A modified three-step iterative method for solving nonlinear mathematical models

Shakir Ali [1*], **Muhammad Anwar Solangi** [2], **Sania Qureshi** [3]
Department of Basic Sciences and Related Studies, Mehran University of Engineering and Technology, Jamshoro, Pakistan

* Corresponding Author: **Shakir Ali**

Article Info

ISSN (online): 2582-7138
Volume: 03
Issue: 03
May-June 2022
Received: 26-04-2022;
Accepted: 11-05-2022
Page No: 321-327

Abstract
In this paper, a modified three-step iterative method is developed for solving non-linear mathematical models. In fact, finding the roots of non-linear problems containing the polynomial and transcendental equation is a classical problem in numerical methods, in which many applications arise in the branches of applied science and engineering. The proposed method is derived by using Taylor series expansion. The proposed method is having sixth-order of convergence. The main aim of this paper is to find out the approximate root of the non-linear equation with fewer iterations and good accurateness. The model is free from second order derivative and requires six function evaluations per iteration. The proposed method is compared with Newton Raphson method and another existing methods and found to have better performance.

Keywords: Non-linear models; Efficiency index; Order of convergence; NR algorithm

1. Introduction

Applied mathematics, engineering, and natural science are filled by techniques whose performance is not reasonable in same situation**.** According to current situation, the demand is higher order of convergence with fewer function evaluations and high efficiency index. In current situation, many authors try to fill that gap, unfortunately, to solve nonlinear equations mathematical studies are not enough, hence it can be handled by numerical techniques. In terms of models the nonlinear techniques are designed with nonlinear equations are observe not only in institutes and industries but also in the field of real-world and medical science. The fluid flow, heat transfer combustion amongst others are some of those fields [1-4].

Mostly, researchers working in the numerical world, are trying to modify a scheme with higher order of convergence, but trying to reduce function evaluations as per iteration, in this situation the efficiency index must be discuss and which is defined as $E = p^{\frac{1}{q}}$, where p is an order of convergence and q is number of function evaluation per iteration.

Due to simplicity, Newton Raphson (NR) [5] is considered best technique among the other classical methods. Some recent modified nonlinear techniques including a new time-efficient method (P6) [6], a new nonlinear ninth-order root finding method [7], and a new third-order derivative-based method [8] are discussed.

In [6-8], the authors blended two different methods with order p_1 and p_2 and developed new technique with order $p_1 p_2$. With higher order of convergence the procedure brings extra function evaluations at every step respectively. In other situation, Noor and Noor [9] developed three-step technique having third order of convergence with five function evaluations per iteration and Noor *et al*. [10] shown two techniques with third and fourth-order convergence with cost of five and eight function evaluations per iteration. Similarly, Shah *et al*. [11] proposed a new three-step method having sixth order of convergence with cost of seven evaluations of the function per iteration.

Main purpose of the work is to develop a new iterative method which gives us higher order of convergence with less function evaluations. The new hybrid three-step method is for solving scalar nonlinear equations.

Motivated from recent papers [6-8], we developed a new three-step technique by merging classical Newton Raphson method in [5] with an algorithm of different operators for computing of single root [12]. After blending, we get a new scheme of sixth-order convergence with six function evaluations, where three functions and three first order derivatives per iteration. The computational cost, efficiency index, CPU time will be compared with other schemes.

2. Materials and methods

Commonly, one variable non-linear equation can be shown as $f(x) = 0$, where $f(x)$ is polynomial and transcendental equation. In cases, the non-linear equations containing trigonometric relations, exponential and transcendental functions are given $f(x) = 0$. Usually, such type of non-linear equations cannot be solved by the direct or analytical approaches. In this case, numerical techniques are preferred to calculate the approximate solution. Then we argue some existing formulas from literature, starting with the Newton method (NR) [5].

2.1 Some existing techniques

The classical method of second order convergence, taking one evaluation of $f(x)$ and one of $f'(x)$, such as:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n = 0,1,2,3 \ldots. \quad (1)$$

Using this information at the previous approximation root x_n, the solution x_{n+1} was obtained.

Halley's method is one of the best methods for solving non-linear equations. This method converges cubically with three function evaluations. [7]

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2f'^2(x_n) - f(x_n)f''(x_n)}, n = 0,1,2,3 \ldots. \quad (2)$$

This three-step method shorten as P6 showing order of convergence six, which is found by theoretically. The efficiency index, computational cost (COC) and absolute error is compared with other existing method. [6]

$$\left.\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= y_n - \frac{f(y_n)}{f'(y_n)} \\ x_{n+1} &= y_n - \frac{f(y_n)+f(z_n)}{f'(y_n)} \end{aligned}\right\}, n = 0,1,2 \ldots \quad (3)$$

They proposed family of sixth order of convergence with five evaluations. The methods are denoted by SA1 and SA2 respectively. Both methods are three step methods which are being with Newton Raphson method. [13]

$$\left.\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= y_n - \frac{f(y_n)}{f'(y_n)} \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(y_n)-f(y_n)} \end{aligned}\right\}, n = 0,1,2 \ldots \quad (4)$$

and

$$\left.\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= y_n - \frac{f(y_n)}{f'(y_n)} \\ x_{n+1} &= z_n - \frac{f(z_n)f(y_n)}{f'(y_n)f(y_n)-2f(z_n)} \end{aligned}\right\}, n = 0,1,2 \ldots \quad (5)$$

To develop fourth order method with two steps for system of non-linear models, having two evaluations of single derivative of two variables per iteration [14].

$$\left.\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} &= y_n - \frac{5f'^2(x_n)+3f'^2(y_n)}{f'^2(x_n)+7f'^2(y_n)} \cdot \frac{f(y_n)}{f'(x_n)} \end{aligned}\right\}, n = 0,1,2 \ldots \quad (6)$$

Using four evaluations of function with distinct variables and a first order derivative $f'(x_n)$. The scheme is known (WM), with fifth order convergence. [15]

$$\left.\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x)} \\ z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \\ w_n &= z_n - \frac{f(z_n)}{f'(x_n)} \\ x_{n+1} &= w_n - \frac{f(w_n)}{f'(x_n)} \end{aligned}\right\}, n = 0,1,2 \ldots \quad (7)$$

2.2 Proposed method

After motivating from some recent studies [6-8], we developed a new scheme which is initiated by using the Newton method (NR) [5], and blended with an algorithm of different operators for computing of single root [12] having third order of convergence shown below:

$$\left.\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} &= y_n - \frac{2f(y_n)}{3f'(y_n)-f'(x_n)} \end{aligned}\right\}, \text{n} = 0,1,2\ldots \tag{8}$$

The recent impression of [6-8] is built on the new technique, in which by blending two different techniques with order p1 and p2 respectively, which suggested high order convergence i.e, p1p2. Each type of blending is not acceptable due to extra cost of function evaluations, according to current situation the demand is about high order of convergence with lower function evacuations. Motivated from the approach of [3-8], in this paper we modified another simultaneous sixth-order method blending a algorithm of different operators for computing of single root [12] with classical Newton Raphson method (NR) [5], therefore the latest method is three-step method.

$$\left.\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\ z_n &= y_n - \frac{f(y_n)}{f'(y_n)} \\ x_{n+1} &= z_n - \frac{2f(z_n)}{3f'(z_n)-f'(y_n)} \end{aligned}\right\}, \text{n} = 0,1,2\ldots \tag{9}$$

The proposed method is shortened as NP6 in discussion.

2.2.1. Order of convergence of the proposed method

Theorem 1. *Let $\beta \epsilon Q$ be the solution of differentiable function $F: Q \subset R \rightarrow R$ for an open interval Q. Formerly, the three-step modified iterative technique has sixth order convergent, and satisfy the error equation:*

$$e_{\,n+1} = (-D_2^5 + 2D_3D_2^3)e_n^6 + O(e_n^7) \tag{10}$$

$where\ e_n = x_n - \beta$

Proof. *Let β be the solution of $f(x_n), x_n$ be the nth approximation to the solution of sixth order and $e_n = x_n - \beta$ be the error time after nth iteration.*

Using Taylor series for $f(x_n)$ about β, we have

$$f(x_n) = f'(\beta)[e_n + D_2e_n^2 + D_3e_n^3] + O[e_n^4] \tag{11}$$

Using Taylor series for $\frac{1}{f'(x_n)}$ about β, we have

$$\frac{1}{f'(x_n)} = \frac{1}{f'(\beta)}[1 + x_1e_n + x_2e_n^2] + O[e_n^3] \tag{12}$$

Where $x_1 = -2D_2$ and $x_2 = -3D_3$

Multiplying (11) and (12) we obtain:

$$\frac{f(x_n)}{f'(x_n)} = [e_n + D_2e_n^2 + D_3e_n^3 + O(e_n^4)][1 + x_1e_n + x_2e_n^2 + O(e_n^3)] \tag{13}$$

Using (13) in first step of (9) gives:

$e_n^\wedge = -[(D_2 + x_1)e_n^2 + (D_2x_1 + x_2)e_n^3] + O[e_n^4]$ (14) Using Taylor series for $f(y_n)$ about β, we have

$$f(y_n) = f'(\beta)[e_n^\wedge + D_2e_n^{\wedge 2} + D_3e_n^{\wedge 3} + O(e_n^{\wedge 4})] \tag{15}$$

Using Taylor series for $\frac{1}{f'(y_n)}$ aboutβ, we have

$$\frac{1}{f'(y_n)} = \frac{1}{f'(\beta)}[1 + x_1e_n^\wedge + x_2e_n^{\wedge 2} + O(e_n^3)] \tag{16}$$

$where\ x_2 = 4D_2^2 - 3D_3$

Multiplying (15) and (16) we have:

$$\frac{f(yn)}{f'(y_n)} = [e_n^{\wedge} + D_2 e_n^{\wedge 2} + D_3 e_n^{\wedge 3} + O(e_n^{\wedge 4})][1 + x_1 e_n^{\wedge} + x_2 e_n^{\wedge 2} + O(e_n^{\wedge 3})] \quad (17)$$

Using (17) in second step of (9) gives:

$$e_n^{\sim} = -[(D_2 + x_1)e_n^{\sim 2} + (D_3 x_1 + D_2 x_1)e_n^{\sim 3} + O[e_n^{\sim 4}] \quad (18)$$

Using Taylor series for $\frac{1}{f'(z_n)}$ aboutβ, we have

$$f(z_n) = f'(\beta)[e_n^{\sim} + D_2 e_n^{\sim 2} + D_3 e_n^{\sim 3}] + O[e_n^{\sim 4}] \quad (19)$$

$$2f(z_n) = 2f'(\beta)[e_n^{\sim} + D_2 e_n^{\sim 2} + D_3 e_n^{\sim 3}] + O[e_n^{\sim 4}] \quad (20)$$

$$3f'(z_n) = f'(\beta)[3 + 6D_2 e_n^{\sim} + 9D_3 e_n^{\sim 2}] + O[e_n^{\sim 3}] \quad (21)$$

$Using$ (16), (20) and (21) in third step of (9) we have:

$$e_{n+1} = (-D_2^5 + 2D_3 D_2^3)e_n^6 + O[e_n^7] \quad (22)$$

Finally, we obtain $e_{n+1} = (-D_2^5 + 2D_3 D_2^3)e_n^6 + +O[e_n^7]$, which shows that the new method (NP6) is sixth order convergent.

Table 1: Theoretical properties of examined techniques.

M	Q	NFE	Coefficient of the asymptotic error term	Reference
NP6	6	6	$(-D_2^5 + 2D_2 D_2^3)$	Recent work
P6	6	5	$2D_2^4$	[6]
N2	2	2	D_2	[5]
SA1	6	5	$45f''(p)\left(\frac{2g'(x)+g(p)f''(p)}{g'(p)f'(p)}\right) - \frac{45g'(p)f''(p)}{g(p)f'(p)^4}$where $g(p) = e^{-xi}$	[13]
SA2	6	5	$45f''(p)\left(\frac{2g'(x)+g(p)f''(p)}{g'(p)f'(p)}\right) - \frac{45g'(p)f''(p)}{g(p)f'(p)^4}$where $g(p) = e^{1/f'(xi)}$	[13]
WM	5	5	$3b^4\gamma_1^4(3b\gamma_1 + 2)(9b^3\gamma_1^3 + 2)(81b^7\gamma_1^7 + 108b^6\gamma_1^6 + 36b^5\gamma_1^5 + 18b^4\gamma_1^4 + 12b^3\gamma_1^3 + 2)$	[15]
A4	4	4	$(-C_2^2 C_2 + 7/2 C_2^3 C_2^4)$	[14]
Ha	3	3	$C_2 C_2^3 - C_2^3$	[7]

2.2.3. Numerical results

In this work, we present few examples which compared with the other existing techniques respectively, when two techniques having same order and also same cost of function evaluation per cycle, the favor is given to method that has less CPU time.
It is shown above that SA1, SA2 and P6, have the same order and function evaluations, in this case preference is given to the P6 scheme because it takes less computer time.
Other important approaches are to compare the efficiency index with the other existing schemes.

Table 2: Efficiency index of the existing schemes.

M	Efficiency index	Order	NFE	Number of evaluations per cycle for n> 1
NP6	1.348006	6	6	3n + $3n^2$
P6	1.430969	6	5	3n + $3n^2$
N2	1.414213	2	2	n + n^2
SA1	1.430969	6	5	2n + $2n^2$
SA2	1.430969	6	5	3n + $2n^2$
WM	1.379729	5	5	3n + $3n^2$
A4	1.414213	5	4	4n + n^2
Ha	1.442249	3	3	n + $n^2 + n^3$

Formula, $= p^{\frac{1}{q}}$, recently used in [6-8] where p is the order of convergence and q is the cost of all new size and secondary evaluations per iteration used by the scheme. In table 2 n, n^2 and n^3 are mentioned, where n shows simple function n^2 shows first-order derivative, and n^3 shows second-order derivative respectively. So, the efficiency of the NP6 scheme for the nonlinear equation is $6^{\frac{1}{3n+3n^2}}$,$n \geq 1$, as it uses three function evaluations and three evaluations of first order-derivatives.

3. Mathematical analyses

A few abbreviations are used in this work that will be discussed here theoretically.

NP6	Modified technique with sixth order convergence given in (9);
P6	A new time efficient and convergent non-linear solver with sixth-order convergence given in (3);
NR	Newton Raphson scheme with second-order convergence given in (1);
A4	An efficient fourth-order Newton-type method (6);
SA1	Some second-derivative-free sixth-order convergent iterative methods with sixth order of convergence are given in (4);
SA2	Some second-derivative-free sixth-order convergent iterative methods with sixth order of convergence are given in (5);
WM	Efficient method for solving system of nonlinear equations with fifth-order of convergence given in (7);
Ha	Classical Halley's method with third-order of convergence (3);

M **Methods**

I	Number of iterations;
p	Order of convergence;
ϵ	Absolute error;
t	Time in second;
NFE	Number of function evaluations;
COC	Computational cost;

For performance and comparison of modified NP6 technique with other methods, we consider various scalar non-linear equations from [6-8]. Each calculation is achieved in MATLAB R2015a, which is installed on an Intel(R) Core (TM) i5-2430M CPU with 4GB of RAM and a processing speed of 2.4 GHz.

***Example* 1.** The existing methods are taken in current work. The exact solution (x^*) up to first 16 decimal places.

$$f_1(x) = 0.5 - \sin(x) \qquad x^* = 2.617993877991494$$
$$f_2(x) = 9x^2 + \sin(x) - 5 \quad x^* = 0.3953236229863151$$
$$f_3(x) = x^3 - 20 \qquad x^* = 6.308777129972690$$

The above test functions are taken from [6, 13].

The nonlinear equations are given in (example: 1-2), comparing the results with other existing methods i.e., P6 in [6], N2 in [5], A4 in [14], SA1 & SA2 in [13], WM in [15], Ha in [7] respectively. The computer time (t), COC, a number of function evaluations, absolute error (ϵ), root of function (x^*) and number of iterations (I) are shown in the (tables. 3-10).

The function $f_1(x)$ with two distinct initial guesses, $x_0 = 2$ & $x_0 = 4$, where NP6, P6, SA1 and SA2 take the same iterations, but NP6 converge to the root in minimum CPU time. WM method is a four-step technique that takes four number of iterations and also takes more CPU time, when the new method is a three-step technique and takes three iterations and less CPU time.

Table 3: The numerical computational for $f_1(x) = 0.5 - \sin(x)$, at initial guess $x_0 = 2$

M	I	x^*	ϵ	t	NFE	COC
A4	4	2.617993877991494	0	0.000019	4	16
Ha	5	2.617993877991494	0	0.000017	3	15
P6	3	2.617993877991494	0	0.000018	5	15
NR	6	2.617993877991494	0	0.000016	2	12
NP6	3	2.617993877991494	0	0.000016	**6**	**18**
SA1	3	2.617993877991494	4.440892098500626e-16	0.000021	5	15
SA2	3	2.617993877991494	0	0.000018	5	15
WM	4	2.617993877991494	0	0.000023	5	20

Table 4: The numerical computational for $f_1(x) = 0.5 - \sin(x)$, at initial guess $x_0 = 4$

M	*I*	x^*	ϵ	*t*	*NFE*	*COC*
A4	4	2.617993877991494	0	0.000019	4	16
Ha	5	2.617993877991494	0	0.000017	3	15
P6	4	2.617993877991494	0	0.000018	5	15
NR	7	2.617993877991494	0	0.000016	2	12
NP6	4	2.617993877991494	0	0.000016	6	18
SA1	4	2.617993877991494	4.440892098500626e-16	0.000021	5	15
SA2	4	2.617993877991494	0	0.000018	5	15
WM	4	2.617993877991494	0	0.000023	5	20

The non-linear equation $f_2(x)$ *with two different initial guesses* $x_0 = 5$ & $x_0 = 3.5$, where SA2 fails at both initial guesses. NR method and Ha method both takes eight iterations at $x_0 = 5$, when NP6 takes only four iterations also new method takes less computer time. Similarly, at $x_0 = 3.5$

Ha method take seven and NR method takes eight iterations and more CPU time than NP6 method.

Table 5: The numerical computational for $f_2(x) = 9x^2 + \sin(x) - 5$ initial guess $x_0 = 5$

M	*I*	x^*	ϵ	*t*	*NFE*	*COC*
A4	5	0.6959325171438922	0	0.000019	4	20
Ha	8	0.6959325171438922	6.661338147750939e-16	0.000017	3	24
P6	4	0.6959325171438922	0	0.000018	5	20
NR	8	0.6959325171438922	0	0.000018	2	16
NP6	4	0.6959325171438922	0	0.000016	6	24
SA1	5	0.6959325171438922	0	0.000021	5	25
SA2	--	Fail	--	--	5	--
WM	5	0.6959325171438922	0	0.000023	5	25

Table 6: The numerical computational for $f_2(x) = 9x^2 + \sin(x) - 5$ initial guess $x_0 = 3.5$

M	I	x^*	ϵ	t	NFE	COC
A4	4	0.6959325171438922	0	0.000017	4	16
Ha	7	0.6959325171438922	0	0.000015	3	21
P6	4	0.6959325171438922	0	0.000018	5	20
NR	8	0.6959325171438922	0	0.000016	2	16
NP6	4	0.6959325171438922	0	0.000016	6	24
SA1	4	0.6959325171438922	0	0.000021	5	20
SA2	--	Fail	--	--	5	--
WM	5	0.6959325171438922	0	0.000023	5	25

The nonlinear function, $f_3(x)$ *with two distinct initial guess* $x_0 = 6$ & $x_0 = 5.2$, at $x_0 = 6$ the A4 method, P6 method, NP6 method and SA2 method takes same number of iterations but NP6 is better in CPU time. Meanwhile, the SA1 has same order as NP6 but it takes seven iterations when NP6 takes only four iterations. Likewise, at $x_0 = 5.2$ except NR and Ha method all of the other takes four iterations but the modified method taking less CPU time.

Table 7: The numerical computational for $f_3(x) = x^3 - 20$, at initial guess $x_0 = 6$

M	I	x^*	ϵ	t	NFE	COC
A4	4	2.714417616594907	0	0.000019	4	16
Ha	5	2.714417616594907	0	0.000017	3	15
P6	4	2.714417616594907	0	0.000018	5	20
NR	8	2.714417616594907	4.440892098500626e-16	0.000016	2	16
NP6	4	2.714417616594907	0	0.000016	6	24
SA1	7	2.714417616594907	4.440892098500626e-16	0.000021	5	35
SA2	4	2.714417616594907	0	0.000018	5	20
WM	5	2.714417616594907	0	0.000023	5	25

Table 8: The numerical computational for $f_3(x) = x^3 - 20$, at initial guess $x_0 = 5.2$

M	I	x^*	ϵ	t	NFE	COC
A4	4	2.714417616594907	4.440892098500626e-16	0.000019	4	16
Ha	5	2.714417616594907	0	0.000017	3	15
P6	4	2.714417616594907	0	0.000018	5	20
NR	7	2.714417616594907	0	0.000016	2	14
NP6	4	2.714417616594907	0	0.000014	6	24
SA1	4	2.714417616594907	4.440892098500626e-16	0.000021	5	20
SA2	4	2.714417616594907	0	0.000018	5	20
WM	4	2.714417616594907	0	0.000023	5	20

Example 2: A cone filled with ice-cream which makes spherical shape. Volume of the ice-cream is shown below:

$$V = \pi\left(\frac{r^2h}{3} + \frac{r^2H}{2} + \frac{H^3}{6}\right) \tag{23}$$

$$f(H) = \pi\left(\frac{r^2h}{3} + \frac{r^2H}{2} + \frac{H^3}{6}\right) - V \tag{24}$$

Find H, where $V = 9.625\ in^3$, $h = 4in$, $\pi = \frac{22}{7}$, $r = 1.1in$. [16]

Table 9: The numerical computational for problem 1, at initial guess $H_0 = 0.6$.

M	I	x*	ϵ	t	NFE	COC
A4	4	1.487894248808740	0	0.000021	4	16
Ha	4	1.487894248808740	0	0.000020	3	12
P6	4	1.487894248808740	0	0.000021	5	20
NR	7	1.487894248808740	0	0.000016	2	14
NP6	4	1.487894248808740	0	0.000019	6	24
SA1	4	1.487894248808740	0	0.000021	5	20
SA2	-	Fail	--	--	5	--
WM	5	1.487894248808740	0	0.000023	5	25

For evaluating the absolute error distribution in the event of scalar nonlinear models Example 1-2 using:

$$\varepsilon_{n+1} = |x_{n+1} - x_n| \tag{25}$$

In computer program numerical results are usual as: $|\varepsilon| < 10^{-16}$. By using (26) formula from [6] we will calculate the order of convergence.

$$p_{n+1} \approx \frac{\log|(\varepsilon_{n+1})/(\varepsilon_n)|}{\log|(\varepsilon_n)/(\varepsilon_{n-1})|} \tag{26}$$

The product of number of evaluations (d) and number of iterations (I) is known as computation cost (COC) which given below:

$$COC = I \times d \tag{27}$$

Conclusion

In this paper, we have developed an iterative technique with sixth-order convergence for solving non-linear equations. It requires six function evaluations, where three function and three first-order derivatives are included. The efficiency index of this scheme is around 1.348 discussed above in table 2. It is mentioned in table 3-9 the proposed method takes less CPU time, hence it is clear that the proposed technique has better performance than the other existing methods. Some nonlinear examples and their results are discussed with number of iterations, COC (computational cost), function evaluations and computer time respectively. The real life problem is also discussed in this work which is related with field of geometry.

References

1. Kaplan E. Understanding GPS: Principles and Applications; Artech House: Boston, MA, USA, 1996.
2. Rouzbar R, Eyi S. Reacting flow analysis of a cavity-based scramjet combustor using a Jacobian-free Newton–Krylov method. Aeronaut. J. 2018; 122:1884-1915.
3. Nourgaliev R, Greene P, Weston B, Barney R, Anderson A, Khairallah S, Delplanque JP. High-order fully implicit solver for all-speed fluid dynamics. Shock Waves. 2019; 29:651-689.
4. Zhang H, Guo J, Lu J, Niu J, Li F, Xu Y. The comparison between nonlinear and linear preconditioning JFNK method for transient neutronics/thermal-hydraulics coupling problem. Ann. Nucl. Energy. 2019; 132:357-368.
5. Abbasbandy S. Improving Newton–Raphson method for nonlinear equations by modified Adomian decomposition method. Applied Mathematics and Computation. 2003; 145(23):887-893.
6. Abro HA, Shaikh MM. A new time-efficient and convergent nonlinear solver. Applied Mathematics and Computation. 2019; 355:516-536.
7. Qureshi S, Ramos H, Soomro AK. A New Nonlinear Ninth-Order Root-Finding Method with Error Analysis and Basins of Attraction. Mathematics. 2021; 9(16):1996.
8. Mastoi AA, Ali A, Shaikh MM, Shaikh AW. A new third-order derivative-based iterative method for nonlinear equations. J. Mech. Cont. & Math. Sci. 2020; 15(10):110123.
9. Noor MA, Noor KI. Three-step iterative methods for nonlinear equations. Appl. Math. Comput. 2006; 183:322-327.
10. Noor MA, Noor KI, Al-Said E, Waseem M. Some new iterative methods for nonlinear equations. Math. Probl. Eng. 2010, 2010, 198943.
11. Shah FA, Noor MA, Waseem M. Some second-derivative-free sixth-order convergent iterative methods for non-linear equations. Maejo Int. J. Sci. Technol. 2016; 10:79.
12. Qureshi UK, Solankiki N, Yasir Ansari M. Algorithm of Difference Operator for Computing a Single Root of Nonlinear Equations. Punjab University Journal of Mathematics. 2020; 51(4).
13. Zafar, Fiza, Gulshan Bibi. A family of fourteenth-order convergent iterative methods for solving nonlinear equations." Chinese Journal of Mathematics. 2014.
14. Fang L, Sun L, He G. An efficient Newton-type method with fifth-order convergence for solving nonlinear equations. Computational & Applied Mathematics. 2008; 27:269-274.
15. Waseem M, Noor MA, Shah FA, Noor KI. An efficient technique to solve nonlinear equations using multiplicative calculus. Turk J Math. 2018; 42:679-691.
16. Gilat A. MATLAB: an introduction with applications. John Wiley & Sons, 2004.