Modified Third Order Iterative Method for Solving Nonlinear Equations

Huda H. Omran
Department of Mathematics, College of Education, Ibn Al-Haitham, University of Baghdad.

Abstract
Many numerical approaches have been suggested to solve nonlinear problems. In this paper, we suggest a new two-step iterative method for solving nonlinear equations. This iterative method has cubic convergence. Several numerical examples to illustrate the efficiency of this method by Comparison with other similar methods is given.

Keywords: Newton's method; Nonlinear equations; Order of convergence; Modified Trapezoidal methods.

Introduction
In this paper, we develop an iterative method to find a simple root x^* of the nonlinear equation $f(x)=0$, where $f:D\subset R\to R$ is a scalar function on an open interval *D*. The design of iterative formulas for solving nonlinear equation $f(x)=0$ is a very interesting and important task in numerical analysis. Many iterative methods have been developed by using many different techniques including Taylor series [1,2], decomposition method [3-5], homotopy techniques [6,7] and quadrature formulas [8-18].

It is well known that the classical Newton's method is one of the best iterative methods for solving the nonlinear equation $f(x)=0$ by using the following iteration

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

where x_n is the n-th approximation of x^*. This method is quadratically convergent in the neighborhood of x^*.

Weerkoon and Fernando [8] rederived the classical Newton's method by approximate the definite integral in Newton's theorem

$$f(x)=f(x_n)+\int_{x_n}^{x} f'(t)\mathrm{d}t \qquad (2)$$

by rectangular rule. Also, they used the trapezoidal rule for approximating the integral in (2) to obtain the modified Newton type iterative scheme,

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

where $y_n=x_n-\dfrac{f(x_n)}{f'(x_n)},\quad n=0,1,\ldots$

This method is convergent of order three. Frontini and Sormani [10,11] used the midpoint rule for approximating the definite integral in (2) to obtain the third-order method:

$$x_{n+1}=x_n-\frac{f(x_n)}{f'\left(x_n-\dfrac{f(x_n)}{2f'(x_n)}\right)}\quad ,\ n=0,1,\ldots \qquad (4)$$

Hasanov and et al [9]derived the following cubically convergent iteration scheme:

$$x_{n+1}=x_n-\frac{6f(x_n)}{f'\left(x_n-\dfrac{f(x_n)}{f'(x_n)}\right)+4f'\left(x_n-\dfrac{f(x_n)}{2f'(x_n)}\right)+f'(x_n)},\ n=0,1,\ldots \qquad (5)$$

by using Simpson method. Rostam and Fuad [18] used the modified trapezoidal rule for approximating the definite integral in (2) to obtain the third and six order convergence iteration schemes:

$$x_{n+1}=x_n-\frac{12f(x_n)+f'^2(x_n)f^2(x_n)\left(f''(x_n)-f''\left(x_n-\dfrac{f(x_n)}{f'(x_n)}\right)\right)}{6f'^2(x_n)\left(f'(x_n)+f'\left(x_n-\dfrac{f(x_n)}{f'(x_n)}\right)\right)} \qquad (6\text{-a})$$

and

$$x_{n+1}=x_n-\frac{\left[12f(x_n)f'^2(x_n)+f^2(x_n)\left(f''(x_n)-f''(y_n)\right)\right]}{\left[6f'^2(x_n)\left(f'(x_n)+f'(y_n)\right)\right]} \qquad (6\text{-b})$$

where

$$y_n=x_n-\frac{2f(x_n)f'(x_n)}{\left(2f'^2(x_n)-f(x_n)f''(x_n)\right)}$$

respectively. Ahlam, Kaleel and et al considered in [19] Newton's theorem by the modified Simpson's rule to compute the integral of (2) and arrived at the following cubically convergent iterative scheme.

$$x_{n+1} = x_n - \frac{\left[60 f(x_n) f'^2(x_n) + f^2(x_n)\left(f''(x_n) - f''(y_n)\right)\right]}{\left[2 f'^2(x_n)\left(7 f'(y_n) + 16 f'(z_n) + 7 f'(x_n)\right)\right]} \quad ....(7)$$

where

$y_n = x_n - \dfrac{f(x_n)}{f'(x_n)}$ and $z_n = x_n - \dfrac{f(x_n)}{2 f'(x_n)}$.

While Homeier in [20] derived the cubically convergent iteration method

$$x_{n+1} = x_n - \frac{f(x_n)}{2}\left(\frac{1}{f'(x_n)} + \frac{1}{f'(x_n - f(x_n)/f'(x_n))}\right) \quad ....(8)$$

by considering Newton's theorem for the inverse function $x = f(y)$ instead of $y = f(x)$.

Modification of Newton's Method

In this paper, we give a new modification of Newton's method with locally cubically convergence. To do this, we approximate the integral in (2) by using the modified Trapezoidal method

$$\int_{x_n}^{x} f(t)dt \approx \frac{(x-x_n)}{2}\left[f'(x) + f'(x_n)\right] + \frac{(x-x_n)^2}{12}\left[f''(x_n) - f''(x)\right]$$

to obtain:

$$f(x) = f(x_n) + \frac{(x-x_n)}{2}\left[f'(x) + f'(x_n)\right] + \frac{(x-x_n)^2}{12}\left[f''(x_n) - f''(x)\right]$$

and to find the solution x_{n+1} of the equation $f(x) = 0$

$$f(x_n) + \frac{(x_{n+1}-x_n)}{2}\left[f'(x_{n+1}) + f'(x_n)\right] + \frac{(x_{n+1}-x_n)^2}{12}\left[f''(x_n) - f''(x_{n+1})\right] = 0$$

Therefore

$$x_{n+1} = x_n - \frac{12 f(x_n) + (x_{n+1}-x_n)^2\left[f''(x_n) - f''(x_{n+1})\right]}{6\left[f'(x_n) + f'(x_{n+1})\right]} \quad ....(9)$$

In [18] Rostam, approximate $(x_{n+1} - x_n)$ in right hand side of (9) by Newton's method and Halley method and arrived to iterative formula (6.a) and (6.b) respectively.

In this paper we replacing $(x_{n+1} - x_n)$, with $\dfrac{-2 f(x_n)}{f'(x_n) + f'(y_n)}$ from (3), and obtain an implicit formula

$$x_{n+1} = x_n - \left[2 f(x_n)\left(f'(x_n) + f'(y_n)\right) + \frac{2}{3}(f(x_n))^2 \left(f''(x_n) - f''(x_{n+1})\right)\right] \Big/ \left[\left(f'(x_n) + f'(y_n)\right)^2 \left(f'(x_n) + f'(x_{n+1})\right)\right]$$

In order to obtain explicit new iterative formula, we replaced x_{n+1} in the right hand side of above equation by Newton's iterative method,

$$x_{n+1} = x_n - \left[2 f(x_n)\left(f'(x_n) + f'(y_n)\right) + \frac{2}{3}(f(x_n))^2 \cdot \left(f''(x_n) - f''(y_n)\right)\right] \Big/ \left(f'(x_n) + f'(y_n)\right)^3 \quad ....(10)$$

where

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

For the order of converge we can prove the following.

Analysis of Convergence

For the order of converge we can prove the following theorem

Theorem:

Let $x^* \in D$ be a simple zero of sufficiently differentiable function $f : D \subseteq R \to R$ for an open interval *D*. If x_0 is sufficiently close to x^*, then the iterative method defined by (10) has third-order convergence, and it then satisfies the error equation

$$e_{n+1} = c_2^2 e_n^3 + O(e_n^4)$$

where, $e_n = x_n - x^*$

and $c_j = \dfrac{1}{j!}\dfrac{f^{(j)}(x^*)}{f'(x^*)}$, $j = 1, 2, \ldots$.

Proof: Let x^* be a simple zero of the function f. By expanding f as a Taylor expansion at x^*, we have:

$$f(x) = \sum_{j=0}^{\infty} \frac{1}{j!} f^{(j)}(x^*)(x - x^*)^{(j)} \quad ....(11)$$

Therefore,

$$f(x_n)=\sum_{j=1}^{\infty}\frac{1}{j!}f^{(j)}(x^*)(x_n-x^*)^j$$
$$=f'(x^*)\sum_{j=1}^{\infty}\frac{1}{j!}\frac{f^{(j)}(x^*)}{f'(x^*)}(x_n-x^*)^j$$
$$=f'(x^*)\sum_{j=1}^{\infty}c_je_n^j \quad \text{.................................}(12)$$

By differentiating (11) one and by substituting $x=x_n$ in the resulting equation one can have

$$f'(x_n)=\sum_{j=1}^{\infty}\frac{1}{(j-1)!}f^{(j)}(x^*)(x_n-x^*)^{j-1}$$
$$=f'(x^*)\sum_{j=1}^{\infty}\frac{1}{(j-1)!}\frac{f^{(j)}(x^*)}{f'(x^*)}(x_n-x^*)^{j-1}$$
$$=f'(x^*)\sum_{j=1}^{\infty}jc_je_n^{j-1} \quad \text{..............................}(13)$$

By continuing in this manner one can get

$$f^{(i)}(x_n)=f^{(i)}(x^*+e_n)=$$
$$f'(x^*)\sum_{j=i}^{\infty}j(j-1)\cdots(j-i+1)c_je_n^{j-i},\ \ i=1,2,\ldots$$
$$\text{.................................}(14)$$

From (12) and (13), we obtain

$$d_n=-\frac{f(x_n)}{f'(x_n)}=-e_n+c_2e_n^2+2(c_3-c_2^2)e_n^3+(3c_4-7c_2c_3$$
$$+4c_2^3)e_n^4+(4c_5-10c_4c_2-17c_4c_3+28c_4c_2^2+33c_3^2c_2-6c_3^2$$
$$+20c_3c_2^2-8c_2^4)e_n^5+(5c_6-13c_5c_2-17c_4c_3+28c_4c_2^2$$
$$+33c_3^2c_2-52c_3c_2^3+16c_2^5)e_n^6+\cdots,$$
$$\text{.................................}(15)$$

Expanding $f'(y_n)$ and $f''(y_n)$ about x_n, one can get:

$$f'(y_n)=f'(x_n+d_n)=f'(x_n)+f''(x_n)d_n$$
$$+\frac{1}{2}f^{(3)}(x_n)d_n^2+\frac{1}{6}f^{(4)}(x_n)d_n^3+\frac{1}{4!}f^{(5)}(x_n)d_n^4$$
$$+\frac{1}{5!}f^{(6)}(x_n)d_n^5+\frac{1}{6!}f^{(7)}(x_n)d_n^6+\cdots$$
$$\text{.................................}(16)$$

And

$$f''(y_n)=f''(x_n+d_n)=f''(x_n)+f^{(3)}(x_n)d_n$$
$$+\frac{1}{2}f^{(4)}(x_n)d_n^2+\frac{1}{3!}f^{(5)}(x_n)d_n^3+\frac{1}{4!}f^{(6)}(x_n)d_n^4$$
$$+\frac{1}{5!}f^{(7)}(x_n)d_n^5+\frac{1}{6!}f^{(8)}(x_n)d_n^6+\cdots$$
$$\text{.................................}(17)$$

By substituting (14) and (15) into (16) and (17), and after simple computation one can obtain:

$$f'(y_n)=f'(x^*)[1+2c_2^2e_n^2+4(c_3c_2-c_2^3)e_n^3+(6c_4c_2-$$
$$11c_3c_2^2+8c_2^4)e_n^4+(8c_5c_2-20c_4c_2^2+28c_3c_2^3-16c_2^5)e_n^5$$
$$+(10c_6c_2-26c_5c_2^2-16c_4c_3c_2+60c_4c_2^3+12c_3^3-68c_3c_2^4$$
$$+32c_2^6)e_n^6+\cdots,$$
$$\text{.................................}(18)$$

and

$$f''(y_n)=f'(x^*)[2c_2+6c_3c_2e_n^2+12(c_3^2-c_3c_2^2)e_n^3+(18c_4c_3$$
$$+12c_4c_2^2-42c_3^2c_2+24c_3c_2^3)e_n^4+(24c_5c_3-12c_4c_3c_2-48c_4c_2^3$$
$$-36c_3^3+120c_3^2c_2^2-48c_3c_2^4)e_n^5+(30c_6c_3+20c_5c_2^3-$$
$$78c_5c_3c_2+72c_4^2c_2-54c_4c_3^2-96c_4c_3c_2^2+144c_4c_2^4+$$
$$198c_3^3c_2-312c_3^2c_2^3+96c_3c_2^5)e_n^6+\cdots,$$
$$\text{.................................}(19)$$

From (12), (13), (14), (18) and (19), and rearranging the resultant equations we can have

$$2f(x_n)(f'(x_n)+f'(y_n))^2+\frac{2}{3}f^2(x_n)(f''(x_n)-f''(y_n))$$
$$=f'^3(x^*)[8e_n+24c_2e_n^2+(36c_3+40c_2^2)e_n^3+(48c_4+100c_3c_2$$
$$+8c_2^3)e_n^4+(\frac{184}{3}c_5+144c_4c_2+42c_3^2+52c_3c_2^2+24c_2^4)e_n^5$$
$$+(76c_6+\frac{260}{3}c_5c_2+116c_4c_3+24c_4c_2^2+134c_3^2c_2+$$
$$68c_3c_2^3-56c_2^5)e_n^6+\cdots$$
$$\text{.................................}(20)$$

From (13) and (18), and simple computation one can obtain:

$$(f'(x_n)+f'(y_n))^3=f'^3(x^*)[8+24c_2e_n+(36c_3+48c_2^2)e_n^2$$
$$+(48c_4+120c_3c_2+8c_2^3)e_n^3+(60c_5+168c_4c_2+54c_3^2$$
$$+72c_3c_2^2+48c_2^4)e_n^4+(72c_6+216c_5c_2+1440c_4c_3$$
$$+48c_4c_2^2+198c_3^2c_2+144c_2^3c_3-120c_2^5)e_n^5+$$
$$(84c_7+264c_6c_2+180c_5c_3+60c_5c_2^2+792c_4^2+$$
$$360c_4c_3c_2+360c_4c_2^3+171c_3^3-102c_3^2c_2^2+792c_3c_2^4-$$
$$1248c_3c_2^3+296c_2^6)e_n^6+\cdots$$
$$\text{.................................}(21)$$

By dividing (20) by (21), one can get:

$$s_n = e_n - c_2^2 e_n^3 + (3c_2^3 - \frac{5}{2}c_3c_2)e_n^4 + (\frac{1}{6}c_5 - 3c_4c_2 - \frac{3}{2}c_3^2 + \frac{19}{2}c_3c_2^2 - 6c_2^4)e_n^5 + (\frac{1}{2}c_6 - \frac{25}{6}c_5c_2 - \frac{7}{2}c_4c_3 + 12c_4c_2^2 + \frac{31}{4}c_2c_3^2 - \frac{43}{2}c_3c_2^3 + 9c_2^5)e_n^6 + \cdots, \qquad \text{..................................(22)}$$

where

$$s_n = \left[2f(x_n)\left(f'(x_n)+f'(y_n)\right)^2 + \frac{2}{3}f^2(x_n)\left(f''(x_n)-f''(y_n)\right)\right] \Big/ \left(f'(x_n)+f'(y_n)\right)^3$$

By substitute (22) into (10), we get the error equation

$$e_{n+1} + x^* = e_n + x^* - e_n + c_2^2 e_n^3 - (3c_2^3 - \frac{5}{2}c_3c_2)e_n^4 - (\frac{1}{6}c_5 - 3c_4c_2 - \frac{3}{2}c_3^2 + \frac{19}{2}c_3c_2^2 - 6c_2^4)e_n^5 - (\frac{1}{2}c_6 - \frac{25}{6}c_5c_2 - \frac{7}{2}c_4c_3 + 12c_4c_2^2 + \frac{31}{4}c_2c_3^2 - \frac{43}{2}c_3c_2^3 + 9c_2^5)e_n^6 + \cdots,$$

Therefore

$$e_{n+1} = c_2^2 e_n^3 + (-3c_2^3 + \frac{5}{2}c_3c_2)e_n^4 + (-\frac{1}{6}c_5 + 3c_4c_2 + \frac{3}{2}c_3^2 - \frac{19}{2}c_3c_2^2 + 6c_2^4)e_n^5 + (-\frac{1}{2}c_6 + \frac{25}{6}c_5c_2 + \frac{7}{2}c_4c_3 - 12c_4c_2^2 - \frac{31}{4}c_2c_3^2 + \frac{43}{2}c_3c_2^3 - 9c_2^5)e_n^6 + \cdots \qquad \text{..................................(23)}$$

This mean that the method defined by (10) is cubically convergent.

Numerical Examples

In this section we experiment with several numerical methods. All computations were done using MAPLE using 32 digit floating point arithmetics (Digits := 32). We use the following stopping criteria for computer programs: (i) $|x_{n-1} - x_n| < \varepsilon$, (ii) $|f(x_n)| < \varepsilon$, where $\varepsilon = 1\times 10^{-16}$.

We employ the new method given by (10), (PM) to solve some non-linear equations and compare with the Newton's method (NM), the Homeier method (HM), the Weerakoon and Fernando method (WFM), the Frontini and Sormani method (FSM),the Hasanov and et al method (HVM), the Rostam and Fuad method (RF) and Ahlam and et al method (AM). We introduce notations: x^*- the approximate zero; x_0 - initial point; D – divergent . The test functions are the same as in [8, 9, 10, 17].

In Table (1), we list the results obtained by proposed method (PM) and compared with Newton method and Weerakoon and Fernando method in some test functions. As we see from this Table, it is clear that the result obtained by our method is most effective as it converges to the root much faster from Newton method. Table (2) shows the number of iterations of each method from different initial points. We can see the present methods is also more efficient for nonlinear equations.

$f_i(x)$	x^*
$f_1(x) = e^{1-x} - 1$	1
$f_2(x) = x^3 - 10$	2.1544346900318837217592935665194
$f_3(x) = (x-1)^3 - 1$	2
$f_4(x) = \cos x - x$	0.73908513321516064165531208767387
$f_5(x) = e^{x^2+7x-30} - 1$	3
$f_6(x) = \sin x - 0.5x$	−1.8954942670339809471440357380936
$f_7(x) = \frac{1}{x} - 1$	1
$f_8(x) = x\sin(1/x) - 0.2e^{-x}$	−1.5379542430019514733278331165978
$f_9(x) = e^x - 1$	0
$f_{10}(x) = x^3 + 4x^2 - 10$	1.3652300134140968457608068289817
$f_{11}(x) = \ln(x)$	1
$f_{12}(x) = xe^{x^2} - \sin^2 x + 3\cos x + 5$	-1.2076478271309189270094167583561

Table (1)
Comparison the Resulting Obtained by Present Method in this Paper with Newton's Method and Weerakoon and Fernando Method in Some Test Functions.

$f_3(x) = (x-1)^3 - 1, \quad x_0 = 0.1, \quad x^* = 2$			
n	***NM***	***WFM***	***PM***
1	0.81152263374485596707818930041152	1.46325763865201684127711121548058	1.98456943242530297475530534001 67
2	10.2577507222238832436394846920 40	1.62632326029030790144548180093 91	1.99999620044956524333265299315 71
3	7.17572308200706433041090872372 17	1.87797480969012660884404087277 83	1.99999999999999999451470212943 043
4	5.12588855399248741028710105046 58	1.99724636287102456831156389790 21	2.00000000000000000000000000000 00
5	3.77017377024862870454344728987 04	1.99999997549638292921455932011 71	
6	2.89022000327573661584811514071 29	1.99999999999999999999998283525 31	
7	2.35344073797360897356371270526 60	2.00000000000000000000000000000 00	
8	2.08426401862741304330410011531 10		
9	2.00637896974323679392631027909 87		
10	2.00004034790260145181901581578 90		
11	2.00000000162786566942115862660 17		
12	2.00000000000000000264994663192 83		
13	2.00000000000000000000000000000 00		
$f_5(x) = e^{x^2+7x-30} - 1, \quad x_0 = 3.9, \quad x^* = 3$			
n	**NM**	**WM**	**PM**
1	3.83243268172786172008770153232 39	3.80108648072097494875045070969 72	3.78972954810862385869941834575 64
2	3.76424316947266514439968291415 57	3.70083103261961079600474805723 12	3.67778252314846555132664824031 06
3	3.69541473375975952642153256798 16	3.59918277776360355450870849927 80	3.56409039700013884383770874248 12
4	3.62593111400126004170911442499 02	3.49610414712154618333810420099 59	3.44862102922321673223468696040 22
5	3.55577871492688789560147906133 50	3.39162822870429652268303926891 84	3.33155401623266498661860247368 04
6	3.48495269756682573407250804936 12	3.28610749619485810370263980717 03	3.21416701849166511335417200354 27
7	3.41347369655761971275311578549 61	3.18126516297461552276349545215 37	3.10296388695288677110209143769 75
8	3.34143339755627950402222125269 94	3.08422527601136561524980227917 65	3.02226440570188201037451982667 24
9	3.26911757864144529489010692954 28	3.01706545013686286075987893431 49	3.00039737764921395543237499171 97
10	3.19733045506393940147731683306 15	3.00024947948328954619638210569 33	3.00000000270498781788745052646 63
11	3.12819542222686693227186719662 01	3.00000000089629440113230086951 15	3.00000000000000000000000085613 35
12	3.06677753032104219888132955724 37	3.00000000000000000000000004164 62	3.00000000000000000000000000000 00
13	3.02245422463586754574538743414 72	3.00000000000000000000000000000 00	
19	3.00000000000000000000000000000 00		
$f_{11}(x) = \ln(x), \quad x_0 = 2.7, \quad x^* = 1$			
n	**NM**	**WM**	**PM**
1	0.01822021287223484654709050857 53	2.66404809414150196839473828907 06	2.65230456884456591913297582331 27
2	0.09119624135604333193000873217 9307	2.56091246256839485036441109321 74	2.48980480948310676934277798633 93
3	0.30958767392587643772296325593 077	2.28984653872227670282819203468 11	2.02970506445089939786386533748 09
4	0.67258354064923297684767957265 237	1.73435736147927051165797894832 94	1.29607899638337942838735336975 01

5	0.93934964497641725950387094165854	1.1421768540649930056227880171602	1.0063455247028848973503503408983
6	0.99812241348385004665611217421354	1.0011513062178431153542762089176	1.0000000638101067645879392658841
7	0.99999823623021465671368928073555	1.0000000006356402761515789407031	1.0000000000000000000000649543764
8	0.999999999999844455715767375066793	1.0000000000000000000000000001070	1.0000000000000000000000000000000
9	0.999999999999999999999999879029878	1.0000000000000000000000000000000	
10	1.0000000000000000000000000000000		
$f_8(x) = x\sin(1/x) - 0.2e^{-x}$, $x_0 = 1$, $x^* = -1.5379542430019514733278331165978$			
n	**NM**	**WM**	**PM**
1	−1.0491160215191835583632832798149	−1.2502871227013327249379102513368	−1.6979563307337326814249734971666
2	−1.9728787823800764521047762729861	−1.5198914493122902935041135904788	−1.5393341473468288466497948028865
3	−1.6286417581291860334290628699112	−1.5379513807765664343918392447833	−1.5379542441122579942433515166426
4	−1.5428834681416290941016891598410	−1.5379542430019514622070310883222	−1.5379542430019514733278331171774
5	−1.5379699779596987211769425401703	−1.5379542430019514733278331165978	−1.5379542430019514733278331165978
6	−1.5379542431630679830127531766232		
7	−1.5379542430019514733447256868422		
8	−1.5379542430019514733278331165978		

Table (2)
Comparison of Various Iterative Methods and the Newton Method from Different Initial Point.

$f_i(x)$	x_0	*Number of Iterations*							
		NM	***HM***	***WFM***	***FSM***	***HVM***	***RFM***	***AM***	***PM***
$\mathbf{f_1}$	-2	8	5	6	5	5	5	5	5
	2	7	4	5	4	4	4	4	4
$\mathbf{f_2}$	1.5	6	4	4	4	4	4	4	4
	2.5	5	3	3	3	3	3	3	3
f_3	1.5	7	5	5	5	5	4	5	5
	3.6	7	5	5	5	5	5	5	5
f_4	-1	8	6	3	6	4	7	D	4
	2.5	5	4	4	4	3	3	3	3
f_5	2.9	7	5	6	5	5	4	5	5
	3.9	18	11	12	11	11	12	11	11
f_6	-2	4	3	3	3	3	3	3	3
	-1	14	6	5	6	5	D	7	7
f_7	0.1	9	6	6	6	6	6	6	6
	1.5	6	4	4	4	4	4	4	4
f_8	-5	9	5	6	6	6	6	6	6
	0.5	5	4	3	4	3	4	3	3
f_9	-1	7	4	5	4	4	4	4	4
	2	7	5	5	5	5	5	5	5
f_{10}	-1	24	9	13	9	76	34	27	10
	2.3	6	4	4	4	4	4	4	4
f_{11}	0.1	7	4	4	4	5	5	5	4
	2.7	9	4	8	4	6	D	D	7
f_{12}	-5	31	19	21	19	20	20	20	19
	4	111	D	17	D	17	D	15	12

Conclusions

In this paper, we present a new third order convergence method for solving nonlinear equation *f(x)=0*. From numerical examples, we show that the efficiency of proposed method is about the same as third order convergence methods WFM, FSM, HM, HVM, RFM and AM in most case, and greater than NM.

References

[1] Burden, R.L., Faires, J.D., "Numerical Analysis", 7th ed., PWS Publishing Company, Boston; 2001.

[2] Wait, R., "The Numerical Solution of Algebraic Equations", A Wiley-Interscience Publication, 1979.

[3] Babolian, E., Biazar, J., "Solution of Nonlinear Equations by Adomian Decomposition Method", Appl. Math. Comput., 132, 167–172, 2002.

[4] Abbasbandy, S., "Improving Newton–Raphson Method for Nonlinear Equations by Modified Adomian Decomposition Method, Appl. Math. Comput., 145, 887–893, 2003.

[5] Chun,C., "Iterative Methods Improving Newton's Method by the Decomposition Method", Comput. Math. Appl., 50, 1559–1568, 2005.

[6] Golbabai, A., Javidi, M., "New Iterative Methods for Nonlinear Equations by Modified HPM", Appl. Math. and Comput., 191, 122–127, 2007.

[7] Golbabai, A., Javidi, M., "A Third-Order Newton Type Method for Nonlinear Equations Based on Modified Homotopy Perturbation Method", Appl. Math. and Comput., 191, 199–205, 2007.

[8] Weerakoom, S., Fernando, T.G.I., "A Variant of Newton's Method with Accelerated Third-Order Convergence", Appl.Math.Lett.13, 87–93, 2000.

[9] Hasanov, V.I., Ivanov, I.G Nedzhibov, G., "A New Modification of Newton's Method", In: Application of Mathematics in Engineering and Economics, Heron Press, Sofia, 278–286, 2002.

[10] Frontini,M., Sormani, E., "Some Variants of Newton's Method with Third-Order Convergence", Appl.Math.Comput.,140, 419–426, 2003.

[11] Frontini, M., E. Sormani, "Modified Newton's Method with Third-Order Convergence and Multiple Roots", Comp. Appl. Math., 345–354, 2003.

[12] Frontini, M., "Hermite Interpolation and a New Iterative Method for the Computation of the Roots of Non-Linear Equations, Calcolo, 40, 109–119, 2003.

[13] Frontini,M., Sormani, E., "Third-Order Methods from Quadrature Formulae for Solving Systems of Nonlinear Equations, Appl. Math. Comput., 149, 771-782, 2004.

[14] Nenad, U., "A Method for Solving Nonlinear Equations", Appl. Math. and Comput., 174, 1416–1426, 2006.

[15] Cordero, A., Torregrosa, J. R., "Variants of Newton's Method Using Fifth-Order Quadrature Formulas", Appl. Math. Comput., 190, 686-698, 2007.

[16] Darvishi,M.T., A. Barati, "A Fourth-Order Method from Quadrature Formulae to Solve Systems of Nonlinear Equations", Appl. Math. Comput., 188, 257-261, 2007.

[17] Muhammad A. N., Khalida I. N., "Improved Iterative Methods for Solving Nonlinear Equations", Appl. Math. and Comput., 184, 270–275, 2007.

[18] Rostam K. S., Fuad W. K., "Three New Iterative Methods for Solving Nonlinear Equations", Aust. J. of Basic and Appl. Sci., 4(6), 1022-1030, 2010.

[19] Ahlam J. K., Huda H. O., Salam J. M., "New Third and Six Order Iterative Methods for Solving Nonlinear Equations", To appear, 2011.

[20] Homeier, H.H.H., "On Newton-Type Methods with Cubic Convergence", J. Comput. Appl. Math., 176, 425-432, 2005.

الخلاصة

العديد مِنْ الاساليب العدديةِ إقترحتْ لحَلّ المسائلِ اللاخطّيةِ. في هذه البحث، نَقترحُ طريقة خطوتِينِ تكراريةِ جديدةِ لحَلّ المعادلاتِ اللاخطّيةِ. هذه الطريقةِ التكراريةِ لَها تقاربُ مكعّبُ. عِدّة أمثلة عددية معطاة لتَصوير كفاءةِ هذه الطريقةِ بالمُقارَنَة مَع الطرقَ المماثلةَ الأخرى.