UNIVERSITY OF CAPE COAST

MODIFIED ITERATIVE METHOD FOR COMPUTING THE APPROXIMATE SOLUTIONS OF NONLINEAR EQUATIONS

BY

PAUL ADU

Thesis submitted to the Department of Mathematics of the School of Physical Sciences, College of Agriculture and Natural Sciences, University of Cape Coast, in partial fulfillment of the requirements for the award of Master of Philosophy degree in Mathematics

JULY 2020

UNIVERSITY OF CAPE COAST

MODIDIED ITERATIVE METHOD FOR COMPUTING THE APPROXIMATE SOLUTIONS OF NONLINEAR EQUATIONS

PAUL ADU

2020

UNIVERSITY OF CAPE COAST

MODIFIED ITERATIVE METHOD FOR COMPUTING THE APPROXIMATE SOLUTIONS OF NONLINEAR EQUATIONS

BY

PAUL ADU

Thesis submitted to the Department of Mathematics of the School of Physical Sciences, College of Agriculture and Natural Sciences, University of Cape Coast, in partial fulfillment of the requirements for the award of Master of Philosophy degree in Mathematics

JULY 2020

DECLARATION

Candidate's Declaration

I hereby declare that this thesis is the result of my own original research and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature .... Date ....

Name: Paul Adu

Supervisor's Declaration

I hereby declare that the preparation and presentation of the thesis were supervised in accordance with the guidelines on supervision of thesis laid down by the University of Cape Coast.

Supervisor's Signature .... Date ....

Name: Prof. N. Mensah

ABSTRACT

This thesis concentrates on developing a Modified Iterative Method for computing the approximate solutions of nonlinear equations. We discus the concept of Error Analysis, Errors in Numerical Methods, Approximation and Convergence. Newton's method is discussed and proved. This study is set out to construct or develop a Modified Iterative Method for computing the approximate solutions of nonlinear equations by using Taylor Series expansion and Adomian Decomposition Method (ADM). The Taylor series is used in this study due to its higher possibility of convergence since it is a power series. In the same vain, the Adomian Decomposition method is a semi analytical method which decomposes the nonlinear equations into a series of functions thereby making the convergence of these functions much easier. The convergence of this method is proved to be of order 2. The Modifield Iterative Method is a modification based on Newton - Raphson's method. Matlab R2020a is used to compute the solutions of some numerical examples with the proposed modified method. The computation of the approximated solutions of the method are compared with some existing iterative methods in literature such as Newton's method, Karthikeyan's method and External Touch Algorithm method. Then we discussed the accuracy of the proposed modified iterative method when applied to single variable nonlinear equations. The study pointed out that, the modified method is comparable with the existing methods. Finally we concluded that the modified iterative method is more accurate than the Newton's method, the External Touch Algorithm method and even to some extent, the Karthikeyen's method.

KEY WORDS

Adomian Decomposition Method

Approximate Solution

Iterative Method

Newton's Algorithms

Nonlinear Equations

Pseudocode

ACKNOWLEDGEMENTS

First of all, I thank the Almighty God for giving me strength and knowledge to go through this program successfully. I would like to express my deepest appreciation and profound gratitude to my supervisor, Prof. Natalia Mensah; Mathematics Professor at the Department of Mathematics, University of Cape Coast (UCC) for her unflinching support, patience,encouragement and guidance throughout my MPhil Study. Prof. you actually inspired me and show me motherly love. Special thanks are extended to Prof. Ernest Yankson; the Head of Mathematics Department, University of Cape Coast (UCC) for his advice. I am also thankful to Dr. Kwabena Anokye Dompreh; a senior lecturer at Physics Department, University of Cape Coast (UCC) for persistence help on Matlab especially writing of the codes.

To my family, I say thank you for your understanding and unending spiritual and moral support throughout all the hard times. Furthermore, I wish to thank my closet friend Adusei - Opoku Afful of KASS and a colleague MPhil Mathematics student at that time for his assistance and encouragements.

DEDICATION

To the Almighty God and my beloved family.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ACRONYMS

ADM	Adomian Decomposition Method
ETA	External Touch Algorithms
KM	Karthikeyan's Method
NM	Newton's Method
PMIM	Proposed Modified Iterative Method

CHAPTER ONE

INTRODUCTION

Background to the Study

Computing the approximate solutions of nonlinear equations by iterative methods have become the obvious choice for mathematicians, scientist and researchers who are into numerical analysis and computing. Finding the roots (solutions) of equations is an important mathematical problem but there are only a few general classes of the equations of the form $f(x) = 0$ that can be solved analytically. These include linear equations and quadratic equations. Cubic and quartic (3rd and 4th degree polynomial) equations can also be solved by using complicated formulae. We usually solve these and other equations approximately by using numerical methods which are often iterative in nature with the help of a calculator or a computer software or programmes such as Maple, Matlab, R console, Fortran and Mathematica.

In computational mathematics, an iterative method can be defined as a mathematical procedure that uses an initial guess to generate a sequence of improving approximate solutions for a class of problems, in which each approximation is derived from the previous ones. A fundamental strategy behind many numerical methods is to replace a difficult scientific problem with a string of simpler ones and carry out a series of an iterative process with the expectation that, the solutions of the simpler problem will converge to the solutions of the original difficult problem. This strategy thrives well on finding zeros of functions. This means that, if we possess an arsenal of numerical methods for locating the zeros of functions, we shall be able to solve such problems.

In mathematics, a linear equation is defined as an equation that can be expressed in the form $\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + ... + \alpha_n x_n + \beta = 0$ where,

the variables (unknowns) are $x_1, x_2, ..., x_n$ and $\alpha_1, \alpha_2, ..., \alpha_n$ are coefficients which are often real numbers as well as
beta the constant term. The coefficients may be considered as the parameters of the equation and may be arbitrary expressions provided they do not contain any of the variables. Simply put, a linear equation in the variable x is an equation that can be written in the form $ax + b = 0$ where a and b are real numbers and $a \neq 0$. On the other hand, an equation is said to be a nonlinear equation if its general representation can be expressed in the form $ax_1^n + bx_2^{n-1} + cx_3^{n-2} + ... + p = 0$, $n \geq 2$. Here $x_1, x_2, x_3, ...$ are the variables and a, b and c are the coefficients while p is the constant term.

A system of nonlinear equations is a set of equations where one or more variables in at least one of the equations have degree two or higher. In numerical mathematics and computing, a root finding method or algorithm may be considered as an algorithm for finding the solution or roots of equations, also known as the zeros of continuous functions. The zeros or roots of a function f from real numbers to real numbers or from complex numbers to complex numbers is the number for which $f(x) = 0$. Generally speaking, the zeros of functions with higher degrees are difficult, expensive or at times impossible to compute exactly or expressed in closed form. However, root finding algorithms or methods provide an alternative approach which is approximation to zeros, expressed either as floating point numbers or as small isolating intervals, or disks for complex root.

In elementary algebra the solution of a first-order degree equation (linear) $ax + b = 0$ is given by the formulae $x = -\frac{b}{a}$ and the roots of second-order degree (quadratic) equation $ax^2 + bx + c = 0$ is given by the formulae

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Formulae also exist for the solutions of all cubic (third-order degree) equations and quartic (fourth-order degree) equations. Practically, they are hardly used due to the fact that they are complicated in their usage. In 1826, it was shown by a Norwegian Mathematician Neils Henrik Abel that it is highly impossible to construct a similar formulae for the solution of a fifth – degree or higher equations. For instance, for fifth-degree polynomial equations such as

$$x^5 - 9x^4 + 2x^3 - 5x^2 + 17x - 8 = 0$$

$$or$$

$$3x^5 + 2x^4 - 15x^3 + 12x^3 - 2x^2 - 6 = 0$$

It may be difficult or impossible to find the exact values for all of the solutions. In addition, non-polynomial equations such as $x - cosx = 0$ may have similar difficulties occurring. For such equations, the solutions are generally approximated by a numerical method (Burden & Faires, 2011). Solving nonlinear equations is one of the most important problems in numerical analysis.

In science and engineering, many of the non-linear and transcendental problems of the form $f(x) = 0$ are complex in nature. This is because it is not always possible to obtain the exact solution by the usual algebraic process. Numerical iterative methods are often used to obtain approximate solution of such problems. Much attention has been given to the development of several iterative methods for solving non-linear equations by researchers mostly in the field of numerical analysis. Numerical analysis is a very important branch of Mathematics and Computer Science that deals with the study of algorithms that make use of numerical approximation in mathematical analysis.

Although numerical analysis is considered by some to be a subject of recent origin and development, this is not in fact so. It deals with the derivation of results in the form of numbers, the numerical analyst is really the lineal descendant of the first caveman who enumerated the number of his wives by putting them into one; one correspondence with the fingers of his hand. The primary activity of the Babylonian scientist was the construction of mathematical tables. An example is the extant, which contains on a tablet the squares of the numbers from 1-60. Astronomical calculation formed a part of the activity of these early numerical analysts. Other energetic numerical analysts were the ancient Egyptians who constructed tables whereby complex fractions could be decomposed into the sum of simpler forms with unit numerators, and invented the method of false position.

For the Greek mathematicians, we find Archimedes in about 22.B.C, approximating the value of π and describing it as less than $\frac{22}{7}$ and greater than $\frac{233}{71}$. Heron the elder, in about 100 B.C made use of the iteration process $\sqrt{a} \sim \frac{1}{2}(x_n + \frac{a}{x_n})$ which is usually ascribed to Newton. Diophantus, about A.D.250, apart from his popular known work on indeterminate equations, was also responsible for a process for the arithmetic solution of quadratic equations. In the nineteenth century there occurred one of the triumphs of numerical analysis, the simultaneous prediction by Adams and Le Verrier in 1845, the existence and position of the planet Neptune (Andrew & Booth, 1966). Numerical analysis involves the study of methods of computing numerical approximations. One of the most studied problems of numerical approximation is the root finding problem. This process involves finding a root or a solution of an equation of the form

$$f(x) = 0, \tag{1.1}$$

for a given function *f*. A root of this equation is also called a zero of the function f.

In this business of numerical approximations, we are often interested in finding x such that $f(x) = 0$. where $f : R^n \to R^n$ denotes a system of n-nonlinear equations and x is the n-dimensional root. Methods employed to solve problems of this nature are called root-finding or zero-finding methods. It is worthwhile to note that the problem of finding a root is equivalent to the problem of finding a fixed-point. To see this consider the fixed-point problem of finding the n-dimensional vector x such that

$$x = g(x), \tag{1.2}$$

where $f : R^n \to R^n$. It can be inferred from the above two equations that we can rewrite equation (1) as a root finding problem by setting $f(x) = x - g(x)$ and likewise we can also recast a root finding problem into a fixed point problem by setting $g(x) = x - f(x)$. More often than not, it is not possible to solve such nonlinear equations analytically. When this occurs, we turn to numerical methods to approximate the solution. These methods employed are usually iterative. Generally speaking, algorithms for solving problems numerically can be divided into two main groups. That is direct methods and iterative methods. Direct methods are those which can be completed in a predetermined number of steps. Iterative methods are methods which converge to the solution over time. The problem of finding an approximate solution to the root of an equation is traced back at least to $1700B.C$.

A cuneiform table in the Yale Babylonian collection dating from that period gives a sexigestimal (base 60) numbers equivalent to 1.414222 as an approximation to $\sqrt{2}$ as a result that is accurate to within 10^{-5} (Burden & Faires, 2011). Solving system of nonlinear equations is one of the most important problems in most numerical computations especially in a diverse range of engineering applications. Most applied problems can be reduced to solving system of nonlinear equations which is one of the well-known problems in mathematics. Its applications in many scientific fields cannot be over emphasized. Consequently, great efforts have been made in recent times by a lot of researchers and many constructive theories and algorithms have been proposed to solve system of nonlinear equations (Li et al., 2015).

Statement of the Problem

Finding the roots of a system of nonlinear equations is one of the important problems in most numerical computations especially in areas associated with engineering applications. Many applied problems can be transformed into solving a system of nonlinear equations which is one of the well-known problems in mathematics. There are countless applications in many scientific related fields. As a result of this, many great efforts have been made in recent years by a lot of researchers and many constructive theorems and algorithms are proposed to solve system of nonlinear equations. Inspite of these, there still exist some problems with numerical methods for solving a system of nonlinear equations. For most traditional methods, for instance, the bisection method convergence is very slow and cannot detect multiple roots even though it is the method that enhances good choice of initial guest of the solution. Newton or Newton-Raphson method is a popular method known for its error decreasing rapidly with each iteration thereby converging very fast to the solution.

However, the convergence and performance characteristics is highly dependent on the good initial guess of the solution. For a bad choice of initial guess (x_0), it may take many iterations to converge to the solution or in the worse case, the method can fail to converge. Another point worth mentioning is that, the results obtained may be improper if the initial guess of the solution is unreasonable. Many different combination of methods and many intelligent algorithms such as particle swarm algorithms and genetic algorithms are applied to solve system of nonlinear equations which can overcome the problem of selecting a reasonable initial guess of the solution. Li et al. (2015) explained that, these algorithms are too complicated or expensive to calculate when there are a number of system of nonlinear equations to solve. Here, a modified iterative method based on the Newton's method is proposed which can overcome the dependence on a reasonable initial guess of the solution and the complex nature of using intelligent and other algorithms. This proposed modified method to some extent is user friendly and improve computational efficiency.

Research Objectives

The objectives of this thesis are as follow:

(i) To construct and prove new modified iterative method to compute the approximate solution of nonlinear equations.

(ii) To establish the convergence of the proposed modified iterative method.

(iii) To compute the approximate solution of some nonlinear equations using the modified iterative method and Matlab software.

Significance of the Study

Most applied problems in physical sciences, biological sciences and engineering can be transformed into system of nonlinear equations. The roots finding of these equations, is therefore crucial to solving contemporary real life problems. Therefore, the efficiency and robustness of the methods used in solving such equations are very paramount to researchers and mathematicians alike. This study is aimed at developing an improved numerical method for solving nonlinear equations that will give a better approximate solutions. It will also add to the body of knowledge in this area of numerical analysis thereby setting the stage for further research in the field of numerical solution of nonlinear equations.

Delimitation

This study is an iterative numerical method for finding the roots of non-linear equations based on the method proposed by Newton-Raphson. The proposed iterative algorithm is particularly suited for finding the zeros of functions for which the derivatives are easy to compute. One class of such functions are polynomial functions. Another important class are functions defined via integrals like $f : x \to \int_a^x F(t)dt + G(x)$.

Limitation

In this study, all the numerical solutions of the nonlinear equations used in this study were computed using Matlab R2020a on my personal computer. The researcher therefore admit that, due to truncation error, the results obtained might have been affected marginally. It may also be possible that when a different software such as R console, Maple, Fortran, Mathematica and so on is used to compute the same non-linear equations on a different computer or advanced calculators, the precision of the results may be at variance with what is obtained in this study.

Organisation of the Study

The thesis is structured in five chapters. Chapter one actually deals with introduction of the study. Here detailed explanations and discussions on iterative numerical method is given, key words in the topic of the thesis are all well explained. This chapter also talks about the background of the study, statement of the problem, objectives of the research, methodology. The rest of the areas under this chapter are significance of the study, delimitation and the limitation of the study.

The chapter two is divided into two sections. The first section has some mathematical concepts directly related to the thesis topic whiles the remaining section has the relevant literature review of the study. This section reviews relevant previous studies done by other researchers in the past either published or not. The convergence, the strengths and weaknesses of these reviewed numerical methods are discussed. In the literature review, the knowledge gap created or filled by the various iterative methods proposed by other researchers are also highlighted in chapter two. Chapter three contains the methodology of this work, here, some mathematical concepts such as conditional and absolute convergence are captured in this chapter.

Adomian decomposition method and Taylor series together with the derivation of Newton – Raphson's method are also featured in chapter three. The aim of this work is to develop a modified iterative method to compute the approximate solutions of nonlinear equations. So the derivation of this proposed modified iterative method and the proof of its convergence are given in chapter four. Some numerical examples of nonlinear equations are given and each numerical method is solved by using Matlab software. Here, the results or findings of each numerical method is analyzed. Finally, the fifth and last chapter of this work talks about summary, conclusions and recommendations.

Chapter Summary

This chapter serves as the introduction to the study. It begins with the background which actually put the problem under study into perspectives. This is followed by the statement of the problem which highlights the knowledge gab of which researchers in that field have failed to resolve. Research objectives which must be achieved at the end of the research is covered in this chapter.The method employed to achieve the stated objectives is discussed. The Taylor series as well as the Adomian decomposition method together with Matlab R2020a software are used in this study. The benefits of the results as well as those who are direct beneficiaries of this study are also outlined.

The kind of equations suitable for this proposed modified method such as equations whose derivatives can be computed easily is discussed. This chapter also highlights on the possibility of obtaining results which are at variance with the results in the work due to the iterative method truncation, the software used as well as the specifications of computer used. Finally how the thesis is organized as far as each chapter is concerned is done in this chapter.

CHAPTER TWO

LITERATURE REVIEW

Introduction

This Chapter presents some discussion on mathematical concepts directly related to numerical methods which are iterative in nature. These concepts are Approximation, Error in Numerical Methods, Errors Analysis. The rest of this chapter is a review on relevant literature on some numerical methods and their convergence.

Errors in Numerical Methods

Any approximation of a function necessarily allows a possibility of deviation from the correct value of the function (Ledder, 2005). However, approximations are inevitable in situations where it is extremely difficult or in the worst situation not possible to get the true value or the exact solution analytically. Error is the term used to denote the amount by which an approximation fails to equal the exact solution or the true value. Whenever any numerical method is applied to system of equations, two forms of error surfaces. These are Truncation and Round off errors. Truncation error in numerical method, is any error that is caused by using simple approximations to represent exact mathematical formulas.

Truncation error comes from the approximation that is inherent in numerical algorithms. For instance, if you use the first n terms of a series in methods that are based on series, you have truncated the series and the method as well. The effect of those ignored terms are called the truncation error. The only way to avoid truncation error completely is to make use of exact calculations. Ledder (2005), suggested that, truncation error can be reduced by applying the same approximation to a larger number of smaller intervals or by switching to a better approximation.

Truncation error analysis is the single most important source of information about the theoretical characteristics that distinguish good methods from poor ones. One can estimate truncation error accurately by a combination of theoretical analysis and numerical experiments. The truncation error also occurs as a result of the conversion of continuous function to a discrete approximation for numerical evaluation. Therefore truncation error is also known as discretisation. Truncation error is composed of two parts namely Local Truncation and Global Truncation Error. Thus, we seek information about errors on both a local and global scale.

Local Truncation Error arises when a numerical method is used to solve initial value problem. It is the amount of truncation error that occurs in one step of a numerical approximation. The local truncation error is defined by T_{n+p} and introduce the local error at x_{n+p}. It is shown as

$$T_{n+p} = y(x_{n+p}) - y(x_n) - h\phi(x_n, y(x_n); h)$$

This error occurs after the first step and form in each step.

$$T_{n+1} = y_n(x_{n+1}) - y(x_{n+1}).$$

Global (or Accumulated) Truncation Error may be defined as the total accumulated over all solution steps. It is denoted by E_n and it is expressed as

$$E_n = y_n - y(t_n),$$

where, y_n is exact solution and $y(t_n)$ is the approximate solution.

Global truncation error is the amount of truncation error that occurs in the use of a numerical approximation to solve a problem. This error is caused by the accumulation of local truncation error in all of the iterations. The Round off Error is another form of error in numerical methods. This error originates due to the fact that a discrete number of significant digits is used to represent real numbers which have infinite digits on computers. Computer representation of numerical values is limited in terms of magnitude and precision. Magnitude here means there are upper and lower bounds on the magnitude of numbers that can be represented whiles precision about the fact that not all numbers can be represented exactly. For instance, according to Burden & Faires (2011), error due to rounding off should be expected whenever computations are performed using numbers that are not powers of two.

Every computer has only a finite word length and a finite total capacity, so only numbers with a finite number of digits can be represented and these real numbers represented in computers are called its machine numbers. Every numerical computation with computer system, must conform to normalized floating point representation format, it must have a finite expansion. As a result, numbers that have non-terminating expansion cannot be represented precisely. Moreover, a number that has a terminating expansion in one base may have a non-terminating expansion in another base (Ward Cheney & Ronald Kincaid, 2008).

The error that results from replacing a number with its floating points is called round off error regardless of whether the rounding off or chopping off method is used. Some types of mathematical operations are more susceptible to round off errors. These are large computations, adding large and small numbers as seen in an infinite series and inner product. Since computers can retain a large number of digits in a computation, round off error is problematic only when the approximation requires that the computer subtract two numbers that are nearly identical (that is subtraction cancellations).

This is exactly what happens if we apply an approximation to intervals that are too small. Thus the effort to decrease truncation error can have the unintended consequence of introducing significant round off error (Ledder, 2005). Practitioners of numerical approximation are most concerned with truncation error, but they also try to restrict their efforts at decreasing truncation error to improvements that do not introduce significant round off error.

Error Analysis

In approximation theory, error is inevitable. Therefore error analysis is very crucial in determining how efficient and robust is the numerical method employed. Error in any approximation is defined by

$$E_r = T_v - A_v,$$

where, $E_r = Error$, $T_v =$ True value and $A_v =$ Approximate value. Suppose the linearization of f about a is used to approximate $f(x)$, that is,

$$f(x) \approx L(x) = f(a) + f'(a)(x-a)$$

The error $E_r(x)$ in this approximation is given by

$$E_r(x) = f(x) - L(x)$$

This implies that,

$$E_r(x) = f(x) - [f(a) + f'(a)(x-a)]$$

It is the vertical distance at x between the graph of f and the tangent line to that graph at $x = a$. It is observed that whenever x approaches a, then the error in this approximation $E_r(x)$ becomes small as compared to the horizontal distance between x and a. For iterative processes, the error c an be approximated as the difference in values between successive iterations. Approximation errors can be measured by two different methods namely Absolute and Relative Errors. Absolute Error ($|E_r|$) is the absolute difference between the true value and the approximate value. Suppose that p^* is an approximation to p. Then the absolute error involved in approximating p with p^* is given by

$$|E_r| = |p - p^*|$$

Relative Error is the absolute difference between the true value and the approximate value divided by the true value. That is, the ratio of the absolute error and the true value. It is expressed as

$$R_E = \frac{|p - p^*|}{p}, p \neq 0.$$

Burden & Faires (2011) admitted that, the relative error is generally a better measure of accuracy than the absolute error because it takes into consideration the size of the number being approximated. In most cases when performing calculations, we are not concerned with the sign of the error but are interested in whether the absolute value of the relative error is smaller than pre-specified tolerance. For such cases, the computation is repeated until $|\varepsilon_a| < \varepsilon_s$. This relationship is called stopping criterion.

Approximation

Many problems in applied mathematics are too difficult to be solved analytically to obtain exact solutions. All that the practitioners in this field hope to do is to find an approximate solutions that are correct to within some acceptably small tolerance. A tangent to a curve $y = f(x)$ at $x = a$ describes the behavior of the graph near the tangential point $M(a, f(x))$ better than any other straight line through M because it goes through M in the same way as the curve $y = f(x)$. The tangent line has equation

$$y = f(a) + f'(a)(x - a)$$

and the approximation

$$f(x) = f(a) + f'(a)(x - a)$$

is called the linear approximation or tangent line approximation of f at a. The linear function whose graph is the tangent

$$L(x) = f(a) + f'(a)(x - a)$$

is called the linearization of f about a (Adams & Essex, 2010). Linearization of mathematical problems is common throughout applied mathematics and numerical analysis. The idea is that, it might be easy to calculate a value $f(a)$ of a function, but difficult or even almost impossible to compute nearby values of f. So we go for the easy computed values of the linear function $L(x)$ whose graph is the tangent line of f at $(a, f(a))$ as an approximation of the curve $y = f(x)$ when x is near a (Stewart, 2008). There is a formulae for solving third – degree (cubic) equation $ax^3 + bx^2 + cx + d = 0$.

This formulae is given by

$$x = \sqrt[3]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right) + \sqrt{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3}}$$
$$+ \sqrt[3]{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right) - \sqrt{\left(\frac{-b^3}{27a^3} + \frac{bc}{6a^2} - \frac{d}{2a}\right)^2 + \left(\frac{c}{3a} - \frac{b^2}{9a^2}\right)^3}} - \frac{b}{3a}x$$

There is a formulae for the solutions of quartic (fourth - degree) equation though it is hardly use in practice due to its complicated nature. Abramowitz & Stegun (1972), indicated that, the formulae for solving the quartic equation $z^4 + a_3z^3 + a_2z^2 + a_1z + a_0 = 0$ is given by

$$\begin{aligned}
z &= x + \frac{1}{2}(a_3 \pm \sqrt{a_3^2 - 4a_2 + 4y_1})x + \frac{1}{2}(y_1 \pm \sqrt{y_1^2 - 4a_0}), \\
Z_1 &= -\frac{1}{4}a_3 + \frac{1}{2}R + \frac{1}{2}D \\
Z_2 &= -\frac{1}{4}a_3 + \frac{1}{2}R - \frac{1}{2}D \\
Z_3 &= -\frac{1}{4}a_3 - \frac{1}{2}R + \frac{1}{2}E \\
Z_4 &= -\frac{1}{4}a_3 - \frac{1}{2}R - \frac{1}{2}E
\end{aligned}$$

where, $R = \sqrt{\frac{1}{4}a_3^2 - a_2 + y_1}$

$$D = \begin{cases} \sqrt{\frac{3}{4}a_3^2 - R^2 - 2a_2 + \frac{1}{4}(4a_3a_2 - 8a_1 - a_3^3)R^{-1}}, & R \neq 0 \\ \sqrt{\frac{3}{4}a_3^2 - 2a_2 + 2\sqrt{y_1^2 - 4a_0}}, & R = 0 \end{cases}$$

$$E = \begin{cases} \sqrt{\frac{3}{4}a_3^2 - R^2 - 2a_2 - \frac{1}{4}(4a_3a_2 - 8a_1 - a_3^3)R^{-1}}, & R \neq 0 \\ \sqrt{\frac{3}{4}a_3^2 - 2a_2 - 2\sqrt{y_1^2 - 4a_0}}, & R = 0 \end{cases}$$

In 1826, it was shown by a Norwegian Mathematician Neils Henrik Abel that, it is highly impossible to construct a similar formulae for the solution of a fifth or higher degree equations. For instance, a fifth-degree polynomial equations such as

$$x^5 - 4x^4 - 7x^3 + 14x^2 - 44x + 120 = 0$$

may be difficult or impossible to find the exact values for all of the solutions. In addition, non-polynomial equations such as

$$x - cosx = 0$$

may have similar difficulties occurring. For such equations the solution are generally approximated by a iterative numerical method (Anton et al., 2012).

Some Iterative Numerical Methods and Their Convergence

Solving nonlinear equations is one of the most important problems in numerical analysis. Much attention has been given to developing several iterative methods for solving nonlinear equations by researchers. A lot of iterative methods for solving non-linear equation $f(x) = 0$ have been proposed. Most of these methods are based on the Newton's method or the secant method. The Newton's method is seen as one of the most efficient method used to solve nonlinear problems. It serves as one of the fundamental tools in numerical analysis, control theory as well as operation research. Newton's method has countless applications in management science, data management, medicine, optimization and engineering. Sir Isaac Newton developed this method in 1669 and it was modified in 1690 by Joseph Raphson and subsequently termed it as Newton – Raphson method. This modified method is used in finding successive better approximations to the roots of the function $f(x)$.

The Newton method depends on an initial good guess and the behavior of the function $f(x)$ near the root. Also the derivative $f'(x)$ cannot be left out in the implementation of the Newton's method. This method converges quadratically. That is the order of convergence is 2. The secant method was specifically developed to overcome the above problems posed by Newton's method but it is slow to converge. According to Burden & Faires (2011), Muller's method will give a rapid convergence without a particular good initial approximation and also has added advantage of being able to approximate complex roots, a major drawbacks of both Newton and Secant methods. It is not quite as efficient as the Newton's method because its order of convergence near the root is approximately $\alpha = 1.84$ as compared to Newton's method of quadratic order $\alpha = 2$.

However, it is better than the Secant method whose order is approximately $\alpha = 1.62$. Brent (2013) suggested a hybrid method that combines the bisection and the secant method to overcome the problems associated with the Newton's method. Meanwhile, in respective of numerous research work already done on Modified Newton's method, researchers continued to work for new improved methods, that are efficient, user friendly, effective, robust and will produced better results or approximations.

Newton's method serves as foundation for many recent proposed numerical methods for solving nonlinear equations. Kanwar et al. (2003) studied new numerical techniques for solving nonlinear equations and suggested a modification of Newton's method which he called an external touch algorithms for solving non-linear functions. Chun (2005) modified Newton Method using Adomian decomposition method conjectured that $n = m, n \geqslant 0$ case produces an iteration scheme of order $(n+2)$ and established that the decomposition method produces efficient results when applied as a corrector to the Newton's method. Basto et al. (2006) constructed a new efficient iterative method for solving nonlinear equations and established cubic order of convergence. This method was based on the proposal of Abbasbandy on improving the Newton-Raphson method for nonlinear equations by modified Adomian decomposition method. Basto et al. (2007) in a numerical study established that the application of Pade approximants (PAs) to the truncated series solutions given by Adomian decomposition technique non-linear equations in particular to Burgers equation can improve the rate of convergence or enlarge the convergence domain (radius of convergence).

His work solved the problem created by the application of Adomian decomposition method to partial differential equations, when the exact solution is not reached, demands the use of truncated solution series which may have small convergence radius and the truncated series may be inaccurate in many regions. Homeier (2005) came out with a Newton –typed method by using quadrature formulae. He later proved that, his new method converges cubically. After comparative analysis of his proposed method and the original Newton's method, he established that, quadrature formulae is effective in making Newton's method have an efficient convergence.

Yun & Petkovic (2011) proposed a simple but efficient iterative method for finding a root of non-linear equations. The study shows that the new method does not need the derivative of $f(x)$ nor the effort to choose good initial guess and obtained quadratic order of convergence. Another setback of the Newton's method is that the condition $f'(x) \neq 0$ in a neighborhood of the required root is problematic indeed for the convergence of the method and its application is restricted. Some modified Newton's method have such problem which restricted their applications. It is against this back drop that Kou et al. (2007) proposed a class of new iterative methods, in which $f'(x) = 0$ in some points is permitted. Convergence analysis show that the new iterative methods converge cubically.

Some researchers in recent times have studied and proposed several new iterative methods for non-linear equations with higher order convergence by using Adomian decomposition method which employs higher order differential derivatives which is a major drawback. Feng (2009) developed a two-step method for nonlinear equations which does not involve higher order derivative of the function and obtained quadratic convergence. This method compute well with other researchers such as Newton-Raphson, Adomian, Babolian, Abbasbandy and Basto proposed methods. Karthikeyan (2010) reviewed the external touch method proposed by Kanwar et al. (2003) and concluded that, even though it converges faster than the Newton's method for some functions, but it is not generally efficient than the Newton method.

He then proposed a new modified Newton's iterative method known as efficient algorithm for minimization of non-linear function which is an update of the work of (Basto et al., 2006). Though this method also converges quadratically, the rate of convergence is faster than the Newton's method and the External Touch technique. Yun (2008), proposed a non-iterative method for non-linear equations in order to overcome the difficulties of good choice of initial guess and improper behavior of $f(x)$ in using Newton and other existing iterative methods.

The method uses two kinds of transforms of $f(x)$ based on hyperbolic tangent function, $\tanh(\beta)x$ and a signum function $sgn(x)$. This method reduces solving nonlinear equation to evaluating an integral of the transformed function. Zavalaus (2014) showed that the quadrature formulae has a third-order convergence and concluded that the quadrature formulae is more efficient than the Newton's method. This work is an update to the method of Homeier (2005) and a modification of Newton method. Mitlif (2014) considered new iterative method for solving nonlinear equations.

This is an efficient three steps iterative method for finding the roots of the nonlinear equation. The convergence analysis is proved to establish its five order of convergence. The new method is comparable with well-known existing methods in literature and in many cases gives better results. The Newton's method is not applicable if the derivative of any function is not defined in any interval. The Newton's method was therefore modified by Steffensen who replaced the first derivative $f'(x)$ by forward difference approximation

$$f'(x) \approx \frac{f(x_n + f(x_n)) - f(x_n)}{f(x_n)}$$

and obtained the famous Steffensen method

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

Steffensen's method is of quadratic convergence and required two functional evaluations per iteration (Ward Cheney & Ronald Kincaid, 2008). Cordero et al. (2012) proposed Steffensen - type method by approximating the derivatives in the well-known fourth-order Ostrowski's method and the sixth-order improved Ostrowski's method by central-difference quotient. The modification of these methods are free from derivatives and proved that the methods obtained preserve their convergences order 4 and 6. Yasmin & Junjua (2012) proposed two new derivative free iterative methods for solving nonlinear equation $f(x) = 0$.

These researchers developed these efficient methods to find the approximation of the root of the nonlinear equation $f(x) = 0$ without the evaluation of the derivative. These new proposed methods are based on central-difference and forward-difference approximations to derivatives. It is proved that one of the method has cubic convergence and the other method has fourth-order convergence. Ujević (2006) and Wu & Fu (2001) did a similar work in this direction. Their works also suggested derivative free iterative method for solving nonlinear equations of the form. However, both works were a one-step quadratically convergence iterative scheme. Ahmad et al. (2012) proposed and analyze two two-step derivative free algorithms for solving non-linear equation of the form $f(x) = 0$. This new two two-step iterative methods combine (Ujević, 2006) and (Wu & Fu, 2001) who suggested derivative free iterative method for solving non-linear equation $f(x) = 0$. It is proved that this new two-step iterative method has quadratic convergence.

Numerical comparison show that, the new developed two-step algorithms are comparable with the existing algorithms and are successful in case where the existing algorithms have fail to converge or have numerical difficulties. Akinsunmade (2016) suggested a modified Newton's method for solving nonlinear programming problem. The method is constructed from Taylor's series expansion and Adomian decomposition method. A comparative study of the new method with Newton's method and other existing methods developed in recent times showed that, the method is more reliable and converges quadratically but faster than the Newton's method as well as some modified Newton's method for solving nonlinear equations in optimization. Inspired and motivated by research going on in this area, we have proposed a modified iterative method for solving a single variable nonlinear equations.

Chapter Summary

This chapter is in two parts. The first part covers some discussion on mathematical concepts directly related to numerical methods which are iterative in nature. These concepts are Errors in numerical methods, Error Analysis and Approximation. The other part of this chapter presents relevant literature on some numerical methods and their convergence. In this literature review, the strengths and weakness of each method is discuss. We also highlighted the knowledge gab created or filled by each of these numerical methods.

CHAPTER THREE

METHODOLOGY

Introduction

This chapter is divided into two parts. The first part deals with mathematical preliminaries in which we discussed convergence analysis. The convergence analysis here covers some forms of convergence analysis namely the absolute convergence, conditional convergence as well as quadratic convergence. The other part focus on the theorems and proof of Taylor series, the Newton-Raphson's method as well as the generation of few Adomian polynomials. These are the methods employed in this thesis to develop the modified iterative method for computing the approximate solutions of nonlinear equations.

Convergence Analysis

Complicated functions can be frequently expressed as series of simpler functions. The convergence of these series of simpler functions implies convergence of the complicated function. The series $\sum_{n=1}^{\infty} a_n$ converges to the sum S that is $\sum_{n=1}^{\infty} a_n = S$ if $\lim_{x\to\infty} S_n = S$ where, S_n is the nth partial sum of $\sum_{n=1}^{\infty} a_n$ where, $S_n = a_1 + a_2 + a_3 + ... + a_n = \sum_{j=1}^{n} a_j$. Thus a series converges if and only if the sequence of its partial sums converges. However, it diverges to infinity or negative infinity if the sequence of the partial sum does so. This therefore means that, the convergence of the series $\sum_{n=1}^{\infty} a_n$ depends on the convergence of the partial sun sequence $S_n = \sum_{j=1}^{n} a_j$ but not a_n

Theorem 1. *If* $\sum_{n=1}^{\infty} a_n$ *converges, then the* $\lim_{x\to\infty} a_n = 0$*. If the* $\lim_{x\to\infty} a_n$ *does not exit or* $\lim_{x\to\infty} a_n \neq 0$*, then the series* $\sum_{n=1}^{\infty} a_n$ *diverges.*

Proof. Given that

$S_1 = a_1$

$S_2 = S_1 + a_2$

$S_3 = S_2 + a_3$

.

.

.

$S_n = S_{n-1} + a_n$

$S_n = a_1 + a_2 + a_3 + ...a_n = \sum_{j=1}^{n} a_j$

$\Rightarrow S_n - S_{n-1} = a_n$

If $\sum_{n=1}^{\infty} a_n$ converges, then $\lim_{x\to\infty} S_n = S$ exist and $\lim_{x\to\infty} S_{n-1} = S$.

Then $\lim_{x\to\infty} a_n = \lim_{x\to\infty} S_n - \lim_{x\to\infty} S_{n-1}$

Hence $\lim_{x\to\infty} a_n = S - S = 0$ □

. The above theorem plays an important role in the understanding of infinite series. However, the converse is not true in general. That is if $lim_{x\to\infty} a_n = 0$, then $\sum_{n=1}^{\infty} a_n$ must converge. The harmonic series is a counterexample showing the falsehood of this assertion. Thus $lim_{x\to\infty}\frac{1}{n} = 0$ but $\sum_{n=1}^{\infty}\frac{1}{n}$ diverges to infinity (Adams & Essex, 2010). If we find that $\lim_{x\to\infty} a_n \neq 0$, we are certain that $\sum a_n$ is divergent but if $lim_{x\to\infty} a_n = 0$, we know nothing about the convergence or the divergence of $\sum a_n$. Some convergence series such as geometric and telescoping series whose sums could be determined exactly because the partial sum S_n could be expressed in closed form as explicit of n whose limit as $n \to \infty$ could be evaluated. It is not usually possible to do this with some given series and therefore it is not feasible to determine the partial sum of the series exactly. However, there are many techniques for determining whether a given series converges and if it does, for approximating the sum to any desired degree of accuracy. Some of these techniques are the integral test, comparison test, ratio test, and root test.

Absolute Convergence

A series $\sum_{n=1}^{\infty} a_n = a_1 + a_2 + a_3 + a_4 + ... + a_n + ...$ is said to be absolutely convergent if the series of absolute values $\sum_{n=1}^{\infty} |a_n| = |a_1| + |a_2| + |a_3| + |a_4| + ... + |a_n| + ...$ converges and is said to diverge absolutely if the series of absolute values diverges.

Theorem 2. *If the series* $\sum_{n=1}^{\infty} |a_n| = |a_1| + |a_2| + |a_3| + |a_4| + ... + |a_n| + ...$ *converges then so does* $\sum_{n=1}^{\infty} a_n = a_1 + a_2 + a_3 + a_4 + ... + a_n + ...$

Proof. Let $\sum_{n=1}^{\infty} a_n$ be absolutely convergent and $b_n = a_n + |a_n|$ for each n

since $-|a_n| \leqslant a_n \leqslant |a_n|$

then $|a_n| - |a_n| \leqslant a_n + |a_n| \leqslant |a_n| + |a_n|$

$0 \leqslant a_n + |a_n| \leqslant 2|a_n|$

But $b_n = a_n + |a_n|$

$\Rightarrow 0 \leqslant b_n + 2|a_n|$

Thus $\sum_{n=1}^{\infty} b_n$ converges by the comparison test

From $b_n = a_n + |a_n|$, we obtained $a_n = b_n - |a_n|$

$\Rightarrow \sum_{n=1}^{\infty} a_n = \sum_{n=1}^{\infty} b_n - \sum_{n=1}^{\infty} |a_n|$ also converges. □

The converse of this theorem is false since alternating harmonic series is a counter example. Thus $\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + ...$ converges although it does not converge absolutely. If all the terms are replace by their absolute values, we get divergent harmonic series

$\sum_{n=1}^{\infty} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ... = \infty$

Conditional Convergence

If $\sum_{n=1}^{\infty} a_n$ is convergent but not absolutely convergent, then we say that it is conditionally convergent or that it converges conditionally. Example of conditional convergent series is the alternating harmonic series. Petrovic (2013) acknowledged that,many 18th century mathematicians ignored the difference between the absolute and the conditional convergence in spite of the evidence to the contrary. It was well known that the harmonic series diverges and that the alternating harmonic series converges. Cauchy was the first to make this distinction.

Quadratic Convergence

In numerical analysis, the order of convergence and the rate of convergence of a convergent iterative method are quantities that represents when and the rapidity with which the iterative method converges to the root. A sequence x_n that converges to L is said to have order of convergence $q \geq 1$ and the rate of convergence μ if

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|^q} = \mu$$

The rate of convergence μ is also called asymptotic error constant. It is therefore possible that two iterative methods may have the same order of convergence but one method may converge faster than the other because of different rate of convergence. A sequence is said to converge with order q to L for $q \geq 1$ if

$$\lim_{n\to\infty} \frac{|x_{n+1} - L|}{|x_n - L|^q} < M$$

for some positive constant $M > 0$. If $q = 2$, then it is quadratic convergence.

Quadratic convergence means that the square of the error at one iteration is proportional to the error at the next iteration. For instance, if the error in one iteration is one significant digit, at the next iteration, it is two digits, then next four digits and so on. Thus a doubling (approximately) with each iteration. The value of x-iterations show the same doubling. this doubling is referred to as quadratic convergence.

Adomian Decomposition Method (ADM)

Most of scientific problems and phenomena occur nonlinearly. Only a limited number of such problems have a precise analytical solution. In the 1985, a mathematician by name George Adomian developed a powerful decomposition method for solving linear or nonlinear and deterministic or stochastic operator equations, including ordinary differential equations, partial differential equations and so on (Rudall & Rach, 2008). George Adomian was at that time chair of the center for Applied Mathematics at the University of Georgia. His method is known as Adomian Decomposition Method (ADM) which is a semi -analytical method for solving nonlinear equations.

This technique is based on the representation of a solution to a functional equation as series of functions. Each term of the series is obtained from a polynomial generated by a power series expansion of an analytic function. The crucial aspect of the method is the employment of the Adomian polynomials which allows for solution convergence of the nonlinear portion of the equation without simply linearizing the system. In recent years, Adomian and other researchers have successfully applied his decomposition method to algebraic equations, ordinary, partial, delay, and non-integer order or fractional differential equations for a wide class of nonlinearities, including polynomial, exponential, trigonometric, hyperbolic, composite, negative power, radical and even fractional or decimal power nonlinearities. Adomian's decomposition method gives us the liberty to solve nonlinear differential equations without having to appeal to the decidedly questionable practices of perturbation or linearization. Although the abstract formulation of Adomian method is very simple, the calculations of the polynomial and the verification of convergence of the function series in specific situation are usually a difficult task.

In view of this, Abbaoui & Cherruault (1994) have reported a new but different formula for fast calculation of the Adomian polynomials, and have developed software that has quickly generated and listed the classical Adomian polynomials from A_0 to A_{100} inclusively. Yang (1994) of the Institute of Applied Physics and Computational Mathematics in Beijing, China, and Jinqing (1993) of the Institute of Atomic Energy in Beijing, China, have also each developed software for rapid generation of the Adomian polynomials, a notion due to Adomian (1976), which is key in solving nonlinear equations, and which notion was named the Adomian polynomials by (Rach, 1984) in obvious recognition of Adomian's breakthrough in mathematics. This method has numerous and varied advantages.

Rach (1984) indicated that, a very important advantage of Adomian's basic method is the elimination of a number of restrictive and generally unsatisfactory assumptions on the nature of stochastic processes, the magnitude of fluctuations, or on the nonlinearities which are inherent in other methods. No linearization or closure approximations are necessary. One doesn't require "weak" nonlinearities or "small" fluctuations, stationarity, gaussian or white noise behavior, etc. Thus, the physical system is not forced into a nice mathematical mold for which solutions are readily available. Babolian & Biazar (2002) in their study on the order of convergence of the Adomian decomposition method admitted that, the Adomian decomposition method is a simple and powerful tool for obtaining the solution of functional equations. They however suggested that, it would be desirable to rearrange the problem in such a way that, the order of convergence of the series be as high as possible, so we can apply Adomian method more efficiently. The Adomian A_n is found for large classes of nonlinearities or for a particular nonlinearity by a generation scheme just as one might develop Hermite, Lagrange, or Laguerre polynomials (Adomian & Rach, 1985). This work was specifically for quadratic, cubic, and general higher – order polynomial equations as well as negative, or non – integral powers, and random algebraic equations.

The Adomian polynomial A_n depending on $h_0, h_1, h_2, h_3, ..., h_n$ is given as

$$A_n(h_0, h_1, h_2, ..., h_n) = \frac{1}{n!}[\frac{d^n}{d\lambda^n} N \left[\sum_{n=1}^{\infty} h_n \lambda^n) \right]_{\lambda=0}, n = 0, 1, 2, 3, ...$$

Some few Adomian polynomials are given by

$$\begin{aligned}
A_0 &= N(h_0) \\
A_1 &= \frac{d}{d\lambda} N \left[\sum_{n=0}^{1} \lambda^{(n)} h_n \right]_{\lambda=0} \\
&= h_1 N'(h_0) \\
A_2 &= \frac{1}{2!} \frac{d^{(2)}}{d\lambda^{(2)}} N \left[\sum_{n=0}^{2} \lambda^{(n)} h_n) \right]_{\lambda=0} \\
&= \frac{1}{2!} \frac{d^{(2)}}{d\lambda^{(2)}} N \left[h_0 + \lambda h_1 + \lambda^2 h_2 \right]_{\lambda=0} \\
&= h_2 N'(h_0) + \frac{h_2}{2!} N''(h_0) \\
A_3 &= \frac{1}{3!} \frac{d^{(3)}}{d\lambda^{(3)}} N \left[\sum_{n=0}^{3} \lambda^{(n)} h_n) \right]_{\lambda=0} \\
&= \frac{1}{3!} \frac{d^{(3)}}{d\lambda^{(3)}} N \left[h_0 + \lambda h_1 + \lambda^2 h_2 + \lambda^3 h_3 \right]_{\lambda=0} \\
&= h_3 N'(h_1) + h_1 h_3 N''(h_0) + \frac{1}{3!} h_3 N'''(h_0)
\end{aligned}$$

Taylor's Series

Definition

If $f(x)$ has derivatives of all order at $x = a$ (that is $f^{(n)}(a)$ exist for $n = 0, 1, 2, 3, ...$) then the series

$$\sum_{n=0}^{\infty} \frac{f(n)(a)}{n!}(x-a)^n = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \frac{f(4)(a)}{4!}(x-a)^4 + ...$$

is called the Taylor series of f about c (Taylor series of f in the powers of $(x-a)$). If $a = 0$, the term Maclaurin series is usually used in place of Taylor series. For any smooth (that is continuously differentiable) function can be approximated as polynomial. The Taylor series provides a means to express this idea mathematically. Taylor series of f about a or the Taylor series of f in powers of $(x-a)$. Taylor series is a general method for writing a power series representation for function. Therefore the Taylor series is a power series.

Theorem 3. *If the nth – order derivative $f^{(n)}(t)$ for all t in the interval containing x and a and if $P_n(x)$ is the nth - order Taylor polynomial for f about a, then,*

$P_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}}{3!}(x-a)^3 +, ..., + \frac{f^{(n)}}{n!}(x-a)^n$

$f(x) \approx P_n(x)$

$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}}{3!}(x-a)^3 +, ..., + \frac{f^{(n)}}{n!}(x-a)^n$

Proof. Given a power series in $(x-a)$ or a power series centered at a or a power series about a

$$f(x) = \sum_{n=0}^{\infty} c_n(x-a)^n \tag{3.1}$$

$$f(x) = c_0 + c_1(x-a) + c_2(x-a)^2 + c_3(x-a)^3 + c_4(x-a)^4 + ... \qquad (3.2)$$

To find the coefficient c_n, put $x = a$ into equation (3.3)

$$\Rightarrow f(a) = c_0$$

Differentiating equation (3.3) gives

$$f'(x) = c_1 + 2c_2(x-a) + 3c_3(x-a)^2 + 4c_4(x-a)^3 + 5c_5(x-a)^4 + ... \qquad (3.3)$$

Substituting $x = a$ into equation (3.4), we have

$$f'(a) = c_1$$

Differentiating equation (3.4) we obtained

$$f''(x) = 2!c_2 + 2.3c_3(x-a) + 3.4c_4(x-a)^2 + 4.5c_5(x-a)^3 + ... \qquad (3.4)$$

Putting $x = a$ into equation (3.5) gives

$$f''(a) = 2!c_2 \Rightarrow c_2 = \frac{f''(a)}{2!}$$

Differentiating equation (3.5) we get

$$f^{(3)}(x) = 2.3c_3 + 2.3.4c_4(x-a) + 3.4.5c_5(x-a)^2 + ... \qquad (3.5)$$

Substituting $x = a$ into equation (3.6) yields

$$f^{(3)}(a) = 2.3c_3 \Rightarrow c_3 = \frac{f^{(3)}(a)}{3!}$$

Differentiating equation (3.6) and substituting $x = a$ into the differential equation yields

$$f^{(4)}(a) = 2.3.4c_4 \Rightarrow c_4 = \frac{f^{(4)}(a)}{4!}$$

It can be observed from the above results that

$$f^{(n)}(a) = 2.3.4.5...nc_n$$

that is

$$f^{(n)}(a) = n!c_n \Rightarrow c_n = \frac{f^{(n)}(a)}{n!} \tag{3.6}$$

Now substituting equation (3.6) into equation (3.2), we obtained

$$f(x) = \sum_{n=1}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

which can be expanded as

$$f(x) = f(a) + f'(a)(x-a) + \frac{f^{(2)}(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \frac{f^{(4)}(a)}{4!}(x-a)^4 + ...$$

$\square$

Now the nth degree Taylor polynomial of $f(x)$ is defined by

$$P_n(x) = \sum_{i=0}^{n} \frac{f^{(i)}(a)}{i!}(x-a)^i$$

This nth degree of the Taylor series is just the partial sum of the series. Therefore the Taylor series can be written as

$$f(x) = P_n(x) + E_n(x)$$

Where the Lagrange remainder or the error term $E_n(x)$ is given by

$$E_n(x) = \frac{f^{(n+1)}(a)}{(n+1)!}(x-a)^{n+1}$$

The Lagrange remainder is naturally the error obtained from the partial sum $P_n(x)$ and the full Taylor series $f(x)$.

Corollary 1: Taylor Theorem for $f(x+h)$

If a function f possess continuous derivatives of order $0, 1, 2, 3, ..., (n+1)$ in a close interval $I = [a, b]$ then for any x in I

$$f(x+h) = \sum_{n=0}^{N} \frac{f^{(n)}(a)}{n!}(h)^i + \frac{f^{(n+1)}(z)}{(n+1)!}h^{n+1}$$

where h is any value such that $x+h$ is in I and where

$$E_{n+1} = \frac{f^{(n+1)}(z)}{(n+1)!}h^{n+1}$$

for some z between x and $x+h$

Proof. $F(a) = f(a) + f'(a)(x-a) + \frac{f^{(2)}a}{2!}(x-a)^2 + \frac{f^{(3)}a}{3!}(x-a)^3$
$+ \frac{f^{(4)}a}{4!}(x-a)^4 + ... + \frac{f^{(n)}a}{n!}x - a^n + B(x-a)^{n+1}$
$F'(a) = f'(a) + \Big[f'(a)(x-a)^0(-1) + f^{(2)}(a)(x-a) + \frac{f^{(2)}(a)}{2!}2(x-a)(-1)\Big] +$

$\Big[\frac{f^{(3)}(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}3(x-a)^2(-1) + \frac{f^{(4)}(a)}{3!}(x-a)^3 + \frac{f^{(4)}(a)}{4!}4(x-a)^3(-1)\Big]$
$+ ... + \Big[\frac{f^{(n)}(a)}{(n-1)!}(x-a)^n(-1) + \frac{f^{(n)}(a)}{n!}n(x-a)^n(-1)\Big]$
$+ \frac{f^{(n+1)}(a)}{n!}(x-a)^n + B(n+1)(x-a)^n(-1)$
$F'(a) = f'(a) - f'(a) + f^{(2)}(a)(x-a) - f^{(2)}(a)(x-a) + \frac{f^{(3)}(a)}{2}(x-a)^2$
$- \frac{f^{(3)}(a)}{2}(x-a)^2 + \frac{f^{(4)}(a)}{3!}(x-a)^3 - \frac{f^{(4)}(a)}{3!}(x-a)^3 + ... + \frac{f^{(n)}(a)}{(n-1)!}(x-a)^{n-1} -$
$\frac{f^{(n)}(a)}{(n-1)!}(x-a)^{n-1} + \frac{f^{(n+1)}(a)}{n!}(x-a)^n + B(n+1)(x-a)^n(-1)$
$\frac{f^{(n+1)}(z)}{n!}(x-a)^n - B(n+1)(x-z)^n = 0$

$$B(n+1)(x-z)^n = \frac{f^{(n+1)}(z)}{n!}(x-z)^n$$

$B = \frac{f^{(n+1)}(z)}{n!(n+1)(x-z)^n}(x-z)^n$

$B = \frac{f^{(n+1)}(z)}{(n+1)!}$

$F(a) = \sum_{n=0}^{N} \frac{f^{(n)}(a)}{n!}(x-a)^n + \frac{f^{(n+1)}(z)}{(n+1)!}(x-a)^{n+1}$

$f(x+h) = \sum_{n=0}^{N} \frac{f^{(n)}(a)}{n!}(x+h-x)^n + \frac{f^{(n+1)}(z)}{(n+1)!}(x+h-x)^{n+1}$

$f(x+h) = \sum_{n=0}^{N} \frac{f^{(n)}(a)}{n!}(h)^n + \frac{f^{(n+1)}(z)}{(n+1)!}(h)^{n+1}$

□

Newton - Raphson's Method

The Newton's method was developed by Sir Isaac Newton in 1669. Newton applied his method to polynomials and computed a sequence of polynomials. In the year 1690, Joseph Raphson modified the method. Therefore this method is now known as Newton - Raphson's method. The Newton's method has been the foundation for almost all other numerical methods for solving nonlinear equations. It is a derivative based method. Newton – Raphson's method applicability extends to system of nonlinear equations, differential equations and integral equations. In this work, it is being applied to a single variable nonlinear equation of the form $f(x) = 0$. The following conditions are very important in using the Newton's method.

1) $f(x)$ is continuous on the closed interval $[a, b]$ which contains the root.

2) $f'(x)$ and $f''(x)$ are also continuous on the closed interval $[a, b]$.

3) $f(a).f(b) < 0$.

4) $f'(x)$ and $f''(x)$ keep the same sign on $[a, b]$.

The tangent line to the curve $y = f(x)$ at $x = x_0$ has the equation

$$y = f(x_0) + f'(x_0)(x - x_0).$$

Given that the points $(x_1, 0)$ and $(x_2, 0)$ lie on the line f. For the point $(x_1, 0)$

$$0 = f(x_0) + f'(x_0)(x_1 - x_0)$$
$$x_1 - x_0 = -\frac{f(x_0)}{f'(x_0)}$$
$$\Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Also the point $(x_2, 0)$ lies on the tangent line so

$$0 = f(x_1) + f'(x_1)(x_2 - x_1)$$
$$x_2 - x_1 = -\frac{f(x_1)}{f'(x_1)}$$
$$\Rightarrow x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

If Newton's method is described in terms of a sequence $x_0, x_1, x_2, ...$ then in general, the Newton's method is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{3.7}$$

Newton's Algorithms

Step 1 : Choose x_0 as the estimate of $f(x) = 0$.

Step 2 : Repeat for $n = 0, 1, 2, 3, ...$

Step 3 : Set $x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$.

Step 4 : Stop when the absolute value of the derivative of the function of the new iterate is sufficiently small, that is $|f'(x_{n+1})| \leq \varepsilon$.

Chapter Summary

This chapter starts with mathematical preliminaries which essentially focuses on the concept of convergence analysis. Some forms of convergence namely absolute convergence, conditional convergence and quadratic convergence are also discussed. This chapter also touches on order and rate of convergence. Adomian Decomposition Method (ADM) is one of the main methods used in the study. This chapter has the detail highlight on this method as well as the generation of some Adomian polynomials. Another method used in this study is Taylor series. In this chapter, we have proved the theorems for Taylor series for $f(x)$ as well as $f(x+h)$. The method constructed in this study is a modification of Newton's method. Therefore the prove of the Newton's method together with its algorithms are provided in this chapter. This chapter is basically about discussion on the concept of convergence and the methods used in this study.

CHAPTER FOUR

RESULTS AND DISCUSSION

Introduction

In this chapter, we present the theorem and proof of the proposed modified iterative method for computing the approximate solution of non-linear equations. The theorem for the convergence of the proposed modified iterative method is proved and we have show that this proposed method has quadratic convergence. Finally, we run up this chapter by discussing numerical examples to illustrate the efficiency of the new algorithm and compare the proposed method with other existing ones in literature.

Theorem 4. *Consider a single nonlinear algebraic or transcendental functions of the form $f(x) = 0$. If*

1 f, $f'(x)$,and $f''(x)$ are continuous and differentiable in the neighborhood of the root r of the function f on the closed interval $[a, b]$ and

2 $f''(x) \neq 0$ then, the iterative method

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

will converge to the root of the function quadratically

Proof of the Proposed Modified Method

We consider nonlinear problem

$$f(x) = 0, \tag{4.1}$$

where $f(x)$ is a function of a single real variable x. The Newton's method is applied to find the solution of equation (4.1) using both first and second derivatives. Let us now consider the function

$$p(x) = x - \frac{p(x)}{p'(x)}$$

where $p(x) = f'(x)$ and $f(x)$ is the function to be evaluated. Using Taylor's series expansion about a gives

$$p(x) = p(a) + p'(a)(x-a) + \frac{p^{(2)}(a)}{2!}(x-a)^2 + \frac{p^{(3)}(a)}{3!}(x-a)^3 + \frac{p^{(4)}(a)}{4!}(x-a)^4 + ... \tag{4.2}$$

Now writing $p(x+h)$ in Taylor series expansion about x, we obtain

$$p(x+h) = p(x) + p'(x)(x+h-x) + \frac{p^{(2)}(x)}{2!}(x+h-x)^2 + \frac{p^{(3)}(x)}{3!}(x+h-x)^3 + \frac{p^{(4)}(x)}{4!}(x+h-x)^4 + ... \tag{4.3}$$

$$= p(x) + hp'(x) + \frac{p^{(2)}(x)}{2!}h^2 + \frac{p^{(3)}(x)}{3!}h^3 + \frac{p^{(4)}(x)}{4!}h^4 + ...$$

$$Let \phi(h) = \frac{h^2p^{(2)})(x)}{2!} + \frac{h^3p^{(3)}(x)}{3!} + \frac{h^4p^{(4)}(x)}{4!} + \frac{h^5p^{(5)}(x)}{5!} + ... \tag{4.4}$$

$$\Rightarrow p(x+h) = p(x) + hp'(x) + \phi(h) \tag{4.5}$$

$$\Rightarrow \phi(h) = p(x+h) - p(x) + hp'(x) \tag{4.6}$$

Given that $p(x) = 0, \Rightarrow p(x+h) = 0$. We can search for a value of h provided $p'(x) \neq 0$. Since from (4.5) $p(x+h) = 0$, then

$$p(x) + hp'(x) + \phi(h) = 0, \tag{4.7}$$

$$hp'(x) = -p(x) - \phi(h)$$

$$\Rightarrow h = -\left[\frac{p(x) + \phi(h)}{p'(x)}\right]$$

$$\therefore h = -\frac{p(x)}{p'(x)} - \frac{\phi(h)}{p'(x)} \tag{4.8}$$

So h can be simplified as

$$h = c + N(h) \tag{4.9}$$

where

$$c = -\frac{p(x)}{p'(x)} \tag{4.10}$$

and

$$N(h) = -\frac{\phi(h)}{p'(x)} \tag{4.11}$$

But from (4.6), $\phi(h) = p(x+h) - p(x) + hp'(x)$

$$\Rightarrow N(h) = -\left[\frac{p(x+h) - p(x) + hp'(x)}{p'(x)}\right] \tag{4.12}$$

Now applying Adomian decomposition to (4.9) which can be written as

$$h - N(h) = c.$$

where c is a constant and $N(h)$ is a nonlinear function with h having the series form

$$h = \sum_{n=0}^{\infty} h_n. \tag{4.13}$$

The nonlinear function is decomposed as

$$N(h) = N(\sum_{n=0}^{\infty} h_n) = \sum_{n=0}^{\infty} A_n, \tag{4.14}$$

where An is the Adomian polynomial depending on $h_0, h_1, h_2, h_3, ..., h_n$. Thus,

$$A_n(h_0, h_1, h_2, h_3, ..., h_n) = \frac{1}{n!}\left[\frac{d^{(n)}}{d\lambda^n} N(\sum_{n=0}^{\infty} \lambda^{(n)} h_n)\right]_{\lambda=0}, n = 0, 1, 2, 3, ...$$

Some few polynomials are given by

$$
\begin{aligned}
A_0 &= N(h_0)\\
A_1 &= \frac{d}{d\lambda}N\left[\sum_{n=0}^{1}\lambda^{(n)}h_n)\right]_{\lambda=0}\\
&= h_1N'(h_0)\\
A_2 &= \frac{1}{2!}\frac{d^{(2)}}{d\lambda^{(2)}}N\left[\sum_{n=0}^{2}\lambda^{(n)}h_n\right]_{\lambda=0}\\
&= \frac{1}{2!}\frac{d^{(2)}}{d\lambda^{(2)}}N\left[h_0+\lambda h_1+\lambda^2h_2\right]_{\lambda=0}\\
&= h_2N'(h_0)+\frac{h_2}{2!}N''(h_0)\\
A_3 &= \frac{1}{3!}\frac{d^{(3)}}{d\lambda^{(3)}}N\left[\sum_{n=0}^{3}\lambda^{(n)}h_n)\right]_{\lambda=0}\\
&= \frac{1}{3!}\frac{d^{(3)}}{d\lambda^{(3)}}N\left[h_0+\lambda h_1+\lambda^2h_2+\lambda^3h_3\right]_{\lambda=0}\\
&= h_3N'(h_1)+h_1h_3N''(h_0)+\frac{1}{3!}h_3N'''(h_0)\\
A_4 &= \frac{1}{4!}\frac{d^{(4)}}{d\lambda^{(4)}}N\left[\sum_{n=0}^{4}\lambda^{(n)}h_n)\right]_{\lambda=0}\\
&= \frac{1}{4!}\frac{d^{(4)}}{d\lambda^{(4)}}N\left[h_0+\lambda h_1+\lambda^2h_2+\lambda^3h_3+\lambda^4h_4\right]_{\lambda=0}\\
&= h_4N'(h_0)+(h_1h_3+\frac{1}{2}h_2)N^{(2)}(h_0)+\frac{1}{2}h_1h_2N^{(3)}(h_0)+\frac{1}{4}h_4N^{(4)}h_0)
\end{aligned}
$$

Substituting (4.13)and (4.14) into (4.9) gives

$$\sum_{n=0}^{\infty}h_n = c+\sum_{n=0}^{\infty}A_n \tag{4.15}$$

It follows from (4.15) that $h_0 = c$ and $h_{n+1} = c+A_n$. From (4.9), $c = -\frac{p(x)}{p'(x)}$ and since $h_0 = c$ we have that

$$h_0 = -\frac{p(x)}{p'(x)}. \tag{4.16}$$

Adding x to both sides of equation (4.16) yields

$$x + h_0 = x - \frac{p(x)}{p'(x)}. \tag{4.17}$$

Given that $y = (x + h_0)$, then equation (4.17) becomes

$$y = x - \frac{p(x)}{p'(x)}$$

which gives an iterative form

$$y_n = x_n - \frac{p(x_n)}{p'(x_n)}. \tag{4.18}$$

From the generation of few Adomian polynomial, $A_0 = N(h_0)$. From (4.12), we obtain

$$A_0 = N(h_0) = -\frac{[p(x + h_0) - p(x) - h_0 p'(x)]}{P'(x)} \tag{4.19}$$

Substituting $y = (x + h_0)$ and $h_0 = -\frac{p(x)}{p'(x)}$ into (4.19) gives

$$A_0 = -\frac{[p(y) - p(x) + \frac{p(x)}{p'(x)} p'(x)]}{P'(x)}$$

$$\Rightarrow A_0 = -\frac{p(y)}{p'(x)}. \tag{4.20}$$

From $h_0 = c$ and $h_{n+1} = c + A_n$. If $n = 0$ then $h_1 = c + A_0 \Rightarrow h_1 = h_0 + A_0$ So h_1 can be expressed as

$$h_1 = h_0 - \frac{p(y)}{p'(x)} \tag{4.21}$$

Substituting (4.16) into (4.21) gives

$$h_1 = -\frac{p(x)}{p'(x)} - \frac{p(y)}{p'(x)} \tag{4.22}$$

$$\Rightarrow x + h_1 = x - \frac{p(x)}{p'(x)} - \frac{p(y)}{p'(x)}$$

$$\Rightarrow x + h_1 = x_n - \frac{p(x_n)}{p'(x_n)} - \frac{p(y_n)}{p'(x_n)} \tag{4.23}$$

Comparing (4.23) to the Newton's method, it is realised that, $x+h_1 = x_{n+1}$. This implies that

$$x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)} - \frac{p(y_n)}{p'(x_n)} \tag{4.24}$$

Substituting (4.18) into (4.24) we arrive at

$$x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)} - \frac{p\left[x_n - \frac{p(x_n)}{p'(x_n)}\right]}{p'(x_n)}. \tag{4.25}$$

Taking $p(x) = f'(x)$ and substituting into (4.25) gives

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} - \frac{f'\left[x_n - \frac{f'(x_n)}{f''(x_n)}\right]}{f''(x_n)} \tag{4.26}$$

Convergence Analysis of the Proposed Method

Theorem 5. *Let $\alpha \in I$, be a zero of sufficiently differentiable function $p : I \to R$ for an open interval I. If x_0 is sufficiently close to α, then*

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

has quadratic convergence

Proof. Since p is sufficiently differentiable, by expanding $p(x)$ and $p'(x)$ about α we have

$$p(x_n) = p'(\alpha)\left[+c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right] \tag{4.27}$$

$$p'(x_n) = p'(\alpha)\left[+2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)\right] \quad , \tag{4.28}$$

where $e_n = x_n - \alpha$, $c_k = \frac{p^{(k)}(\alpha)}{k!p'(\alpha)}, k = 2, 3, 4, ...$ Dividing equation (4.27) by (4.28), that is

$$\frac{p(x_n)}{p'(x_n)} = \frac{p'(\alpha)\left[+c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + O(e_n^5)\right]}{p'(\alpha)\left[+2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)\right]}$$

$$= e_n - c_2e_n^2 + 2(c_2^2 - c_3)e_n^3 + (7c_2c_3 - 3c_4 - 4c_2^3)e_n^4 + O(e_n^5) \tag{4.29}$$

Substituting $e_n = x_n - \alpha$, equation (4.29) becomes

$$x_n - \frac{p(x_n)}{p'(x_n)} = \alpha + c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 +$$
$$(3c_4 + 4c_2^3 - 7c_2c_3)e_n^4 + O(e_n^5) \tag{4.30}$$

From (4.18), we have $y_n = x_n - \frac{p(x_n)}{p'(x_n)}$. Expanding $p(y_n)$ about α (4.30) gives

$$p(y_n) = p'(\alpha)[c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + (3c_4 + 4c_2^3 - 7c_2c_3)e_n^4 + O(e_n^5)] \tag{4.31}$$

Dividing (4.31) by (4.28), we have

$$\frac{p(y_n)}{p'(x_n)} = \frac{p'(\alpha)\,[c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + (3c_4 + 4c_2^3 - 7c_2c_3)e_n^4 + O(e_n^5)]}{p'(\alpha)\,[+2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)]}$$

$$= c_2e_n^2 + 2(c_3 - 2c_2^2)e_n^3 + (13c_2^3 - 14c_2c_3 + 3c_4)e_n^4 + O(e_n^5) \tag{4.32}$$

Subtracting equation (4.32) from (4.30), we obtain

$$x_{n+1} = \alpha + c_2e_n^2 + 2(c_3 - c_2^2)e_n^3 + \ldots$$

which implies that the modified iterative method has quadratic convergence. $\square$

The Algorithm for the Modified Iterative Method

The modified iterative method constructed above is presented in the following algorithm.

Step 1 : choose x_0 as the estimate of the solution of $f(x)$

Step 2 : repeat for $n = 0, 1, 2, 3, 4, \ldots$

Step 3 : set $y_n = x_n - \frac{f'(x_n)}{f''(x_n)}$.

Step 4 : set $x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} - \frac{f'\left[x_n - \frac{f'(x_n)}{f''(x_n)}\right]}{f''(x_n)}$.

Step 5 : stop when the absolute value of the first derivative of the function at the new point is sufficiently small, that is $|f'(x_{n+1})| \leq \varepsilon$

Numerical Examples

The following nonlinear problems were solved by using four different methods namely: Newton's method, External Touch Algorithm, Karthikeyan's Method and the Proposed Modified Iterative Method using mathlab R2020a.

1) $f_1(x) = 2^x + 3x - 2, \quad x_0 = 1.5$

2) $f_2(x) = xcosx + 2x^2 + 3x - 2, \quad x_0 = 0.5$

3) $f_3(x) = x^2 - 2 - exp(-x), \quad x_0 = 2$

4) $f_4(x) = x^2 - (1-x)^5, \quad x_0 = 0.2$

5) $f_5(x) = x^5 - 4x^4 - 7x^3 + 14x^2 - 44x + 120, \quad x_0 = 1.5$

Table of Results for the Solved Equations

The following are the meaning of the initials as used in the table headings. NM = Newton - Raphson's Method (1669), ETA = External Touch Algorithm by Kanwar et.al (2003), KM = Karthikeyan's method (2010) and last initial PMIM is the Proposed Modified Iterative Method which we have proposed in this study.

Table 1: **Computational results for** $f_1(x) = 2^x - x^2 + 3x - 2, \quad x_0 = 1.5$

Iterations	NM	ETA	KM	PMIM
0	1	0.536333	0.852857	0.646055
1	1.577681	1.266454	1.601667	1.624554
2	1.493612	1.166399	1.494166	1.49472
3	1.491645	1.164039	1.491646	1.491646
4	1.491644	1.164038	1.491644	1.491644
5	1.491644	1.164038	0.345955	0.345955
6	1.491644	1.164038		
7	0.345955	-0.42583		

Table 2: **Computational results for** $f_2(x) = xcos(x) + 2x^2 + 3x - 2, \ \ x_0 = 0.5$

Iterations	NM	ETA	KM	PMIM
0	0.5	-0.27202	0.186446	-0.55266
1	1.966844	1.713734	2.194971	2.093663
2	1.542506	1.224772	1.556745	1.57059
3	1.492343	1.164876	1.49254	1.492737
4	1.491644	1.164038	1.491644	1.491644
5	1.491644	1.164038	0.345955	0.345955
6	1.491644	1.164038		
7	0.345955	-0.42583		

Table 3: **Computational results for** $f_3(x) = x^2 - 2 - exp(-x), \ \ x_0 = 2$

Iterations	NM	ETA	KM	PMIM
0	2	1.750898	2.135206	2.241818
1	1.54909	1.232593	1.580729	1.56516
2	1.492533	1.165104	1.492783	1.493034
3	1.491644	1.164038	1.491644	1.491645
4	1.491644	1.164038	0.345945	0.345943
5	1.491644	1.164038	0.345955	0.345955
6	0.345955	-0.42583		
7	0.345955	-0.42583		

Discussions on the Numerical Results

The following observations can be deduced from the tables presented above. From Table 1, the original Newton's method and the External Touch Algorithm converge at 7th iteration which is the last iteration while the Karthikayen's method and the Proposed Modified Iterative method converge at the 5th iteration. The results from this table clearly showed that, there is a more improved outcome at every stage of the iteration in the Karthikeyan's method (2010) and the Proposed Modified Iterative method. However, the Proposed Iterative method approaches the approximate root rapidly than the Karthikeyan's method. This implies that, the Proposed Modified Iterative method performs better than the Newton's method, External Touch Algorithms and the Karthikeyan's method (2010).

Table 4: **Computational results for** $f_4(x) = 1 - (1-x)^5, \quad x_0 = 0.2$

Iterations	NM	ETA	KM	PMIM
0	1.5	1.1837	1.57902	1.65304
1	0.811321	0.054912	1.027831	0.930396
2	0.407385	-0.5167	0.392617	0.377382
3	0.342506	-0.42046	0.342972	0.343438
4	0.345941	-0.42581	0.345945	0.345943
5	0.345955	-0.42583	0.345955	0.345955
6	0.345955	-0.42583		
7	0.345955	-0.42583		

Table 5: **Computational results for**
$f_5(x) = x^5 - 4x^4 - 7x^3 + 14x^2 - 44x + 120, \quad x_0 = 1.5$

Iterations	NM	ETA	KM	PMIM
0	1.5	1.512832	1.504203	1.508364
1	2.131516	2.139122	2.130784	2.130051
2	2.005712	2.014011	2.00568	2.005647
3	2.000012	2.008345	2.000012	2.000012
4	2	2.008334	2	2
5	2	2.008334		
6	2	2.008334		

In Table 2, the Karthikeyan's method and the Proposed Modified Iterative method converge at the 5th iteration. The Newton's method converges at the 7th iteration, the External Touch Algorithms method also converges at the 7th iteration just like the Newton's method but the convergence value is different form all the other methods. This may be probably due to the nature of the iteration scheme constructed by Kanwar et.al (2003). In spite of this, the External Touch Algorithm reults in this case is an improvement over that of Newton's method.

Also, a critical look at table 2 shows that, the Proposed Modified Iterative Method outperformed the Karthikeyan's method. It can be inferred from table 3 that, the Proposed Modified Iterative method and the Karthikeyan's method have their convergence at the 5th iteration as compared to the External Touch Algorithm and the Newton's method which converge at the 7th iteration. The results from table 3 indicates that , apart from the 3rd iteration, the Proposed Modified Iterative Method perform better than the Karthikeyan's Method, the External Touch Algorithm and Newton's method.This is because the rapidity of its convergence is faster than the rest of the methods.

The results on the table 4 shows a similar trend as in table 3. From table 4, the Newton's method and the External Touch Algorithm converge at the 7th iteration but the value that the External touch Algorithm converged to is different from actual approximate value as seen from the table. Once again the Proposed Iteration Method has shown its supremacy over the Karthikeyan's method, External Algorithms Method and the well - known Newton's method though it converges at the 5th iteration just like the Karthikeyan's method (2010). In table 5, the original Newton's and the External Touch Algorithm converge at the 6th iteration. Also, the Karthikeyan's method (2010) and the Proposed Modified Iterative method, converge at the 4th iteration. A comparative analysis of the results from this table shows that, the Newton's method is better than the External Touch Algorithm.

The Karthikeyan's method (2010) is indeed an improvement over the Newton's method and the External Touch Algorithm. In the same vain, the results vividly showed that, there is a more improved results at every stage of iteration in the Proposed Modified Iterative method as compared to that of the Karthikeyan's method (2010). From the discussion so far, it is clear that the Proposed Modified Iterative method has a higher rate of convergence than the Newton's method, the External Touch Algorithm and the Karthikeyan's method (2010) though they all have order of convergence to be 2. Therefore the Proposed Modified Iterative method may be describe as an improvement over the Newton's method, the Kanwar et.al External Touch Algorithm method (2003) and the Karthikeyan's method (2010).

Chapter Summary

In this chapter, we made use of the Adomian Decomposition Method (ADM) and Taylor series to construct the Proposed Modified Iterative method. The theorem and proof of this modified iterative method is given in this chapter. The theorem for the convergence of the proposed method is given and we have proved that, this method converges quadratically. The algorithms for the proposed modified iterative method is presented. We have given numerical examples by solving some nonlinear equations using our proposed method and Matlab R2020a software. The results we obtained are discussed thoroughly by comparing the performance of our method with other existing methods used in this study which are the Newton's method, the Karthikeyan's method and the External Touch Algorithm method. Detail discussion is done on the performance of the various methods from the results of the numerical examples presented in the tables above and conclusion is drawn.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Overview

Numerical methods, and for that matter iterative methods have become indispensable aspect of mathematics in recent times as mathematicians in particular and researchers in general have resolved to solve complex problems that can not be solved analytically.This aspect of mathematics cuts across almost every field of mathematics. It also has vast applications in engineering, computer science and science in general. The Proposed Modified Iterative method is a two-step iterative method of which one serves as the predictor and the other as a corrector. Even though, our method is a two -step iterative method, it is different from those known in literature due to the way it was constructed and the nature of the iterative scheme. This method is proved to converge quadratically. In this chapter, we summarize and draw a conclusion about the work done in this thesis.

Summary

To compute the approximate solutions of nonlinear equations and any other equations require numerical methods which are iterative in nature. A modified iterative scheme for solving single variable nonlinear equations of the form $f(x)$=0 where, $f(x)$ is a nonlinear function is presented in this work. The proposed modified iterative algorithm was constructed using Taylor series, Newton-Raphson's Method and the Adomian Decomposition Method (ADM). The developed modified scheme serves as an improvement over Newton - Raphson's method, the External Touch Algorithm proposed by Kanwar et.al (2003), Karthikeyan's method (2010) for solving nonlinear equations. This method converges quadratically. Numerical solutions are presented in five different tables using five test functions. A comparison analysis was done using matlab R2020a software. It is evident that, our method is more efficient as compare to the well-known Newton-Raphson's method, the External Touch Algorithm method and the Karthikeyan's method (2010).

Conclusion

Nonlinear problems are one of the most solved problems Numerical analysis and are often solved iteratively to obtain approximate solutions. In this thesis we have been able to construct modified iterative method for solving nonlinear equations based on the Newton's method and have prove that this method converges quadratically. By considering the results obtained from the numerical examples, it implies that, the constructed method produces better results than the other ones.

Recommendations

In this thesis we considered Taylor series expansion of the function $f(x + h)$ to the third term ignoring all the other terms and obtained quadratic convergence. We therefore suggest a further research whereby one will take the Taylor series expansion of the function $f(x+h)$ to a step further by truncating the expansion at the fourth or five term in order to obtain higher convergence for better approximations. Also, the constructed algorithm in this thesis is meant to solve nonlinear equations in single variable. Therefore we recommend a further research of developing an iterative scheme based on the Newton's method that can solve system of nonlinear equations.

REFERENCES

Abbaoui, K., & Cherruault, Y. (1994). Convergence of adomian's method applied to differential equations. *Computers & Mathematics with Applications*, *28*(5), 103–109.

Abramowitz, M., & Stegun, I. (1972). Solutions of quartic equations. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, 17–18.

Adams, R. A., & Essex, C. (2010). *calculus: a complete course: 7th edition* (Vol. 4). Addison-Wesley Boston.

Adomian, G., & Rach, R. (1985). On the solution of algebraic equations by the decomposition method. *Journal of Mathematical Analysis and Applications*, *105*(1), 141–166.

Ahmad, F., Hussain, S., & Raza, M. (2012). New derivative free iterative method for solving non-linear equations. *Academic Research International*, *2*(1).

Akinsunmade, A. E. (2016). A new iterative newton's method for nonlinear optimization problems. *ResearchGate*, *3*(2).

Andrew, D., & Booth, D. (1966). *Numerical methods: 3rd edition*. Butterworth & Co. Ltd.

Anton, H., Bivens, I. C., & Davis, S. (2012). *Calculus early transcedentals: 10th edition*. John Wiley & Sons.

Babolian, E., & Biazar, J. (2002). On the order of convergence of adomian method. *Applied Mathematics and Computation*, *130*(2-3), 383–387.

Basto, M., Semiao, V., & Calheiros, F. L. (2006). A new iterative method to compute nonlinear equations. *Applied Mathematics and Computation*, *173*(1), 468–483.

Basto, M., Semiao, V., & Calheiros, F. L. (2007). Numerical study of modified adomian's method applied to burgers equation. *Journal of Computational and Applied Mathematics*, *206*(2), 927–949.

Brent, R. P. (2013). *Algorithms for minimization without derivatives.* Courier Corporation.

Burden, R., & Faires, J. (2011). *Numerical analysis 9th edition.* Boston: Brooks/Cole.

Chun, C. (2005). Iterative methods improving newton's method by the decomposition method. *Computers & Mathematics with Applications*, *50*(10-12), 1559–1568.

Cordero, A., Hueso, J. L., Martínez, E., & Torregrosa, J. R. (2012). Steffensen type methods for solving nonlinear equations. *Journal of Computational and Applied Mathematics*, *236*(12), 3058–3064.

Feng, J. (2009). A new two-step method for solving nonlinear equations. *International Journal of Nonlinear Science*, *8*(1), 40–44.

Homeier, H. (2005). On newton-type methods with cubic convergence. *Journal of Computational and Applied Mathematics*, *176*(2), 425–432.

Jinqing, F. (1993). Inverse operator theory method and its applications in nonlinear physics [j]. *Progress In Physics*, *4*.

Kanwar, V., Singh, S., Sharma, J., & Mamta. (2003). New numerical techniques for solving non-linear equations. *Indian Journal of Pure & Applied Mathematics*, *34*(9), 1339–1349.

Karthikeyan, K. (2010). Efficient algorithm for minimization of non-linear function. *Applied Mathematical Sciences*, *4*(69), 3437–3446.

Kou, J., Li, Y., & Wang, X. (2007). Fourth-order iterative methods free from second derivative. *Applied Mathematics and Computation*, *184*(2), 880–885.

Ledder, G. (2005). *Differential equations: A modeling approach: 1st edition.* McGraw-Hill Higher Education.

Li, Y., Wei, Y., & Chu, Y. (2015). Research on solving system of nonlinear equations based on improved pso. *Mathematical Problems in Engineering*, *2015*.

Mitlif, R. J. (2014). New iterative method for solving nonlinear equations. *Baghdad Science Journal*, *11*(4), 1649–1654.

Petrovic, J. S. (2013). *Advanced calculus: theory and practice.* CRC Press.

Rach, R. (1984). A convenient computational form for the adomian polynomials. *Journal of Mathematical Analysis and Applications*, *102*(2), 415–419.

Rudall, B. H., & Rach, R. C. (2008). A new definition of the adomian polynomials. *Kybernetes*.

Stewart, J. (2008). *Calculus: Early transcendentals, volume 6e.* Brooks/-Cole, Belmont, California.

Ujević, N. (2006). A method for solving nonlinear equations. *Applied Mathematics and Computation*, *174*(2), 1416–1426.

Ward Cheney, E., & Ronald Kincaid, D. (2008). *Numerical mathematics and computing: 6th edition.* Belmont: Thomson Corporation.

Wu, X., & Fu, D. (2001). New high-order convergence iteration methods without employing derivatives for solving nonlinear equations. *Computers & Mathematics with Applications*, *41*(3-4), 489–495.

Yang, Y. (1994). Convergence of the adomian method and a new algorithm for the adomian polynomials. *Private Communication*.

Yasmin, N., & Junjua, M. (2012). Some derivative free iterative methods for solving nonlinear equations. *Academic Research International*, *2*(1), 75–82.

Yun, B. I. (2008). A non-iterative method for solving non-linear equations. *Applied Mathematics and Computation*, *198*(2), 691–699.

Yun, B. I., & Petkovic, M. S. (2011). A quadratically convergent iterative method for nonlinear equations. *J. Korean Math. Soc*, *48*(3), 487–497.

Zavalaus, G. (2014). A modification of newton method with third-order convergence. *American Journal of Numerical Analysis*, *2*(4), 98–101.

APPENDICE

APPENDIX A: PROPOSED METHOD MATLAB PSEUDOCODE

Below is the Matlab codes used in this thesis to obtained the results displayed on the various tables.

The x_0 value is the initial guess of each function which is used to obtained the actual approximate solution of the functions.

syms x

myfun $= 2^x + 3x - 2;\ \ x_0 = 1.5;$

%myfun $= xcosx + 2x^2 + 3x - 2;\ \ x_0 = 0.5;$

%myFun $= x^2 - 2 - exp(-x);\ \ x_0 = 2;$

%myFun $= x^2 - (1-x)^5; x_0 = 0.2;$

%myFun $= x^5 - 4x^4 - 7x^3 + 14x^2 - 44x + 120,\ \ x_0 = 1.5$

f = symfun(myfun,x);

$f1$ = symfun(diff(f),x);

$f2$ = symfun(diff($f1$),x);

%$f = @(x)exp(x) - 3 * x;$

%$fp = @(x)$ diff(f);

$x_0 = 1.5;$

$N = 100;$

tol $= 1E - 10;$

$x(1) = x_0$; % Set initial guess $n = 2$;

nfinal $= N + 1;$

while $(n <= N + 1)$

$fe = f(x(n-1));$

$fpe = f1(x(n-1));$

$fp2 = f1(x(n-1));$

$x(n) = x(n-1) - fe/fpe;$

if (abs(fe) <= tol)

nfinal $= n$;

break;

end

$n = n + 1$;

end

$y =$ 0:nfinal-1;

$y1 = x$(1:nfinal);

%$y1 =$ double(abs($y1$));

$y9 =$ diff($y1, 1$);

$y10 =$ diff($y1, 2$);

%$f3 = y9$(1:end-1);

$f2 =$ subs(($f2$));

$f2 =$ double($f2$);

$f4 = y10$(1:end);

$yK = y1(1\text{:end}-2) - f3./(2 * f2(1\text{:end}-2) - f4)$

$yP = y1(1\text{:end}-2) - f3./f2(1\text{:end}-2)$

$f1 =$ double(subs($f1$));

$f =$ double((1));

$x =$ double((x));

ETA $= y1 - 2. * 1./(f1 + sqrt(1 - f1.^2))$;

ETA = real(ETA);

$A =$[yK' yP'] % $yK = Kathi$, yP is the proposed

$B =$ [$y1'$ ETA'] % $y1 =$ Newton , ETA

$xlswrite('My_results', A)$

$xlswrite('My_results1', B)$