.433418	0.347303
	C	20	20	20	20	25	20	30	25	20
	γ	4.06e-430	4.06e-430	4.46e-1324	1.58e-864	2.09e-1692	3.39e-312	4.89e-961	3.81e-1955	6.09e-370
f_9 $x_0 = 1$	I	5	5	5	5	4	4	5	4	4
	p	5.9999	5.9743	6.9999	7	8.0476	8.0000	7.9743	8.9772	9.0181
	T	0.460929	0.463776	0.519239	0.680695	0.514498	0.353051	0.507500	0.421843	0.460098
	C	20	20	20	20	20	20	25	20	20
	γ	3.19e-699	3.19e-699	7.35e-1786	5.27e-1393	3.79e-338	4.39e-385	1.13e-933	2.05e-395	7.91e-496
f_{10} $x_0 = 1.3$	I	5	5	4	4	4	4	4	4	4
	p	5.9999	5.9851	7.0002	6.96	7.9242	7.9997	7.9069	9.0125	8.9651
	T	0.444561	0.446000	0.720694	0.431389	0.452477	0.347532	0.404631	0.422791	0.439164
	C	20	20	16	16	20	20	20	20	20
	γ	3.75e-1209	3.75e-1209	2.60e-391	5.30e-348	1.06e-523	3.64e-560	2.74e-340	1.44e-721	8.30e-771

Table 3. Numerical results of Examples 11 to 14

Examples with guesses	Parameters	Sixth order methods		Seventh order methods		Eighth order methods			Ninth order methods	
		M1	MK6	M2	MK7	M3	M4	MZM	M5	NRM
f_{11} $x_0 = 2$	I	27	27	27	1436		27		18	49
	p	5.9999	5.9999	7.0000	1.0000		7.9997		9.0000	8.9997
	T	4.124009	5.186493	4.743452	631.165398		4.055858		3.366097	12.724999
	C	108	108	108	5744	**div**	135	**div**	90	245
	γ	1.65e-1449	1.65e-1449	5.13e-1401	8.52e-299		2.58e-316		3.54e-1112	3.20e-366
f_{12} $x_0 = 0.5$	I	18	18	17	17	17	22	22	22	22
	p	1.0000	1.0000	0.9999	0.9999	1.0000	1.0000	0.9998	0.9993	0.9999
	T	2.430319	2.477873	2.556034	3.106842	3.022087	2.884994	3.218211	3.426171	3.644212
	C	72	72	68	68	85	110	110	110	110
	γ	2.32e-318	2.32e-318	1.96e-305	7.88e-303	1.12e-306	1.39e-308	4.19e-303	6.98e-310	7.69e-310
f_{13} $x_0 = 10$	I	26	26	32		73	18	34		22
	p	5.9999	5.9999	6.9999		8.0001	7.9999	8.0000		9.0002
	T	3.646444	3.813766	5.043901		15.204804	2.480658	7.301787		5.660603
	C	104	104	128	**div**	365	90	170	**div**	110
	γ	3.11e-959	3.11e-959	3.31e-406		1.14e-1017	1.28e-605	9.90e-1688		3.23e-664
f_{14} $x_0 = 1$	I	8	8	8	8	7	9	9	9	9
	p	1.0000	1.0000	0.9999	0.9999	1.0000	0.9999	0.9999	0.9999	1.0000
	T	1.137198	1.166319	1.289778	1.573001	1.342675	1.237104	1.419554	1.531073	1.624203
	C	32	32	32	32	35	45	45	45	45
	γ	8.28e-354	8.28e-354	5.70e-359	4.66e-358	1.29e-300	2.95e-339	4.64e-333	4.63e-344	6.43e-341

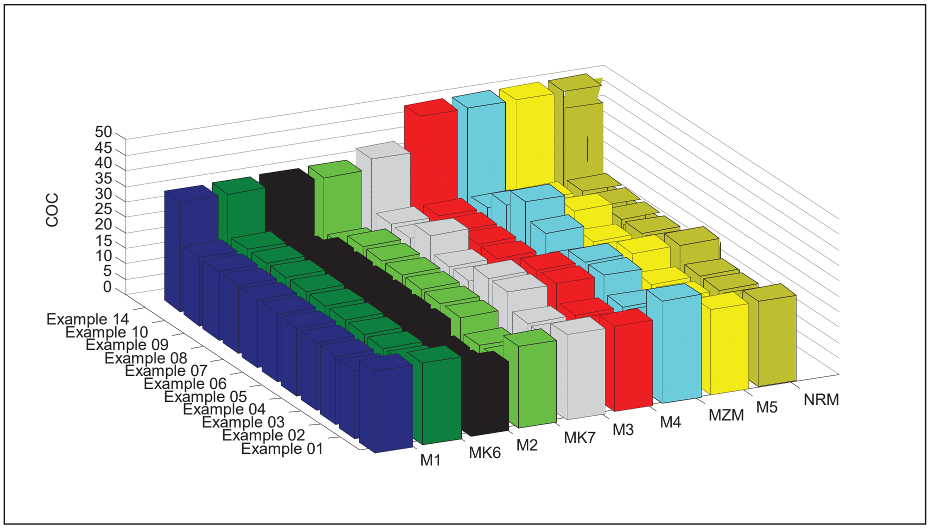

Fig. 16. COC for Examples 11-13

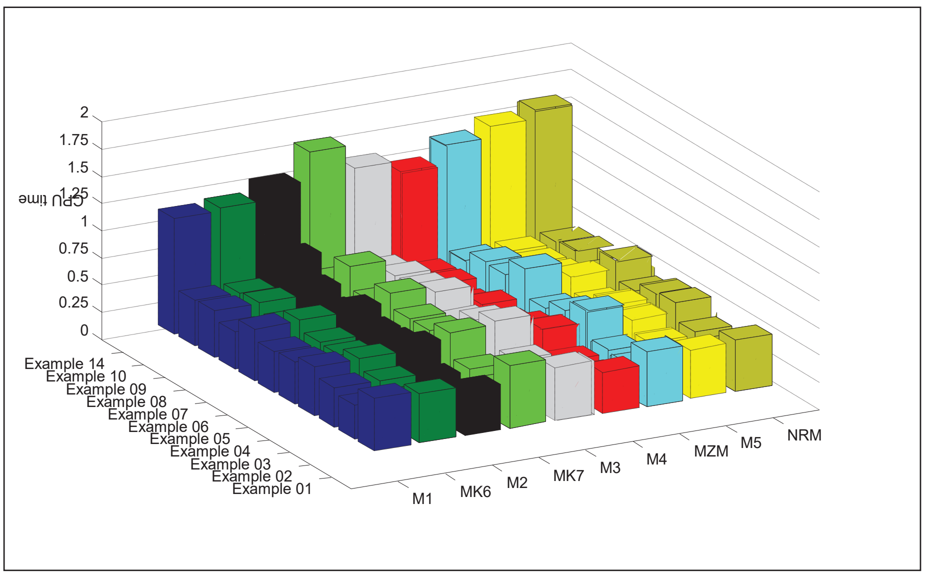

Fig. 17. CPU time for Examples 1-10 and Example 14

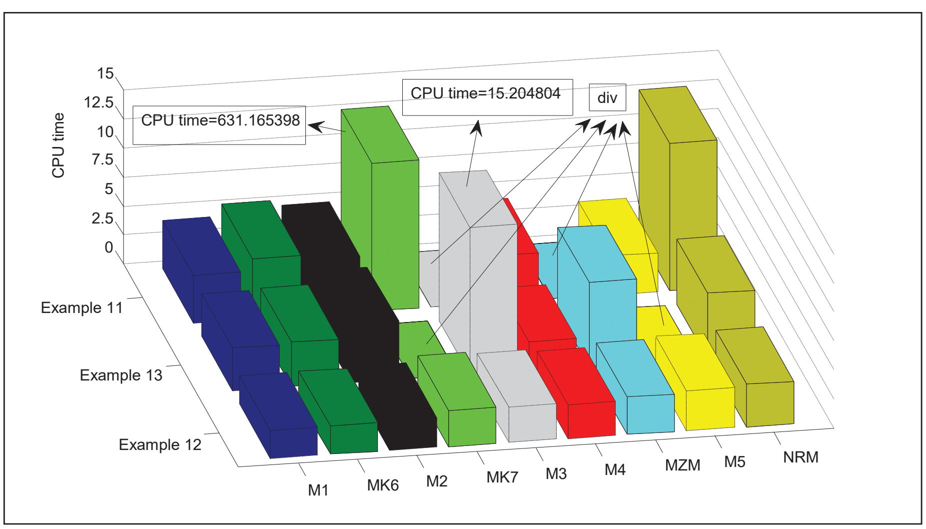

Fig. 18. CPU time for Examples 11-13

method are also comparable as seen from Figs. 15 and 16; M5 method is much better than NRM for example 11, whereas the NRM method takes slight edge over the former for example 8. The M5 method diverges in example 13, which happens mostly when methods run out of their asymptotic regimes of convergence due to a particular range of initial guesses. The CPU times of the proposed M5 method are always lower than those in NRM method for all examples where these methods are applicable, as shown in Figures 17 and 18, except for examples 8 and 13 due to same reasons.

It should be noted that there exist some highly nonlinear equations like our examples 12 and 14, in which most of the conventional methods have to compromise on the order of convergence and instead of the expected higher order methods usually end up showing almost linear convergence. From Table 3 for the case of examples 12 and 14, it is clear that all the discussed methods compromise upon the order of convergence, and converge almost linearly instead of the expected higher order convergence. This is evident in all proposed methods M1-M5 and other discussed methods, MK6, MK7, MZM and NRM methods. However, in these situations, for examples 12 and 14, Figures. 15-18 clearly depict the preference of the proposed methods over other discussed methods from viewpoints of CPU times and somewhere with respect to COC value.

The exhaustive comparison of the computational performance of the proposed M1-M5 methods with similar order methods from literature, MK6, MK7, MZM and NRM show the utility and efficiency of proposed methods. The proposed methods using algorithms 1 and 2 tend to verify the theoretical order of convergence where ever possible, like other methods and from viewpoints of COC and CPU times show comparatively better results than other methods. Among all proposed methods and discussed methods, the M2 method has highest efficiency index for scalar nonlinear equations, which is slightly less than any other optimal eighth order iterative method, i.e. 1.6818, but in most of the examples, M2 gives better solutions than other higher order convergent methods in terms of number of iterations, error drop and CPU times.

6. CONCLUSION

Two three-step algorithms have been proposed in this work to extend order of convergence of iterative methods for scalar nonlinear equations. The suggested algorithms can be applied on any two-step nth order convergent iterative method to increase the convergence orders to (n + 3) and (n + 4) by using the divided difference approximation in the third step of proposed algorithms. Theorems regarding the extension in order of convergence have been proved theoretically. Five new methods have been derived as tested as application of the proposed algorithms. The efficiency indices of the proposed methods M1-M5 are found to be higher and equivalent to some methods in literature. The computational performance of the new methods has been tested for various nonlinear equations from literature including some highly nonlinear case study equations. The numerical results show that the proposed methods perform good in terms of error drop, total computational cost, number of iterations and CPU time as compared to other methods having same efficiency index and same order of convergence.

7. ACKNOWLEDGEMENT

Author would like to thank reviewers for constructive feedback to improve the contents in this paper. Further, the facilities provided by the Mehran University of Engineering and Technology in conducting this research work are highly acknowledged.

8. REFERENCES

1. S. C. Chapra. Applied numerical methods. *Columbus:* McGraw-Hill, (2012).
2. C. F. Gerald. Applied numerical analysis. *Pearson Education India,* (2004).
3. S. U. Chaudhary, S. Y. Shin, J. K. Won, and K. H. Cho. Multiscale modeling of tumorigenesis induced by mitochondrial incapacitation in cell death. *IEEE Transactions on Biomedical Engineering*, 58(10): 3028-3032 (2011).
4. T. Baumeister and A. M. Sadegh. Marks' standard handbook for mechanical engineers (Vol. 1, p. 978). *New York: McGraw-Hill* (1978).
5. J. P. Holman. Heat transfer, 10th edition. *Mc-GrawHill Higher education* (2010).
6. S. Dhillon and A. Kostrzewski. Clinical pharmacokinetics. *Pharmaceutical Press* (2006).
7. J. Kiusalaas. Numerical Methods in Engineering with MATLAB, *Cambridge* (2005).
8. M. Rafiullah, D. K. R. Babajee and D. Jabeen Ninth order method for nonlinear equations and its dynamic behaviour, *Apulensis, Acta Universitatis,*

45: 73–86 (2016).

9. L. Hou and X. Li. Twelfth-order method for nonlinear equations, *IJRRAS*, 3: 30-36 (2010).
10. Z. Hu, G. Liu and T. Li. An iterative method with ninth-order convergence for solving nonlinear equations, *International Journal of Contemporary Mathematical Sciences* 6.1: 17-23 (2011).
11. F. Zafar and G. Bibi. A Family of Fourteenth-Order Convergent Iterative Methods for Solving Nonlinear Equations, *Chinese Journal of Mathematics* (2014).
12. M. Rafiullah and D. Jabeen. New Eighth and Sixteenth Order Iterative Methods to Solve Nonlinear Equations, *International Journal of Applied and Computational Mathematics,* 3.3: 2467-2476 (2017).
13. A. Cordero, J. L. Hueso, E. Martínez and J. R. Torregrosa. A family of iterative methods with sixth and seventh order convergence for nonlinear equations, *Mathematical and Computer Modelling* 52.9-10: 1490-1496 (2010).
14. M. T. Darvishi and A. Barati. A third-order Newton-type method to solve systems of nonlinear equations, *Applied Mathematics and Computation,* 187(2): 630-635 (2007).
15. S. K. Khattri, and R. P. Agarwal. Quadrature based optimal iterative methods, *arXiv preprint arXiv*:1004.2930 (2010).
16. J. R. Sharma, R. Sharma and N. Kalra. A novel family of composite Newton–Traub methods for solving systems of nonlinear equations, *Applied Mathematics and Computation* 269: 520-535 (2015).
17. S. Weerakoon and T. G. I. Fernando. A variant of Newton's method with accelerated third-order convergence, *Applied Mathematics Letters,* 13(8): 87-93 (2000).
18. M. Ostrowski. Solution of equations and systems of equations, *Acad. Press, Neyy York* (1966).
19. H. T. Kung and J. F. Traub. Optimal order of one-point and multipoint iteration, *Journal of the ACM (JACM),* 21(4): 643-651 (1974).
20. M. Rafiullah . Multi-step Higher Order Iterative Methods for Solving Nonlinear Equations, *MS-Thesis, Higher Education Commission of Pakistan, Spring* (2013).
21. M. K. Jain. Numerical methods for scientific and engineering computation, *New Age International,* (2003).
22. M. Z. Ullah, S. Serra-Capizzano, F. Ahmad and E. S. Al-Aidarous. Higher order multi-step iterative method for computing the numerical solution of systems of nonlinear equations: Application to nonlinear PDEs and ODEs. *Applied Mathematics and Computation,* 269: 972-987 (2015).
23. M. A. Noor, W. A. Khan, K. I. Noor and E. Al-Said. Higher-order iterative methods free from second derivative for solving nonlinear equations, *International Journal of Physical Sciences,* 6(8): 1887-1893 (2011).
24. A. Cordero, J. L. Hueso, E. Martínez and J. R. Torregrosa. A modified Newton-Jarratt's composition. *Numerical Algorithms*, 55(1): 87-99 (2010).
25. H. A. Abro and M. M. Shaikh. A new time-efficient and convergent nonlinear solver. *Applied Mathematics and Computation,* 355:516-536 (2019).
26. W. Y. Yang, W. Cao, T. S. Chung and J. Morris. Applied numerical methods using MATLAB, *John Wiley & Sons* (2005).