PREDICTOR CORRECTOR ITERATIVE METHOD FOR SOLVING NONLINEAR EQUATIONS

Muhammad Tanveer
Department of mathematics and statistics, University of Lahore, Lahore Pakistan
tanveer.7955180@yahoo.com
Iftikhar Ahmad and Waqas Aslam
Department of Mathematics, Lahore Leads University, Lahore, Pakistan
iftikhar.ahmad@leads.edu.pk, waqas.aslam841@gmail.com
Qaisar Mehmood
Govt. Science College Wahdat road, Lahore Pakistan
Qaisar47@hotmail.com

ABSTRACT**:** *The aim of this paper is to introduce a new predictor corrector iterative method for solving nonlinear equations. We propose a two-step predictor corrector iterative method having convergence of order nine and efficiency index* 1.7321. *We compare our proposed iterative method with Newton's Raphsan method, Halley's method, Householder method, Abbasbandy's method and etc. to check validity and efficiency by choosing some test functions.*

2010 Mathematics Subject Classification: 37F50.
Key words and phrases: *non-linear equation, iterative methods, convergence analysis, Newton's Raphsan method, Halley method, Abbasbandy's method*

INTRODUCTION

The boundary value problems (BVPs) in Kinetic theory of gases, elasticity and other applied areas are mostly reduced in solving single variable nonlinear equations. Hence, the problem of approximating a solution of the nonlinear equations $f(x)=0$, is important. The numerical methods for the roots of such equations are called iterative methods. Many such iterative methods for solving nonlinear equations are in literature for example [1-30] and the reference therein. There are two types of iterative methods, i.e. derivative free methods [27] and, higher order iterative methods involving derivatives [1-26]. Here, we are interested in finding higher order iterative method involving derivative.

In this paper, a new predictor corrector iterative method for solving nonlinear equations is presented. It is shown that this new algorithm has convergence or order nine and efficiency index 1.7321. The breakup of the paper is: In the second section, we give new iterative method predictor corrector iterative method. In the third section, we proved that convergence order of presented iterative method is at-least nine. In fourth section, we compare the efficiency index of presented iterative method with some other iterative methods. In the fifth section, some test examples are solved to check the fast convergence of presented iterative method. In the sixth section, we made some conclusions.

NEW ITERATIVE METHOD

Consider the nonlinear algebraic equation

$$f(x)=0 \qquad (2.1)$$

We assume that α is a simple zero of Eq. (2.1) and γ is an initial guess sufficiently close to α . Using the Taylor's series, we have

$$f(\gamma)+(x-\gamma)f'\gamma)+\frac{1}{2!}(x-\gamma)^2 f''(\gamma)+...=0$$

(2.2)

If $f'(\gamma)\neq 0$, we can evaluate the above expression (2.2) as follow's:

$$f(\gamma)+(x-\gamma)f'(\gamma)=0.$$

This formulation is used to suggest the following iterative method,

Algorithm 2.1 For a given x_0, compute the approximate solution x_{n+1} by the iterative scheme

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

This is well known Newton's method (NM) [9,10] for root-finding of nonlinear functions, which converges quadratically.

Also from (2.2), we obtain

$$x=\gamma-\frac{2f(\gamma)f'(\gamma)}{2f'^2(\gamma)-f(\gamma)f''(\gamma)}$$

This formulation allows us to suggest the following iterative method for solving nonlinear equation (2.1).

Algorithm 2.2 For a given x_0, compute the approximate solution x_{n+1} by the iterative scheme

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

This is known as Halley's Method, which has cubic convergence [2,12,19].

Algorithm 2.3 For a given x_0, compute the approximate solution x_{n+1} by the iterative scheme

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}-\frac{f^2(x_n)f''(x_n)}{2f'^3(x_n)}$$

This is so-called Householder method, which has convergence of order three [9].

Abbasbandy improve Newton-Raphson method by modified Adomian decomposition method, and develop following third order iterative method [1]

Algorithm 2.4 For a given x_0, compute the approximate solution x_{n+1} by the iterative scheme

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}-\frac{f^2(x_n)f''(x_n)}{2f'^3(x_n)}-\frac{f^3(x_n)f'''(x_n)}{6f'^4(x_n)}.$$

This is so-called Abbasbandy method for root-finding of nonlinear functions.

Noor and Noor [26], have suggested the following two-step method

Algorithm 2.5 For a given x_0, compute the approximate solution x_{n+1} by the iterative scheme

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

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

Traub [28] considered following two-step iterative methods of convergence order three and four respectively

Algorithm 2.6 For a given x_0, compute the approximate solution x_{n+1} by the iterative scheme

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

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

Algorithm 2.7 For a given x_0, compute the approximate solution x_{n+1} by the iterative scheme

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

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

We suggest following two-step iterative method called as predictor corrector iterative method (PCIM).

Algorithm 2.8 For a given x_0, compute the approximate solution x_{n+1} by the following iterative schemes:

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

$$x_{n+1}=y_n-\frac{f(y_n)}{f'(y_n)}-\frac{f^2(y_n)f''(y_n)}{2f'^3(y_n)}-\frac{f^3(y_n)f'''(y_n)}{6f'^4(y_n)}$$

this is our new as predictor corrector iterative method (PCIM).

CONVERGENCE ANALYSIS

In the following theorem, we will find convergence order of new as predictor corrector iterative method (PCIM).

Theorem 3.1 Let α is a root of the equation $f(x)=0$. If $f(x)$ is sufficiently smooth in the neighborhood of α , then our new predictor corrector iterative method (PCIM) has 9th order of convergence.

Proof. To prove the convergence of the predictor corrector iterative method (PCIM) is nine, suppose that α is a root of the equation $f\left(x\right)=0$ and e_n be the error at nth iteration, than $e_n=x_n-\alpha$ then by using Taylor series expansion, we have

$$f(x_n)=f'(x_n)e_n+\frac{1}{2!}f''(x_n)e_n^2+\frac{1}{3!}f'''(x_n)e_n^3+\frac{1}{4!}f^{(iv)}(x_n)e_n^4+\frac{1}{5!}f^{(v)}(x_n)e_n^5+\frac{1}{6!}f^{(vi)}(x_n)e_n^6+O(e_n^7)$$

$$f(x_n)=f'(\alpha)[e_n+c_2e_n^2+c_3e_n^3+c_4e_n^4+c_5e_n^5+c_6e_n^6+c_7e_n^7+O(e_n^8)] \qquad \textbf{(3.1)}$$

$$f'(x_n)=f'(\alpha)[1+2c_2e_n+3c_3e_n^2+4c_4e_n^3+5c_5e_n^4+6c_6e_n^5+7c_7e_n^6+O(e_n^7)] \qquad \textbf{(3.2)}$$

$$f''(x_n)=f'^2(\alpha)[2c_2+6c_3e+12c_4e_n^2+20c_5e_n^3+30c_6e_n^4+42c_7e_n^5+56c_8e_n^6+72c_9e_n^7+O(e_n^8)], \qquad \textbf{(3.3)}$$

where

$$c_n=\frac{1}{n!}\frac{f^{(n)}(\alpha)}{f'(\alpha)}$$

By using (3.1), (3.2) and (3.3) we have

$$y_n=f'(\alpha)[\alpha+(-c_3+c_2^2)e_n^3+(-3c_4+6c_2c_3-3c_2^3)e_n^4+(12c_2c_4+6c_3^2-6c_5+6c_2^4-18c_3c_2^2)e_n^5+(-10c_6+19c_4c_3$$
$$+20c_5c_2-28c_2c_3^2-29c_2^2c_4+37c_2^3c_3-9c_2^5)e_n^6+(27c_3c_5+48c_2^3c_4+66c_3^2c_2^2-42c_5c_2^2-51c_2^4c_3-15c_7-72c_2c_4c_3+$$
$$30c_2c_6+12c_4^2-12c_3^3+9c_2^6)e_n^7+(36c_6c_3+42c_2c_7-33c_2c_4^2-36c_3^2c_4+36c_3^3c_2-42c_2^4c_4+21c_2^5c_3-63c_3^2c_2^3+57c_2^3c_5$$
$$-57c_2^2c_6+27c_4c_5-21c_8+117c_2^2c_4c_3-84c_5c_2c_3)e_n^8+(56c_2c_8+28c_6c_4+46c_7c_3-28c_9+3c_2^2c_4^2-28c_5c_3^2-54c_2^5c_4$$
$$-188c_2^4c_3^2+62c_3^3c_2^2-19c_3c_4^2-74c_2^2c_7+64c_2^3c_6-18c_2^4c_5+135c_2^6c_3+12c_2c_4c_3^2-40c_5c_2c_4+86c_2^3c_3c_4-92c_2c_3c_6+$$
$$66c_3c_2^2c_5+10c_5^2-27c_2^8)e_n^9+O(e_n^{10})] \qquad \textbf{(3.4)}$$

$$f(y_n) = f^{'}(\alpha)[(-c_3 + c_2^2)e_n^3 + (-3c_4 + 6c_2c_3 - 3c_2^3)e_n^4 + (12c_2c_4 + 6c_3^2 - 6c_5 + 6c_2^4 - 18c_3c_2^2)e_n^5 + (-10c_6 + 19c_4c_3 + 20c_5c_2 - 27c_2c_3^2 - 29c_2^2c_4 + 35c_2^3c_3 - 8c_2^5)e_n^6 + (-66c_2c_4c_3 + 54c_3^2c_2^2 - 33c_2^4c_3 + 42c_2^3c_4 + 3c_2^6 + 27c_3c_5 - 42c_5c_2^2 - 15c_7 + 30c_2c_6 + 12c_4^2 - 12c_3^3)e_n^7 + (57c_2^2c_4c_3 + 24c_3^3c_2 - 72c_5c_2c_3 - 63c_2^5c_3 + 21c_3^2c_2^3 + 45c_2^3c_5 + 21c_2^7 - 24c_2c_4^2 + 36c_6c_3 + 42c_2c_7 - 36c_3^2c_4 - 57c_2^2c_6 + 27c_4c_5 - 21c_8)e_n^8 + (56c_2c_8 + 28c_6c_4 + 46c_7c_3 - 28c_9 - 69c_2^2c_4^2 - 28c_5c_3^2 \; 220c_2^5c_4 - 573c_2^4c_3^2 + 193c_3^3c_2^2 - 19c_3c_4^2 - 74c_2^2c_7 + 44c_2^3c_6 + 58c_2^4c_5 + 408c_2^6c_3 - 62c_2c_4c_3^2 - 4c_5c_2c_4 + 434c_2^3c_3c_4 - 72c_2c_3c_6 - 46c_3c_2^2c_5 + 10c_5^2 - c_3{}^4 - 81c_2^8)e_n^9 + O(e_n^{10})] \quad \textbf{(3.5)}$$

$$f^{'}(y_n) = f^{'}(\alpha)[1 + (-2c_2c_3 + 2c_2^3)e_n^3 + (-6c_2c_4 + 12c_3c_2^2 - 6c_2^4)e_n^4 + (24c_2^2c_4 + 12c_2c_3^2 - 12c_5c_2 + 12c_2^5 - 36c_2^3c_3)e_n^5 \; (-20c_2c_6 + 38c_2c_4c_3 + 40c_5c_2^2 - 62c_3^2c_2^2 - 58c_2^3c_4 + 77c_2^4c_3 - 18c_2^6 + 3c_3^3)e_n^6 + (18c_3^2c_4 - 60c_3^3c_2 + 186c_3^2c_2^3 - 162c_2^2c_4c_3 - 120c_2^5c_3 + 54c_5c_2c_3 + 96c_2^4c_4 - 84c_2^3c_5 - 30c_2c_7 + 60c_2^2c_6 + 24c_2c_4^2 + 18c_2^7)e_n^7 + (-252c_2c_4c_3^2 - 36c_3^4 + 36c_5c_3^2 - 378c_2^4c_3^2 + 324c_3^3c_2^2 + 360c_2^3c_3c_4 - 204c_3c_2^2c_5 + 105c_2^6c_3 + 27c_3c_4^2 + 72c - 2c_3c_6 + 84c_2^2c_7 - 66c_2^2c_4^2 - 84c_2^5c_4 + 114c_2^4c_5 - 114c_2^3c_6 + 54c_5c_2c_4 - 42c_2c_8)e_n^8 + (-54c_2^9 + 60c_6c_3^2 - 226c_3^3c_4 + 384c_3^4c_2 - 1022c_3^3c_2^3 + 440c_2^5c_3^2 + 108c_3c_2^7 - 56c_2c_9 + 6c_2^3c_4^2 - 104c_2^6c_4 - 148c_2^3c_7 + 128c_2^4c_6 - 36c_2^5c_5 + 20c_2c_5^2 + 56c_2c_6c_4 + 1080c_2^2c_4c_3^2 - 338c_2^4c_3c_4 + 108c_3c_4c_5 + 360c_2^3c_3c_5 - 392c_5c_2c_3^2 + 92c_2c_7c_3 - 254c_2c_3c_4^2 - 244c_3c_2^2c_6 - 80c_5c_2^2c_4 + 112c_2^2c_8)e_n^9 + O(e_n^{10})] \quad \textbf{(3.6)}$$

$$f^{''}(y_n) = f^{'^2}(\alpha)[2c_2 + (-6c_3^2 + 6c_3c_2^2)e_n^3 + (-18c_4c_3 + 36c_2c_3^2 - 18c_2^3c_3)e_n^4 + (72c_2c_4c_3 + 36c_3^3 - 36c_3c_5 + 36c_2^4c_3 - 108c_3^2c_2^2)e_n^5 + (-60c_6c_3 + 126c_3^2c_4 + 120c_5c_2c_3 - 168c_3^3c_2 - 198c_2^2c_4c_3 + 222c_3^2c_2^3 - 54c_2^5c_3 + 12c_2^4c_4)e_n^6 + (144c_3c_4^2 - 576c_2c_4c_3^2 + 504c_2^3c_3c_4 - 72c_2^2c_4^2 - 72c_2^5c_4 + 162c_5c_3^2 + 396c_3^3c_2^2 - 252c_3c_2^2c_5 - 306c_2^4c_3^2 - 90c_7c_3 + 180c_3c_2c_6 - 72c_3^4 + 54c_2^6c_3)e_n^7 + (-918c_3c_2c_4^2 - 360c_3^3c_4 + 306c_4c_3c_5 - 1260c_2^4c_3c_4 + 1710c_2^2c_4c_3^2 + 504c_2^3c_4^2 - 144c_5c_2^2c_4 + 252c_2^6c_4 + 108c_4^3 + 216c_6c_3^2 + 252c_2c_7c_3 + 216c_3^4c_2 + 126c_2^5c_3^2 - 378c_3^3c_2^3 + 342c_2^3c_3c_5 - 342c_3c_2^2c_6 - 126c_3c_8 - 504c_5c_2c_3^2)e_n^8 + (-552c_2c_3^2c_6 - 168c_2^4c_3c_5 - 1584c_4c_5c_2c_3 + 276c_7c_3^2 + 456c_3^2c_2^2c_5 + 912c_4c_2^3c_5 - 4068c_3^2c_4c_2^3 + 4194c_2^2c_4^2c_3 + 1608c_3^3c_4c_2 + 336c_3c_2c_8 + 408c_4c_6c_3 - 240c_2^2c_4c_6 + 2940c_2^5c_4c_3 + 384c_2^3c_3c_6 - 444c_3c_2^2c_7 - 1002c_3^2c_4^2 - 1992c_2^4c_4^2 - 648c_4c_2^7 - 864c_2c_4^3 + 432c_4^2c_5 - 168c_3c_9 - 188c_5c_3^3 - 1128c_2^4c_3^3 + 372c_3^4c_2^2 + 810c_2^6c_3^2 + 60c_3c_5^2 - 162c_3c_2^8 + 20c_5c_2^6)e_n^9 + O(e_n^{10})] \quad \textbf{(3.7)}$$

$$f^{'''}(y_n) = f^{'^3}(\alpha)[6c_3 + (-24c_4c_3 + 24c_2^2c_4)e_n^3 + (-72c_4^2 + 144c_2c_4c_3 - 72c_4c_2^3)e_n^4 + (288c_2c_4^2 + 144c_3^2c_4 - 144c_4c_5 + 144c_2^4c_4 - 432c_2^2c_4c_3)e_n^5 + (-240c_4c_6 + 456c_3c_4^2 + 480c_4c_5c_2 - 672c_2c_4c_3^2 - 696c_2^2c_4^2 + 888c_2^3c_3c_4 - 216c_2^5c_4 + 60c_5c_3^2 - 120c_3c_2^2c_5 + 60c_2^4c_5)e_n^6 + (1008c_4c_3c_5 - 720c_5c_2c_3^2 + 1080c_2^3c_3c_5 - 1368c_5c_2^2c_4 - 360c_2^5c_5 + 1152c_2^3c_4^2 + 1584c_2^2c_4c_3^2 - 1224c_2^4c_3c_4 - 360c_4c_7 - 1728c_3c_2c_4^2 + 720c_2c_6c_4 + 288c_4^3 - 288c_3^3c_4 + 216c_2^6c_4)e_n^7 + (-5616c_4c_5c_2c_3 - 720c_5c_3^3 + 720c_3c_5^2 - 5040c_2^4c_3c_5 + 5040c_3^2c_2^2c_5 + 3888c_4c_2^3c_5 - 720c_5^2c_2^2 + 1260c_5c_2^6 + 1188c_4^2c_5 + 864c_4c_6c_3 + 1008c_2c_4c_7 - 792c_2c_4^3 - 864c_3^2c_4^2 + 864c_3^3c_4c_2 - 1008c_2^4c_4^2 + 504c_2^5c_4c_3 - 1512c_3^2c_4c_2^3 - 1368c_2^2c_4c_6 - 504c_4c_8 + 2808c_2^2c_4^2c_3)e_n^8 + (22464c_5c_2^2c_4c_3 - 2208c_2c_4c_3c_6 + 1344c_2c_4c_8 - 6720c_5^2c_2c_3 - 1776c_2^2c_4c_7 \; 360c_6c_3^2c_2^2 - 22920c_5c_3^2c_2^3 - 5112c_5c_3^2c_4 + 3240c_2^6c_4c_3 - 360c_6c_2^4c_3 + 1488c_3^3c_4c_2^2 + 2064c_2^3c_3c_4^2 - 5280c_5c_2c_4^2 + 1104c_4c_7c_3 + 1200c_5c_6c_3 + 16320c_2^5c_5c_3 + 1536c_4c_2^3c_6 - 4512c_2^4c_4c_3^2 - 1200c_5c_2^2c_6 - 10392c_2^4c_5c_4 + 288c_2c_4^2c_3^2 + 7680c_5c_3^3c_2 + 4560c_2^3c_5^2 - 3240c_5c_2^7 + 2400c_4c_5^2 + 672c_6c_4^2 - 672c_4c_9 + 72c_2^2c_4^3 - 1296c_2^5c_4^2 - 456c_3c_4^3 \; 648c_4c_2^8 - 120c_6c_3^3 + 120c_6c_2^6)e_n^9 + O(e_n^{10})] \quad \textbf{(3.8)}$$

Using equations (3.4) to (3.8) in Algorithm 2.8, we get

$$x_{n+1} = \alpha + (12c_2^4c_3^2 - 8c_3^3c_2^2 - 8c_2^6c_3 + 2c_3^4 + 2c_2^8)e_n^9 + O(e_n^{10}) \;,$$

which implies that

$$e_{n+1} = (12c_2^4c_3^2 - 8c_3^3c_2^2 - 8c_2^6c_3 + 2c_3^4 + 2c_2^8)e_n^9 + O(e_n^{10})$$

which shows that new predictor corrector iterative method (PCIM) has 9th order of convergence.

COMPARISONS OF EFFICIENCY INDEX

The term "efficiency index" is used to compare the performance of different iterative methods. It depends upon the order of convergence and number of functional evaluations of the iterative method. If "r" denotes the order of convergence and "N_f" denote the number of functional evaluations of an iterative method, then the efficiency index E.I is defined as:

$$E.I = r^{\frac{1}{N_f}}$$

On this basis, the Newton's method [9,10] has an efficiency of $2^{\frac{1}{2}} \approx 1.4142$. HouseHolder method [10] has order of convergence three and the number of functional evaluations required for this method is three, so its efficiencies is $3^{\frac{1}{3}} \approx 1.4422$. The Abbasbandy method [1] has order of convergence three and number of functional evaluation required is four, so its efficiencies is $3^{\frac{1}{4}} \approx 1.3161$. Halley's method [2,12,19] has order of convergence three and the number of functional evaluations required for this method is three, so its efficiencies is $3^{\frac{1}{3}} \approx 1.4422$. Kuo [15] has developed several method that each require two function evaluations and two derivative evaluations and these methods achieve an order of convergence is six, so having efficiencies of $6^{\frac{1}{4}} \approx 1.5651$. Noor and Noor has developed several method that each require one function evaluation and two derivative evaluations and these methods achieve an order of convergence is five, so having efficiencies of $5^{\frac{1}{3}} \approx 1.7100$. Now we move to calculate the efficiency index of our new predictor corrector iterative method (PCIM) as follows: The new predictor corrector iterative method (PCIM) need one evaluation of the function and three of its first, second and third derivatives. So the number of functional evaluations of this method is four. *i.e*

$$N_f = 4$$

Also, in the earlier section, we have proved that the order of convergence of our new predictor corrector iterative method (PCIM) is nine. *i.e*

$$r = 9$$

Thus the efficiency index of new predictor corrector iterative method (PCIM) is:

$$E.I = 9^{\frac{1}{4}} \approx 1.7321.$$

The efficiencies of the methods we have discussed are summarized in Table given below.

Table 1. Comparison of efficiencies of various methods

Method	Number of function or Derivative evaluations	Efficiency index
NM, quadratic	2	$2^{\frac{1}{2}} \approx 1.4142$
HHM 3rd order	3	$3^{\frac{1}{3}} \approx 1.4422$
AM's 3rd order	4	$3^{\frac{1}{4}} \approx 1.3161$
HM 3rd order	3	$3^{\frac{1}{3}} \approx 1.4422$
Kou's 6th order	4	$6^{\frac{1}{4}} \approx 1.5651$
NNM 5th order	3	$5^{\frac{1}{3}} \approx 1.7100$
PCIM	4	$9^{\frac{1}{4}} \approx 1.7321$

It can be seen from the above comparison table that the efficiency of the new predictor corrector iterative method (PCIM) is much higher as compare to other iterative methods.

NUMERICAL EXAMPLES

We present some examples to illustrate the efficiency of the developed new predictor corrector iterative method (PCIM). We compare the Newton method (NM), the Halley's method (HM), the Householder's method (HHM), the Abbasbandy's method (AM), the Noor and Noor method (NNM) and new predictor corrector iterative method (PCIM) (Algorithm 2.8) introduced in this present paper. We choose following test examples, $f_1 = ln(x) + x,$ $f_2 = sin(x) - 10x + 10,$ $f_3 = e^x - 5x^2,$ $f_4 = xe^x - 1,$ $f_5 = x^3 - 4x^2 + x - 10.$

Table 2. Comparison of NM, HM, HHM, AM, NNM and PCIM

Method	N	N_f	$\lvert f(x_{n+1}) \rvert$	x_{n+1}
$f_1, x_0 = 1$				

NM	5	10	$1.686119e-22$	0.567143290409783872999968662210
HM	3	9	$1.616714e-19$	
HHM	3	9	$1.582099e-27$	
AM	3	12	$3.302671e-15$	
NNM	3	9	$1.616714e-19$	
PCIM	2	8	$1.820986e-55$	

Table 3. Comparison of NM, HM, HHM, AM, NNM and PCIM

Method	N	N_f	$\|f(x_{n+1})\|$	x_{n+1}
$f_2, x_0 = 2$				
NM	4	8	$2.302192e-20$	1.088597752397893618454937714710
HM	3	9	$3.695657e-30$	
HHM	3	9	$7.237667e-30$	
AM	3	12	$1.508024e-25$	
NNM	3	9	$3.695657e-30$	
PCIM	2	8	$3.322821e-90$	

Table 4. Comparison of NM, HM, HHM, AM, NNM and PCIM

Method	N	N_f	$\|f(x_{n+1})\|$	x_{n+1}
$f_3, x_0 = 1$				
NM	5	10	$2.582585e-18$	0.605267121314618484567862381243
HM	3	9	$6.812054e-15$	
HHM	4	12	$3.670165e-37$	
AM	4	16	$2.618014e-36$	
NNM	3	9	$6.812054e-15$	
PCIM	2	8	$1.358323e-41$	

Table 5. Comparison of NM, HM, HHM, AM, NNM and PCIM

Method	N	N_f	$\|f(x_{n+1})\|$	x_{n+1}
$f_4, x_0 = 0.5$				
NM	4	8	$3.450646e-20$	0.567143290409783872999968662210
HM	3	9	$8.740333e-39$	
HHM	3	9	$9.899022e-32$	
AM	3	12	$8.840229e-35$	
NNM	3	9	$8.740333e-39$	
PCIM	2	8	$2.610899e-113$	

Table 6. Comparison of NM, HM, HHM, AM, NNM and PCIM

Method	N	N_f	$\|f(x_{n+1})\|$	x_{n+1}
$f_5, x_0 = 4.5$				
NM	4	8	$3.964602e-17$	4.306913199721865187030462632425
HM	3	9	$2.363615e-31$	
HHM	3	9	$7.788400e-27$	
AM	3	12	$9.586776e-28$	
NNM	3	9	$2.363615e-31$	
PCIM	2	8	$3.189230e-93$	
$x_0 = 5.3$				
NM	5	10	$2.024500e-15$	4.306913199721865187030462632425
HM	4	12	$5.276438e-45$	
HHM	4	12	$2.475287e-34$	
AM	4	16	$1.928654e-36$	
NNM	4	12	$5.276438e-45$	
PCIM	2	8	$4.303821e-42$	

Tables 2-6 shows the numerical comparisons of the Newton's method (NM), the Halley's method (HM), the Househ"older's method (HHM), the Abbasbandy's method (AM), Noor and Noor method (NNM) and the new predictor corrector iterative method (PCIM) (Algorithm 2.8). The columns represent the number of iterations N and the number of functions or derivatives evaluations N_f required to meet the stopping criteria, and the magnitude $|f(x)|$ of $f(x)$ at the final estimate x_n.

CONCLUSIONS

A new predictor corrector iterative method (PCIM) for solving nonlinear functions has been obtained. We can concluded from tables (1-6) that,

1. The efficiency index of new predictor corrector iterative method (PCIM) is 1.7321.
2. The convergence order of new predictor corrector iterative method (PCIM) is nine.
3. By using some examples the performance of new predictor corrector iterative method (PCIM) is also discussed. New predictor corrector iterative method (PCIM) performance is very well as compared to NM, HM, HHM, AM and NNM as discussed in Table $(1-6)$.

REFERENCES

[1] S. Abbasbandy., *Improving Newton-Raphson method for nonlinear equations by modified Adomian de composition method*, Appl. Math. Comput. vol. 145, no. 2-3, pp. 887--893, 2003.

[2] I. K. Argyros, *A note on the Halley method in Banach spaces*,Appl. Math. Comput. 58 (1993), pp. 215--224.

[3] A. Ali, M. S. Ahmad, W. Nazeer and M. Tanveer, *New modified two-step Jungck iterative method for solving nonlinear functional equations.* Sci.Int. (Lahore), 27(4), 2959-2963, 2015. [4] A. Ali, Q. Mehmood, M. Tanveer, A. Aslam and W. Nazeer, *Modified new third-order iterative method for nonlinear equations.* Sci.Int.(Lahore),27(3),1741,1744,2015.

[5] A. Ali, M. S. Ahmad, M. Tanveer, Q. Mehmood and W. Nazeer, *Modified two-step fixed point iterative method for solving nonlinear functional equations,* Sci.Int.(Lahore), 27(3), 1737-1739, 2015.

[6] A. Ali, W. Nazeer, M. Tanveer, M. Ahmad., *Modified Golbabi and Javidi's method (MGJM) for solving nonlinear functions with convergence of order six.* Sci.Int. (Lahore), 28(1), 89-93 ,2015

[7] M. S. Ahmad, A. Ali, M. Tanveer, A. Aslam and W. Nazeer, *New fixed point iterative method for solving nonlinear functional equations.* Sci.Int.(Lahore),27(3),1815-1817,2015.

[8] E. Babolian and J. Biazar, *Solution of nonlinear equations by modified Adomian decomposition method,* Appl. Math. Comput, vol.132, no.1, 167--172, 2002.

[9] R. L. Burden, J. D. Faires, *Numerical Analysis (Sixth ed.),* Brooks/Cole Publishing Company, Calif., 1997.

[10] M. Frontini and E. Sormani, *Some variant of Newton's method with third order convergence,* Appl. Math. Comput, 140, 419-426.

[11] A. Golbabai, M. Javidi , *A third-order Newton type method for nonlinear equations based on modified homotopy perturbation method,* Appl. Math. Comput, 191 (2007) 199–-205.

[12] E. Halley., *A new exact and easy method for finding the roots of equations generally and without any previous reduction,* Phil. Roy. Soc. London 18 (1964) 136--147.

[13] H. Homerier., *A modified Newton's Method for root finding with cubic convergence,* J. Appl. Math. Comput. 157(1), 227-230.

[14] P. Jarratt, *Some efficient fourth order multipoint methods for solving equations*, BIT, 9, 119--124.

[15] J. Kuo, *The improvements of modified Newton's method*, Applied Mathematics and Computation, 189, 602-609.

[16] S. M. Kang, W. Nazeer, M. Tanveer, Q. Mehmood and K. Rehman, *Improvements in Newton-Rapshon Method for Nonlinear Equations Using Modified Adomian Decomposition Method*. International Journal of Mathematical Analysis, Vol.9,2015,no.39,1919-1928.

[17] S. M. Kang, W. Nazeer, A. Rafiq and C. Y. Youg, *A new third order iterative method for scalar nonlinear equations.* Int. Journal of Math. Analysis, Vol. 8, 2014, no. 43, 2141 – 2150

[18] M. S. Khan, A. Nazir, W. Nazeer., *ITERATIVE METHOD FOR SOLVING NONLINEAR FUNCTIONS WITH CONVERGENCE OF ORDER FOUR*,Sci.Int.(Lahore),28(1),77-81,2016.

[19] A. Melman., *Geometry and convergence of Halley's method*, SIAM Rev. 39(4) (1997) 728--735.

[20] T. J. McDougall, S. J. Wotherspoon, *A simple modification of Newton's method to achieve convergence of order* $\left(1+\sqrt{2}\right)$, Applied Mathematics Letters, vol. 29, no. March 2014, 20-25 (2014).

[21] W. Nazeer, A. Naseem, S. M. Kang and Y. C. Kwun Generalized Newton Raphson's method free from second derivative, J. Nonlinear Sci. Appl. (to appear)

[22] W. Nazeer, M. Tanveer, S. M. Kang and Y. C. Kwun, New third-order Fixed point iterative method for solving nonlinear functional equations and Polynomiography, J. Nonlinear Sci. Appl. (to appear)

[23] W. Nazeer., M. Tanveer,, S. M. Kang., and A. N., *A new Householder's method free from second derivatives for solving nonlinear equations and polynomiography.* J. Nonlinear Sci. Appl. 9 (2016), 998-1007

[24] W. Nazeer, S. M. Kang, M. Tanveer and A. A. Shahid, *Modified Two-step Fixed Point Iterative Method for Solving Nonlinear Functional Equations with Convergence of Order Five and Efficiency Index 2.2361*. Wulfinia Journal. *Vol 22, No. 5;May 2015.*

[25] W. Nazeer, S. M. Kang and M. Tanveer, *Modified Abbasbandy's Method for Solving Nonlinear Functions with Convergence of Order Six*, International Journal of Mathematical Analysis Vol. 9,2015, no.41,2011-2019

[26] K. Inayat Noor, M. Aslam Noor, Predictor–corrector Halley method for nonlinear equations, Appl. Math. Comput., in press, doi:10.1016/j.amc.11.023.

[27] F. Soleymani., *Optimal fourth-order iterative methods free from derivative*, Miskolc Math. Notes 12(2011), 255-264.

[28] J. Traub., *Iterative Methods for the solution of Equations, Prentice-Hall*, Englewood Cliffs, New Jersey, USA, 1964.

[29] M. Tanveer, M. Ahmad, A. Ali, W. Nazeer, K. Rehman., *Modified Householder's method (MHHM) for solving nonlinear functions with convergence of order six.* Sci.Int.(Lahore), 28(1), 83-87, 2016.

[30] S. Weerakoon and T. G. I. Fernando., *A variant of Newton's Method with accelerated third-order convergence*, Appl. Math. Letters, 2013, 87-93