Eighth order Predictor-Corrector Method To Solve Quadratic Riccati Differential Equations

A Thesis Submitted to the Department of Mathematics in Partial Fulfillment for the Requirements of the Degree of Masters of Science in Mathematics

Numerical Analysis

Prepared by:
Wase Kasahun
Under the supervision of:
Alemayehu Shiferaw (Ph.D.)
Solomon Gebregiorgis (M.Sc.)

Jimma, Ethiopia
February, 2020

Declaration

This thesis is original work and has not been presented for a Degree in any other Universities.

Thesis submitted by: Wase Kasahun

Signature : __________________ date: __________________

Advisor: Alemayehu Shiferaw (Ph.D.)

Signature : __________________ date: __________________

Co-Advisor: Solomon Gebregiorgis (M.Sc.)

Signature: __________________ date: __________________

Abstract

In this thesis, eighth order predictor-corrector method is presented for solving quadratic Riccati differential equations. The solution domain is discretized and the stability and convergence of the method have been investigated. To validate the applicability of the proposed method, five model examples with exact solutions have been considered and numerically solved by using MATLAB software. The numerical results are presented in tables and figures for different values of mesh size h. Maximum absolute errors are also presented. Concisely, the present method gives better result than existing numerical methods reported in the literature. .

Acknowledgment

First of all, I am indebted to my almighty God who gave me long life, strength and helped me to reach this time. Next, I would like to express my deepest gratitude to my advisor Alemayehu Shiferaw (Ph.D.) and my co-advisor Solomon Gebregiorgis (M.Sc.) for their unreserved support, unlimited advice, constructive comments and immediate responses that helped me in developing this research.

Contents

List of Tables page

List of Figures page

Chapter 1

Introduction

1.1 Background of the study

Numerical analysis plays a significant role and helps us to find an approximate solution for problems which are difficult to solve analytically. In the field of computational mathematics, numerical methods are most widely utilized to solve equations arising in the field of physics, engineering and other sciences. The design and computation of the numerical algorithm is one of the mathematical challenges that researchers are facing to, but Scientists in the field of computational mathematics are trying to develop numerical methods by using computers for further application (Burden and Faires, 2011).

The attempt to solve physical problems led gradually to mathematical models involving an equation in which a function and its derivatives play important roles. However, the theoretical development of this new branch of mathematics - Ordinary Differential Equations has its origins rooted in a small number of mathematical problems. These problems and their solutions led to an independent discipline with the solution of such equations an end in itself.

The Riccati equation named after the Italian mathematician Jacopo Francesco Riccati (O'Connor and Robertson, 1996), is a basic first-order nonlinear ordinary differential equation. It has the form

$$\frac{dy}{dx} = p(x) + q(x)y + r(x)y^2 \tag{1.1}$$

which can be considered as the derivative of a function in terms of the function itself. It is assumed that $y(x)$, $p(x)$, $q(x)$ and $r(x)$ are real functions of the real argument x . It is well known that solutions to the general Riccati equation are not available and only special cases can be treated (Ince 1956). Even though the equation is nonlinear, similar to the second order inhomogeneous linear ordinary differential equations one needs only a particular solution to find the general solution (Anas *et.al*. 2010).

Riccati equation naturally arises in many fields of quantum mechanics; in particular, in quantum chemistry, the Wentzel-Kramers-Brillouin approximation and supersymmetry theories. Recently, methods for solving the Gross-Pitaevskii equation arising in Bose-Einstein condensates based on Riccati equation were introduced (Anas *et.al*. 2010). The book of Reid (1972) contains the fundamental theories of Riccati equation, with applications to random processes, optimal control, and diffusion problems. Important in engineering and science applications that today are known as the classical proved, such as stochastic realization theory, optimal control, robust stabilization, and network synthesis, the newer applications include such areas as financial mathematics (Biazar and Islami, 2010). Nonlinear differential equations are essential tools for modeling many physical situations, for instance, spring mass systems, resistor-capacitor-induction circuits, bending of beams, chemical reaction, pendulums, the motion of rotating mass around body and so on.

Due to the nonlinear structure of the Riccati equation, the general solution of the Riccati equation cannot be easily found. Therefore, one has to use numerical techniques or approximate method for obtaining its solutions. Recently, Gemechis File and Tesfaye Aga (2016) used fourth order Runge-Kutta method for solving quadratic Riccati differential equations. Vinod and Dimple (2016) presented Newton-Raphson based modified Laplace Adomian decomposition method for solving quadratic

Riccati differential equations. Gemadi $et.al.$ (2017) presented fifth order predictor corrector method for solving quadratic Riccati differential equation. The multi-stage variational iteration method is applied as a new efficient method for solving quadratic Riccati differential equation by Batiha (2015). Tan and Abbasbandy (2008) employed the analytic technique called Homotopy Analysis Method (HAM) to solve a quadratic Riccati equation. The solution of Riccati equation with variable co-efcient by differential transformation method is presented by Mukherje and Roy (2012). Very recently, Fateme and Esmaile (2017) presented approximate solution for quadratic Riccati differential equations by Bezier curves method.

There are many continuous attempts to get a method that yields more accurate results. The purpose of this study is to formulate a more accurate and stable method for solving quadratic Riccati differential equation than some existing methods in the literature.

1.2 Objectives of the study

1.2.1 General objective

The general objective of this study is to develop eighth order predictor-corrector method for solving quadratic Riccati differential equation.

1.2.2 Specific objectives

The specific objectives of the present study are:

- To formulate eighth order Adams-Bashforth-Moulton predictor-corrector method for solving quadratic Riccati differential equation.
- To establish the stability and convergence of the proposed method.
- To compare the accuracy of the present method with some existing method in the literature.

1.3 Significance of the study

The outcomes of this study may have the following importance:

- Provide some background information for other researchers who work on this area.
- Help the graduate students to acquire research skills and scientific procedures.

1.4 Delimitation of the Study

This study is delimited to eighth order predictor-corrector method for solving quadratic Riccati differential equation of the form:

$$\frac{dy}{dx} = p(x) + q(x)y + r(x)y^2, \qquad y(x_0) = \alpha,\ \ x_0 \leq x \leq x_f$$

where $p(x),\ q(x)$ and $r(x)$ are continuous with $r(x) \neq 0$ and $x_0,\ x_f,\ \alpha$ are arbitrary constants for $y(x)$, which is unknown function.

Chapter 2

Literature Review

2.1 Predictor Corrector Methods

In numerical analysis, predictor-corrector methods belong to a class of algorithms designed to integrate ordinary differential equations to find an unknown function that satisfies a given differential equation. All such algorithms proceed in two steps. First the initial, “prediction” step, starts from a function fitted to the function-values and derivative-values at a preceding set of points to extrapolate this function’s value at a subsequent, new point. Then next, ”corrector” step refines the initial approximation by using the predicted value of the function and another method to interpolate that unknown function’s value at the same subsequent point (Bucher, 2003).

The methods of Euler, Heun, Taylor and Runge-Kutta are called single-step methods because they use only the information from one previous point to compute the values at the successive points, that is, only the initial point (x_0, y_0) is used to compute (x_1, y_1) and in general y_k is needed to compute y_{k+1} . After several points have been found it is feasible to use several prior points in the calculation. The eighth order predictor corrector method uses y_{k-7}, y_{k-6}, y_{k-5}, y_{k-4}, y_{k-3}, y_{k-2}, y_{k-1} and y_k in the calculation of y_{k+1} . This method is not self-starting; eight initial points, (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_4), (x_5, y_5), (x_6, y_6) and (x_7, y_7) must be given in advance in order to generate the points (x_k, y_k) for $k \geq 8$ (Johan and Kurtis, 2004)and we apply the eighth order Runge-Kutta method using the first seven nodal points.

A desirable feature of a multistep method is that the local truncation error (LTE) can be determined and a correction term can be included, which improves the accuracy of the answer at each step. Also, it is possible to determine if the step size is small enough to obtain an accurate value for (y_{k+1}) , yet large enough so that unnecessary and time-consuming calculations are eliminated. If the code for the subroutine is fine-tuned, then the combination of a predictor and corrector requires only two function evaluations of $f(x,y)$ per step (Johan and Kurtis, 2004).

2.2 Higher order Runge-Kutta method

One of the most celebrated methods for the numerical solution of differential equations is the one originated by Runge and elaborated by Heun, Kutta, Nystrom, and others. This method is usually given considerable prominence in texts where numerical methods are discussed. In contrast to step-by-step procedures based on formulas for numerical quadrature the Runge-Kutta method (as it is usually called) enjoys two conspicuous advantage :

1. No special devices are required for starting the computation.
2. The length of the step can be modified at any time in the course of the computation without additional labor.

On the other hand it is open to two major objections:

1. The process does not contain in itself any simple means for estimating the error or for detecting computation mistakes. It is true that Bieberbach 6 ha found an expression which provides an upper bound for the error at a given step of the Runge-Kutta process (or more accurately, the Kutta process). However this estimate depends on quantities which do not appear directly in the computation, and therefore requires some additional separate calculation.
2. Each step requires from substitutions into the differential equation. For the case of complicated equations tIllS may demand an excessive amount of labor per step.

By accident or design it happens that examples usually chosen in textbooks to illustrate the Runge-Kutta method are such that the method appears in a very favorable light.

Higher order methods are capable of achieving highly accurate approximations of differential equations solutions at lower computational cost than lower order methods. The fact that there is no automatic construction method for (explicit) Runge-Kutta methods of a give order with a minimum number of stages makes the search for methods of higher and higher order an interesting challenge. For given order$p$ it is not known in general how large the number of stages s must be to achieve this order. For orders 1, 2, 3 and 4, the lowest possible number of stages is $s = p$. However, for $p = 5$ and $p = 6$ the lowest possibility is $s = p+1$. For, $p = 7$, $s = 9$ stages are necessary whereas for $p = 8$, the minimum number of stages is $s = 11$. Above this, very little is known.

Table 2.1: show some detail of the chronology of attempts to obtain increasingly high orders (Butcher,1996).

p	s	author	year
2	2	Runge	1895
3	3	Heun	1900
4	4	kutta	1901
5	6	kutta	1901
5	6	Nystrom	1925
6	8	Huta	1956
6	7	Butcher	1964
7	9	Butcher	1968
8	11	Curtis	1970
8	11	Cooper and Verner	1972
10	18	Curtis	1975
10	17	Hairer	1978

Runge-Kutta method in use is of order eight in difference-equation form, is given by the following (Curtis,1970).

$$w_{i+1} = w_i + h/840(41k_1 + 27k_4 + 272k_5 + 27k_6 + 216k_7 + 216k_9 + 41k_{10})$$

where

$k_1 = f(t_i, w_i)$

$k_2 = f(t_i + h(4/27), w_i + (h4/27)k_1)$

$k_3 = f(t_i + h(2/9), w_i + (h/18)(k_1 + 3k_2))$

$k_4 = f(t_i + h(1/3), w_i + (h/12)(k_1 + 3k_3))$

$k_5 = f(t_i + h(1/2), w_i + (h/8)(k_1 + 3k_4))$

$k_6 = f(t_i + h(2/3), w_i + (h/54)(13k_1 - 27k_3 + 42k_4 + 8k_5))$

$k_7 = f(t_i + h(1/6), w_i + (h/4320)(389k_1 - 54k_3 + 966k_4 - 824k_5 + 243k_6))$

$k_8 = f(t_i + h, w_i + (h/20)(-234k_1 + 81k_3 - 1164k_4 + 656k_5 - 122k_6 + 800k_7))$

$k_9 = f(t_i + h(5/6), w_i + (h/288)(-127k_1 + 18k_3 - 678k_4 + 456k_5 - 9k_6 + 576k_7 + 4k_8))$

$k_{10} = f(t_i + h, w_i + (h/820)(1481k_1 - 81k_3 + 7104k_4 - 3376k_5 + 72k_6 - 5040k_7 - 60k_8 + 720k_9))$

for each $i = 0, 1, ..., N-1$.

2.3 Recent Development

Opanuga *et.al*. (2015), proposed a novel approach for solving quadratic Riccati differential equations. Here, the authors considered a numerical technique called differential transform method for the solution of Riccati differential equations.

Khalid *et.al*. (2015) offered, an effective perturbation iteration algorithm for solving Riccati differential equations . Their iteration algorithm is tactfully employed to obtain the approximate solution of some nonlinear Riccati differential equations. The competence and accuracy of the method presented is demonstrated with the help of three examples, inclusive of the quadratic Riccati Equation.

Biazar and Eslami (2010) used differential transform method for solving quadratic

Riccati differential equation. Here, the authors consider differential transform method to solve quadratic Riccati differential equation. The results derived by differential transform method are compared with the results of homotopy analysis method and Adomian decomposition method and It is shown that this method used for quadratic Riccati differential equation is more effective and promising than homotopy analysis method and Adomain decomposition method.

Gemechis and Tesfaye (2016), used classical RK4 method for solving the nonlinear Riccati quadratic differential equation. The stability of the method for the problem under consideration has also been investigated. The approximate solution obtained by the proposed method versus the exact solution for different values of mesh size on some nodal points has been given and validated the efciency of the method using four model examples.

Fateme and Esmaile (2017), introduce an approximate solution for quadratic Riccati differential equation. In their technique, the Bezier curves method is considered as an algorithm to find the approximate solution of the nonlinear Riccati equation. Some numerical examples are given to demonstrate the computational efficiency of their method.

Chapter 3

Methodology

3.1 Study period and site

The study was conducted at Jimma University under the department of Mathematics from August 2018 to February 2020 G.C.

3.2 Study Design

This study employs mixed-design (documentary review design and experimental design) on quadratic Riccati differential equation

3.3 Source of Information

The relevant sources of information for this study are books, published articles and related studies from internet and the experimental result will be obtained by writing MATLAB code for the present numerical methods

3.4 Mathematical Procedure of the Study

In order to achieve the stated objectives, the study follow the following procedures:

1. Define the problem.

2. Discretize the solution domain.

3. Use 8^{th} order Runge kutta method as starter.
4. Develop the Eighth order Adams-Bashforth Predictor scheme for the problem.
5. Develop the Eighth order Adams-Moulton Corrector scheme for the problem.
6. Establish the stability and convergence of the proposed scheme.
7. Write MATLAB code for the proposed scheme.
8. Validate using numerical examples.

Chapter 4

Description of the Method, Results and Discussion

4.1 Description of the Method

Consider the quadratic Riccati differential equation of the form:

$$\frac{dy}{dx} = p(x) + q(x)y + r(x)y^2, \quad y(x_0) = \alpha, \ \ x_0 \leq x \leq x_f$$

where $p(x), \ q(x)$ and $r(x)$ are continuous with $r(x) \neq 0$ and $x_0, \ x_f, \ \alpha$ are arbitrary constants for $y(x)$, which is unknown function.To describe the scheme, we denote the problem in Eq.(1.1) as:

$$\frac{dy}{dx} = f(x,y) \tag{4.1}$$

And divide the interval $[x_0, x_f]$ into N equal sub intervals of mesh length h and the mesh points given by $x_i = x_0 + ih$, i=1, 2, ..., N. Then $h = \frac{x_f - X_0}{N}$, where N is positive integer. Integrating Eq. (4.1) on interval $[x_i, x_{i+1}]$ we obtain:

$$\int_{x_i}^{x_{i+1}} \frac{dy}{dx} dx = \int_{x_i}^{x_{i+1}} f(x,y)dx$$

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} f(x,y)dx$$

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x,y)dx \tag{4.2}$$

To derive the method, we approximate $f(x,y)$ by Newton's backward difference interpolation polynomials.

4.1.1 Description of Adams-Bashforth Predictor Method

Let us take N data values (x_i, f_i), (x_{i-1}, f_{i-1}), (x_{i-2}, f_{i-2}),..., (x_{i-N+1}, f_{i-N+1}), . For this data, we fit the Newton's backward difference interpolating polynomial of degree $N-1$ and we get: (Erwin *et.al*. 2011)

$$P_{N-1}(x) = f(x_i + kh) = f_i + k\nabla f_i + \frac{k(k+1)}{2!}\nabla^2 f_i + \ldots$$
$$+\frac{k(k+1(k+2)\ldots(k+N-2))}{(N-1)!}\nabla^{N-1} f_i + T_N^p \qquad (4.3)$$

where $k = \frac{x-x_i}{h}$ and T_N^p=$\frac{k(k+1(k+2)\ldots(k+N-1))}{N!}$ h^N $f^{(N)}(\zeta)$ is the error term, where ζ lies in the same interval containing the points x_i, x_{i-1}, . . . ,x_{i-N+1} and N the limit of integration in Eq.(4.2) becomes: $x = x_i \implies k = 0$, $x = x_{i+1} \implies k = 1$, and $dx = hdk$. Replacing $f(x,y)$ by $P_{N-1}(x)$ in Eq.(4.2) and using Eq.(4.3) we get:

$$y(x_{i+1}) = y(x_i) + \int_0^1 \{f_i + k\nabla f_i + \frac{k(k+1)}{2!}\nabla^2 f_i + \ldots\}dx \qquad (4.4)$$

By choosing different values for N, we get different methods. But for this particular study, we choose the value for $N = 8$ which is of order eighth method. Now, on integrating term by term in Eq. (4.4) with respect to k , we obtain:
$\int_0^1 kdk = \frac{1}{2}$,
$\int_0^1 k(k+1)dk = \frac{5}{6}$,
$\int_0^1 k(k+1)(k+2)dk = \frac{9}{4}$,
$\int_0^1 k(k+1)(k+2)(k+3)dk = \frac{251}{30}$,
$\int_0^1 k(k+1)(k+2)(k+3)(k+4)dk = \frac{475}{12}$,
$\int_0^1 k(k+1)(k+2)(k+3)(k+4)(k+5)dk = \frac{19087}{84}$,
$\int_0^1 k(k+1)(k+2)(k+3)(k+4)(k+5)(k+6)dk = \frac{36799}{24}$,
$\int_0^1 k(k+1)(k+2)(k+3)(k+4)(k+5)(k+6)(k+7)dk = \frac{1070017}{90}$,

Thus, we get:

$$
\begin{aligned}
y(x_{i+1}) &= y(x_i)+h[f_i+\frac{1}{2}\nabla f_i+\frac{5}{12}\nabla^2 f_i+\frac{3}{8}\nabla^3 f_i+\frac{251}{720}\nabla^4 f_i+\frac{475}{144}\nabla^5 f_i+\frac{19087}{60480}\nabla^6 f_i \\
&\quad +\frac{5257}{17280}\nabla^7 f_i]+T_8 \\
&= y_i+h[f_i+\frac{1}{2}(f_i-f_{i-1})+\frac{5}{12}(f_i-2f_{i-1}+f_{i-2})+\frac{3}{8}(f_i-3f_{i-1}+3f_{i-2}-f_{i-3}) \\
&\quad +\frac{251}{720}(f_i-4f_{i-1}+6f_{i-2}-4f_{i-3}+f_{i-4})+\frac{475}{144}(f_i-5f_{i-1}+10f_{i-2}-10f_{i-3}+ \\
&\quad 5f_{i-4}-f_{i-5})+\frac{19087}{60480}(f_i-6f_{i-1}+15f_{i-2}-20f_{i-3}+15f_{i-4}-6f_{i-5}+f_{i-6}) \\
&\quad +\frac{5257}{17280}(f_i-7f_{i-1}+21f_{i-2}-35f_{i-3}+35f_{i-4}-21f_{i-5}+7f_{i-6}+f_{i-7})]+T_8 \\
&= y_i+\frac{h}{120960}[434241f_i-1152169f_{i-1}+2183877f_{i-2}-2664477f_{i-3}+2102243f_{i-4} \\
&\quad -1041723f_{i-5}+295767f_{i-6}-36799f_{i-7}]+T_8
\end{aligned}
\tag{4.5}
$$

where $T_8^P = \frac{1070017}{3628800}\, h^8 f^{(8)}(\zeta)$ is the local truncation error. Hence, Eq. (4.5) is called eight order predictor method.

4.1.2 Description of Adams-Moulton Corrector Method

Let us take $N+1$ data values (x_{i+1}, f_{i+1}), (x_i, f_i), (x_{i-1}, f_{i-1}), (x_{i-2}, f_{i-2}),..., (x_{i-N+1}, f_{i-N+1}),. For this data, we fit the Newton's backward difference interpolating polynomial of degree N and we get: (Erwin *et.al*. 2011)

$$
\begin{aligned}
P_N(x) = f(x_i+kh) = & f_{i+1}+(k-1)\nabla f_{i+1}+\frac{k(k-1)}{2!}\nabla^2 f_{i+1}+...+ \\
& \frac{(k-1)k(k+1)...(k+N-1))}{N!}\nabla^N f_{i+1}+T_N^c
\end{aligned}
\tag{4.6}
$$

where $k=\frac{x-x_i}{h}$, $x-x_{i+1}=(x-x_i)-(x_{i+1}-x_i)=kh-h=h(k+1)$ and

$$
T_N^c=\frac{k(k+1)(k+2)...(k+N-1))}{(N+1)!}h^{N+1}f_{N+1}(\zeta)
$$

is the error term, when ζ lies in some interval containing the points x_{i+1}, x_i, x_{i-1}, x_{i-2},..., x_{i-N+1} and x .

The limits of integration in Eq. (4.2) becomes:

$x = x_i \implies k = 0, x = x_{i+1} \implies k = 1$ and $dx = hdk$.

Replacing $f(x,y)$ by $P_{N-1}(x)$ in Eq. (4.2) and using Eq. (4.6) we get:

$$y(x_{i+1}) = y(x_i) + \int_0^1 \{f_{i+1} + (k-1)\nabla f_{i+1} + \frac{k(k-1)}{2!}\nabla^2 f_{i+1} + ...\}dx \tag{4.7}$$

By choosing different values for N, we get different methods. But for this particular study, we choose the value for $N = 7$ which is of order eighth method.

Now, on integrating term by term in Eq. (4.7) with respect to k , we obtain:

$\int_0^1 (k-1)dk = -\frac{1}{2}$,
$\int_0^1 (k-1)kdk = -\frac{1}{6}$,
$\int_0^1 (k-1)k(k+1)dk = -\frac{1}{4}$,
$\int_0^1 (k-1)k(k+1)(k+2)dk = -\frac{19}{30}$,
$\int_0^1 (k-1)k(k+1)(k+2)(k+3)dk = -\frac{9}{4}$,
$\int_0^1 (k-1)k(k+1)(k+2)(k+3)(k+4)dk = -\frac{863}{84}$,
$\int_0^1 (k-1)k(k+1)(k+2)(k+3)(k+4)(k+5)dk = -\frac{1375}{24}$,
$\int_0^1 (k-1)k(k+1)(k+2)(k+3)(k+4)(k+5)(k+6)dk = -\frac{33953}{90}$,

Thus, we get:

$$
\begin{aligned}
y(x_{i+1}) &= y(x_i)+h[f_{i+1}-\frac{1}{2}\nabla f_{i+1}\frac{1}{12}\nabla^2 f_{i+1}-\frac{1}{24}\nabla^3 f_{i+1}-\frac{19}{720}\nabla^4 f_{i+1}-\frac{3}{160}\nabla^5 f_{i+1}\\
&\quad -\frac{863}{60480}\nabla^6 f_{i+1}-\frac{275}{17280}\nabla^7 f_{i+1}]+T_7\\
&= y_i+h[f_{i+1}-\frac{1}{2}(f_{i+1}-f_i)-\frac{1}{12}(f_{i+1}-2f_i+f_{i-1})-\frac{1}{24}(f_{i+1}-3f_i+3f_{i-1}-f_{i-2})\\
&\quad -\frac{19}{720}(f_{i+1}-4f_i+6f_{i-1}-4f_{i-2}+f_{i-3})-\frac{3}{160}(f_{i+1}-5f_i+10f_{i-1}-10f_{i-2}\\
&\quad +5f_{i-3}-f_{i-4})-\frac{863}{60480}(f_{i+1}-6f_i+15f_{i-1}-20f_{i-2}+15f_{i-3}-6f_{i-4}+f_{i-5})\\
&\quad -\frac{275}{24192}(f_{i+1}-7f_i+21f_{i-1}-35f_{i-2}+35f_{i-3}-21f_{i-4}+7f_{i-5}+f_{i-6})]+T_7\\
&= y_i+\frac{h}{120960}[36799f_{i+1}+139849f_i-121797f_{i-1}+123133f_{i-2}-88547f_{i-3}\\
&\quad +41499f_{i-4}-11351f_{i-5}+1375f_{i-7}]+T_7 \qquad (4.8)
\end{aligned}
$$

Where, $T_7^C=-\frac{33953}{3628800}h^9\ f^{(8)}(\zeta)$ is local truncation error.
Hence, Eq.(4.8) is called eight order corrector method.

Remark 4.1 *The eighth order predictor corrector method uses y_{k-7}, y_{k-6}, y_{k-5}, y_{k-4}, y_{k-3}, y_{k-2},y_{k-1} and y_k, in the calculation of y_{k+1}. This method is not self-starting; eight points,(x_0,y_0), (x_1,y_1), (x_2,y_2), (x_3,y_3), (x_4,y_4), (x_5,y_5), (x_6,y_6) and (x_7,y_7) must be known in advance in order to generate the points (x_k,y_k) for $k\geq 8$ and we apply the eighth order Runge-Kutta method (Curtis,1970) using the first seven nodal points*

4.2 Stability and Convergence Analysis

For multi-step methods, the problems involved with consistency, convergence, and stability are compounded because of the number of approximations involved at each step.In the one step methods,the approximation y_{i+1} depends directly only on the previous approximation y_i, where as the multi-step methods use at least two of the previous approximations, and the usual methods that are employed involve more.

The general multi-step method for approximating the solution to the initial-value problem (Burden and Faires,2011)

$$\frac{dy}{dx} = f(x,y), \quad y(x_0) = \alpha, \;\; x_0 \leq x \leq x_f$$

has the form

$$y_{i+1} = \alpha_{k-1}y_i + \alpha_{k-2}y_{i-1} + \alpha_0 y_{i+1-k} + hF(x_i, y_{i+1}, y_i, ..., y_{i+1-k}) \tag{4.9}$$

where $y_0 = \alpha_0, y_1 = \alpha_1, \cdots, y_{k-1} = \alpha_{k-1},$ for each $i = k-1, k, \cdots, N-1$ and $\alpha_0, \alpha_1, \alpha_2, ..., \alpha_{k+1}$ are constants.

Throughout the analysis, two assumptions will be made concerning the function F:

- If $f \equiv 0$ (that is, if the differential equation is homogeneous), then $F \equiv 0$ also.
- F satisfies a Lipschitz condition with respect to $\{y_j\}$,in the sense that a constant L exists and, for every pair of sequences $\{V_j\}_{j=0}^{N}$ and $\{\bar{V}_j\}_{j=0}^{N}$ and for $i = m-1, m, ..., N-1$, we have

$$|F(t_i, h, V_{i+1}, ..., Vi+1-m) - F(t_i, h, \bar{V}_{i+1}, ..., \bar{V}i+1-m)| \leq L \sum_{j=0}^{m} |V_{i+1-j} - \bar{V}_{i+1-j}|$$

The explicit Adams-Bashforth and implicit Adams-Moulton methods satisfy both of these conditions, provided f satisfies a Lipschitz condition

Definition 4.1 *(Root condition): Let λ_1 , λ_2, . . . ,λ_N are the (not necessarily distinct) roots of the characteristic equation given by:*

$$P(\lambda) = \lambda^N - \alpha_{k-1}\lambda^{N-1} - ... - \alpha_1\lambda - \alpha_0 \tag{4.10}$$

If $|\lambda_i| \leq 1$ for $\quad i = 1, 2, 3, \cdots, k$ and all roots with absolute value 1 are simple roots, then the difference method is said to satisfy the root condition.

Definition 4.2 *(Stability)(David, 2008):*

i. Methods that satisfy the roots condition in which $|\lambda_i| = 1$ is the only root of the characteristic equation with magnitude one is called strongly stable.

ii. Methods that satisfy the root condition and have more than one distinct root with magnitude one is called weakly stable.

iii. Methods that do not satisfy the root condition are called unstable.

Theorem 4.1 *The eighth-order predictor method in Eq.* (4.5) *is strongly stable.*

Proof: The eighth order predictor method in Eq. (4.5) can be expressed as:

$$y_{i+1} = y_i + hF(x_i, y_{i+1}, y_i, ..., y_{i-7}) \tag{4.11}$$

where

$$\begin{aligned} hF(x_i, y_{i+1}, y_i, ..., y_{i-7}) \;\; = \;\; & \frac{h}{120960}[434241 f_i - 1152169 f_{i-1} + 2183877 f_{i-2} - 2664477 f_{i-3} \\ & +2102243 f_{i-4} - 1041723 f_{i-5} + 295767 f_{i-6} - 36799 f_{i-7}] \end{aligned}$$

In this case, we have: $k = 8$, $\alpha_0 = \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = 0$ and $\alpha_7 = 1$.
The characteristic equation for the method becomes:

$$\begin{aligned} P(\lambda) \;\; = \;\; & \lambda^8 - \lambda^7 = \lambda^7(\lambda - 1) = 0 \\ & \lambda_1 = 1, \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = \lambda_7 = \lambda_8 = 0 \end{aligned}$$

are the roots of the polynomial.
Therefore, it satisfies the root condition and is strongly stable by Definition 4.2 (i).

Theorem 4.2 *The eight-order corrector method in Eq. (4.8) is also strongly stable.*

Proof: The eight-order corrector method in Eq. (4.8) can be expressed as:

$$y_{i+1} = y_i + hF(x_i, y_{i+1}, y_i, ..., y_{i-6}) \tag{4.12}$$

where

$$\begin{aligned} hF(x_i, y_{i+1}, y_i, ..., y_{i-6}) \;\; = \;\; & \frac{h}{120960}[36799 f_{i+1} + 139849 f_i - 121797 f_{i-1} + 123133 f_{i-2} \\ & -88547 f_{i-3} + 41499 f_{i-4} - 11351 f_{i-5} + 1375 f_{i-6}] \end{aligned}$$

In this case, we have: $k = 7$, $\alpha_0 = \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = 0$ and $\alpha_6 = 1$
The characteristic equation for the method becomes:

$$\begin{aligned} P(\lambda) &= \lambda^7 - \lambda^6 = \lambda^6(\lambda - 1) = 0 \\ &\qquad \lambda_1 = 1, \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = \lambda_7 = 0 \end{aligned}$$

are the roots of the polynomial.
Therefore, it satisfies the root condition and is strongly stable by Definition 4.2 (i).

Definition 4.3 *(Consistency) Any linear multi-step method is said to be consistent, if the local truncation error $T_k(h) \to 0$ as $h \to 0$. (David, 2008)*

From Eq. (4.5) and Eq. (4.8), we get:

$$T_8^P = \frac{1070017}{3628800} h^8 f^{(8)}(\zeta) \text{ and } T_7^C = -\frac{33953}{3628800} h^7 f^{(7)}(\zeta).$$

Hence $T_k(h) \to 0$ as $h \to 0$.
Therefore, the methods in Eq. (4.5) and Eq. (4.8) are consistent by Definition 4.3.

Consistency and zero stability are the necessary and sufficient conditions for the convergence of any multi-step method. Hence, the methods are convergent since they are both consistent and stable (Burden and Faires,2011).

4.3 Numerical Examples

In order to test the validity of the proposed method, four quadratic Riccati differential equations have been considered. Since all predictor corrector methods are not a self-starter, we take the eight order Runge-Kutta method for the first seven nodal points. For each N, the point wise absolute errors are approximated by the formula, $||E|| = |y(x_i) - y_i|$ for $i = 0, 1, 2, ..., N$ and where, $y(x_i)$ and y_i are the exact and computed approximate solution of the given problem respectively, at the nodal point x_i .

Example 4.1 *Consider the following quadratic Riccati differential equation.*

$$\frac{dy}{dx}=e^{x}-e^{3x}+2e^{2x}y-e^{x}y^{2},\ \ y(0)=1,\quad 0\leq x\leq 1$$

The exact solution is $y(x)=e^{x}$.

Table 4.1: Comparison of absolute errors for example 4.1

x	N=10	N=40	N=70	N=100	N=200
present	method				
0.1	$1.6209e-14$	$0.0000e+00$	$0.0000e+00$	$2.2204e-16$	$0.0000e+00$
0.2	$5.0404e-14$	$0.0000e+00$	$2.2204e-16$	$0.0000e+00$	$0.0000e+00$
0.3	$1.2235e-13$	$0.0000e+00$	$6.6613e-16$	$0.0000e+00$	$0.0000e+00$
0.4	$2.7200e-13$	$4.4409e-16$	$4.4409e-16$	$4.4409e-16$	$0.0000e+00$
0.5	$5.8176e-13$	$6.6613e-16$	$4.4409e-16$	$4.4409e-16$	$0.0000e+00$
0.6	$1.2188e-12$	$4.4409e-16$	$0.0000e-16$	$6.6613e-16$	$0.0000e+00$
0.7	$2.5251e-12$	$8.8818e-16$	$8.8818e-16$	$8.8818e-16$	$0.0000e+00$
0.8	$1.2705e-11$	$1.3323e-15$	$8.8818e-16$	$1.7764e-15$	$4.4409e-16$
0.9	$2.9535e-11$	$1.3323e-15$	$2.2204e-15$	$2.2204e-15$	$0.0000e+00$
1	$4.8137e-11$	$1.7764e-15$	$2.6645e-15$	$2.6645e-15$	$4.4409e-16$
Gemechis	and Tesfaye	(2016)			
0.1	$1.1153e-07$	$4.5427e-10$	$4.8711e-11$	$1.1722e-11$	$7.3475e-13$
0.2	$2.6297e-07$	$1.0710e-09$	$1.1484e-10$	$2.7633e-11$	$1.7317e-12$
0.3	$4.6838e-07$	$1.9073e-09$	$2.0451-10$	$4.9211e-11$	$3.0835e-12$
0.4	$7.4674e-07$	$3.0404e-09$	$3.2600e-10$	$7.8447e-11$	$4.9163e-12$
0.5	$1.1637e-06$	$4.5748e-09$	$4.9051e-10$	$1.1803e-10$	$7.3965e-12$
0.6	$1.6338e-06$	$6.6511e-09$	$7.1312e-10$	$1.7160e-10$	$1.0753e-11$
0.7	$2.3239e-06$	$9.4596e-09$	$1.0142e-09$	$2.4406e-10$	$1.5293e-11$
0.8	$3.2569e-06$	$1.3257e-08$	$1.4214e-09$	$3.4203e-10$	$2.1432e-11$
0.9	$4.5182e-06$	$1.8390e-08$	$1.9717e-09$	$4.7445e-10$	$2.9703e-11$
1	$6.2225e-06$	$2.5327e-08$	$2.7154e-09$	$6.5340e-10$	$4.0941e-11$

Table 4.2: The maximum absolute errors for Examples 4.1 with different values of N.

N=10	N=50	N=100	N=150	N=200
$4.8137e-11$	$1.3323e-15$	$2.6645e-15$	$3.5527e-15$	$4.4409e-16$

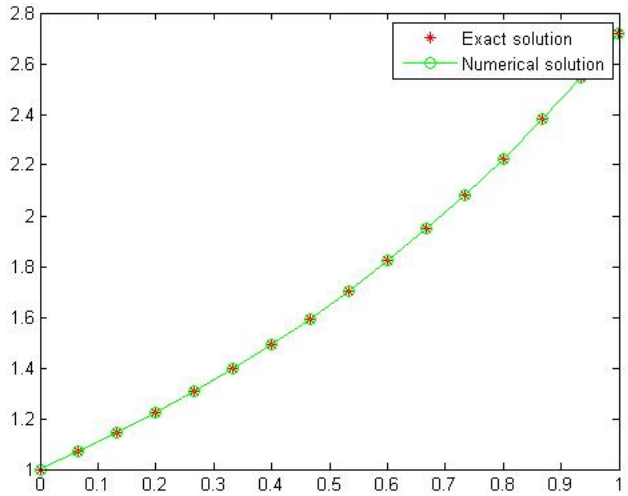

Figure 4.1: The graph of numerical and exact solution of example 4.1 for N=15

Example 4.2 *Consider the following quadratic Riccati differential equation.*

$$\frac{dy}{dx} = -\sin x + (\cos^2 x)y(x) + (\cos x)y^2(x), \quad y(0) = 1, \qquad 0 \le x \le 1$$

The exact solution is $y(x) = \cos x$*.*

Table 4.3: Comparison of absolute errors for example 4.2

x	N=10	N=40	N=70	N=100	N=200	N=400
present	method					
0.1	$1.9151e-13$	$4.2077e-12$	$9.0683e-13$	$1.5854e-13$	$5.2180e-15$	$3.3307e-16$
0.2	$4.7656e-10$	$1.1793e-11$	$8.2212e-13$	$1.4400e-13$	$4.7740e-15$	$1.1102e-16$
0.3	$9.0813e-10$	$1.0744e-11$	$7.4840e-13$	$1.3112e-13$	$4.3299e-15$	$1.1102e-16$
0.4	$8.8986e-10$	$9.8378e-12$	$6.8534e-13$	$1.2002e-13$	$3.8858e-15$	$1.1102e-16$
0.5	$2.1039e-10$	$9.0720e-12$	$6.3194e-13$	$1.1069e-13$	$3.4417e-15$	$2.2204e-16$
0.6	$1.1254e-09$	$8.4365e-12$	$5.8775e-13$	$1.0270e-13$	$2.8866e-15$	$9.9920e-16$
0.7	$2.9308e-09$	$7.9186e-12$	$5.5200e-13$	$9.6256e-14$	$2.7756e-15$	$1.9984e-15$
0.8	$2.7489e-09$	$7.5059e-12$	$5.2336e-13$	$9.1260e-14$	$2.7756e-15$	$3.5527e-15$
0.9	$2.6555e-09$	$7.1858e-12$	$5.0115e-13$	$8.7264e-14$	$2.4425e-15$	$5.1070e-15$
1	$2.5273e-09$	$6.9467e-12$	$4.8461e-13$	$8.4599e-14$	$2.2204e-15$	$6.7724e-15$

Table 4.4: The maximum absolute error for Examples 4.2 with different values of N.

N=10	N=50	N=100	150	200
$2.5273e-09$	$2.4388e-12$	$8.4599e-14$	$9.9920e-15$	$2.2204e-15$

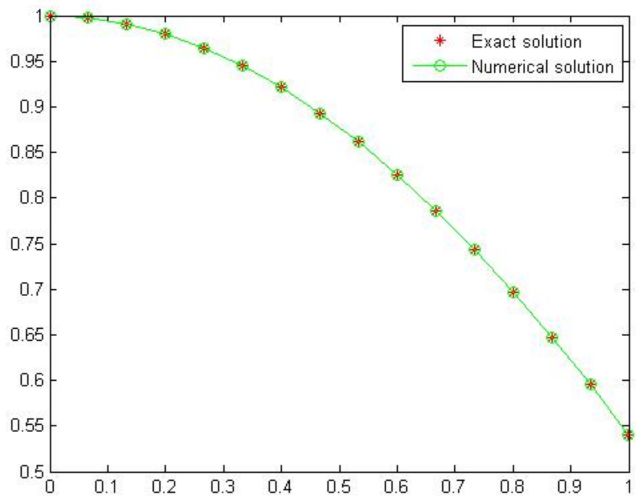

Figure 4.2: The graph of numerical and exact solution of example 4.2 for N=15

Example 4.3 *Consider the following quadratic Riccati differential equation*

$$\frac{dy}{dx} = -\frac{1}{1+x} + y(x) - y^2(x), \ \ y(0) = 1, \quad 0 \leq x \leq 1$$

The exact solution is $y(x) = \frac{1}{1+x}$.

Table 4.5: Comparison of absolute errors for example 4.3

x	N=10	N=40	N=70	N=100	N=200	N=400
present	method					
0.1	$7.2927e-09$	$1.1063e-10$	$2.0551e-11$	$5.1508e-12$	$3.4039e-13$	$2.1982e-14$
0.2	$1.1505e-08$	$1.5498e-10$	$1.8855e-11$	$4.7724e-12$	$3.1597e-13$	$2.0539e-14$
0.3	$1.3882e-08$	$1.3455e-10$	$1.7661e-11$	$4.4893e-12$	$2.9743e-13$	$1.9318e-14$
0.4	$1.5170e-08$	$1.2330e-10$	$1.6788e-11$	$4.2758e-12$	$2.8333e-13$	$1.8541e-14$
0.5	$1.5819e-08$	$1.1644e-10$	$1.6142e-11$	$4.1155e-12$	$2.7267e-13$	$1.8097e-14$
0.6	$1.6102e-08$	$1.1201e-10$	$1.5670e-11$	$3.9970e-12$	$2.6490e-13$	$1.6542e-14$
0.7	$1.6187e-08$	$1.0909e-10$	$1.5335e-11$	$3.9126e-12$	$2.5924e-13$	$1.5432e-14$
0.8	$7.5180e-08$	$1.0724e-10$	$1.5114e-11$	$3.8570e-12$	$2.5546e-13$	$1.4433e-14$
0.9	$1.1630e-07$	$1.0621e-10$	$1.4990e-11$	$3.8256e-12$	$2.5357e-13$	$1.3767e-14$
1	$1.3724e-07$	$1.0584e-10$	$1.4990e-11$	$3.8157e-12$	$2.5291e-13$	$1.3101e-14$
Gemechis	and Tesfaye	(2016)				
0.1	$3.8296e-07$	$1.2712e-09$	$1.3226e-10$	$3.1445e-11$	$1.9426e-12$	$1.2057e-13$
0.2	$5.7951e-07$	$1.9396e-09$	$2.0206e-10$	$4.8062e-11$	$2.9710e-12$	$1.8452e-13$
0.3	$6.8133e-07$	$2.2939e-09$	$2.3918e-10$	$5.6914e-11$	$3.5196e-12$	$2.1860e-13$
0.4	$7.3394e-07$	$2.4816e-09$	$2.5893e-10$	$6.1630e-11$	$3.8125e-12$	$1.8452e-13$
0.5	$7.6091e-07$	$2.5808e-09$	$2.6941e-10$	$6.4137e-11$	$3.9686e-12$	$2.4647e-13$
0.6	$7.7483e-07$	$2.6340e-09$	$2.7506e-10$	$6.5490e-11$	$4.0530e-12$	$2.5280e-13$
0.7	$7.8257e-07$	$2.6648e-09$	$2.7834e-10$	$6.6278e-11$	$4.1022e-12$	$2.5668e-13$
0.8	$7.8799e-06$	$2.6865e-09$	$2.8066e-10$	$6.6837e-11$	$4.1374e-12$	$2.5946e-13$
0.9	$7.9326e-06$	$2.7069e-09$	$2.8284e-10$	$6.7358e-11$	$4.1697e-12$	$2.6190e-13$
1	$7.9961e-06$	$2.7304e-09$	$2.8533e-10$	$6.7954e-11$	$4.2070e-12$	$2.6240e-13$

Table 4.6: The maximum absolute errors for Examples 4.3 with different values of N.

N=10	N=50	N=100	N=150	N=200
$1.3724e-07$	$5.1045e-11$	$3.8157e-12$	$7.8493e-13$	$2.5291e-13$

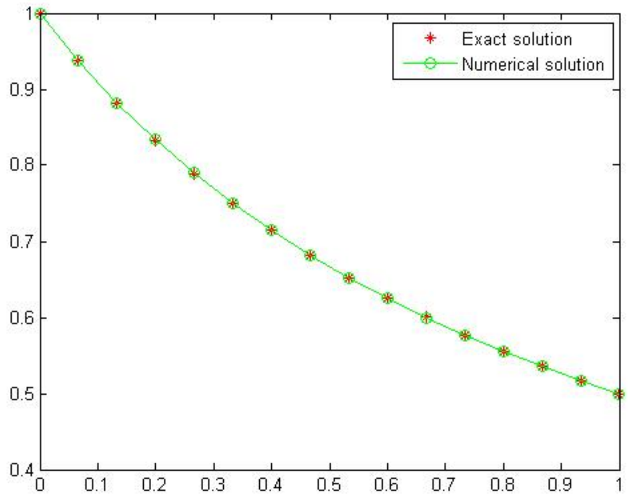

Figure 4.3: The graph of numerical and exact solution of example 4.3 for N=15

Example 4.4 *Consider the following quadratic Riccati differential equation.*

$$\frac{dy}{dx} = -y(x) + y^2(x), \;\; y(0) = \frac{1}{2}, \quad 0 \leq x \leq 1$$

The exact solution is $y(x) = \frac{e^{-x}}{1+e^{-x}}$.

Table 4.7: Comparison of absolute errors for example 4.4

x	N=10	N=40	N=70	N=100	N=200
present	method				
0.1	$5.1668e-11$	$5.2819e-13$	$9.0983e-14$	$1.5321e-14$	$5.5511e-16$
0.2	$1.5901e-10$	$1.4886e-12$	$9.0372e-14$	$1.5321e-14$	$4.4409e-16$
0.3	$3.2072e-10$	$1.4712e-12$	$8.9208e-14$	$1.5044e-14$	$5.5511e-16$
0.4	$5.3435e-10$	$1.4466e-12$	$8.7708e-14$	$1.4710e-14$	$5.5511e-16$
0.5	$7.9628e-10$	$1.4154e-12$	$8.5820e-14$	$1.4433e-14$	$4.4409e-16$
0.6	$1.1017e-09$	$1.3779e-12$	$8.3711e-14$	$1.4100e-14$	$4.4409e-16$
0.7	$1.4450e-09$	$1.3350e-12$	$8.1046e-14$	$1.3600e-14$	$4.4409e-16$
0.8	$7.4020e-09$	$1.2875e-12$	$7.8271e-14$	$1.3156e-14$	$3.3307e-16$
0.9	$1.3352e-09$	$1.2362e-12$	$7.5218e-14$	$1.2657e-14$	$2.2204e-16$
1	$1.2376e-09$	$1.1820e-12$	$7.2109e-14$	$1.2101e-14$	$3.3307e-16$
Gemechis	and Tesfaye	(2016)			
0.1	$1.3034e-09$	$5.1104e-12$	$5.4512e-13$	$1.3090e-13$	$7.7716e-15$
0.2	$2.6416e-09$	$1.0386e-11$	$1.1082e-12$	$2.6618e-13$	$1.6209e-14$
0.3	$4.0592e-09$	$1.5994e-11$	$1.7074e-12$	$4.1006e-13$	$2.5036e-14$
0.4	$5.5965e-09$	$2.2081e-11$	$2.3574e-12$	$5.6632e-13$	$3.4917e-14$
0.5	$7.2871e-09$	$2.8762e-11$	$3.0709e-12$	$7.3769e-13$	$4.5575e-14$
0.6	$9.1554e-09$	$3.6115e-11$	$3.8556e-12$	$9.2620e-13$	$5.7176e-14$
0.7	$1.1215e-08$	$4.4178e-11$	$4.7156e-12$	$1.1327e-12$	$7.0222e-14$
0.8	$1.3467e-08$	$5.2940e-11$	$5.6490e-12$	$1.3568e-12$	$8.4266e-14$
0.9	$1.5900e-08$	$6.2346e-11$	$6.6502e-12$	$1.5970e-12$	$9.9254e-14$
1	$1.8492e-08$	$7.2299e-11$	$7.7088e-12$	$1.8510e-12$	$1.1591e-13$

Table 4.8: The maximum absolute errors for Examples 4.4 with different values of N.

N=10	N=50	N=100	N=150	N=200
$1.2376e09$	$3.8669e-13$	$1.2101e-14$	$9.9920e-16$	$3.3307e-16$

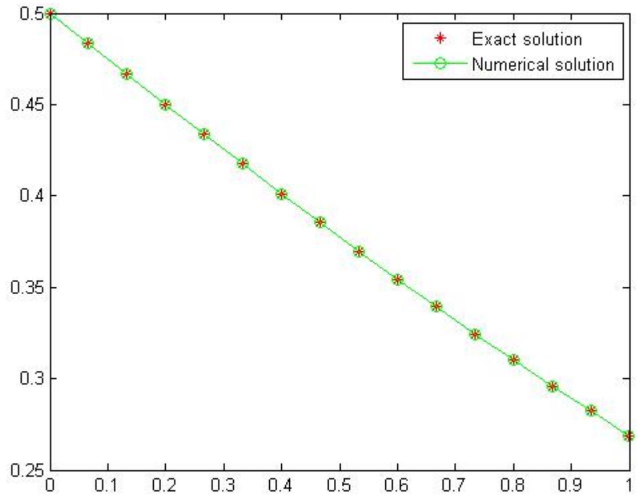

Figure 4.4: The graph of numerical and exact solution of example 4.4 for N=15

Example 4.5 *Consider the following quadratic Riccati differential equation.*

$$\frac{dy}{dx} = \left(\frac{1}{2(1+x)} - \sqrt{x+1}\right) y(x) + y^2(x), \;\; y(0) = 1, \quad 0 \leq x \leq 1$$

The exact solution is $y(x) = \sqrt{x+1}$.

Table 4.9: Comparison of absolute errors for example 4.5

x	N=10	N=40	N=70	N=100	N=200	N=400
present	method					
0.1	$1.4651e-08$	$2.4095e-10$	$4.5286e-11$	$1.1260e-11$	$7.2786e-13$	$4.6407e-14$
0.2	$3.0951e-08$	$4.5548e-10$	$5.2663e-11$	$1.3088e-11$	$8.4577e-13$	$5.3957e-14$
0.3	$4.9486e-08$	$5.2983e-10$	$6.1297e-11$	$1.5234e-11$	$9.8410e-13$	$6.2839e-14$
0.4	$7.0879e-08$	$6.1749e-10$	$7.1448e-11$	$1.7756e-11$	$1.1471e-12$	$7.4163e-14$
0.5	$9.5829e-08$	$7.1749e-10$	$8.3418e-11$	$2.0731e-11$	$1.3392e-12$	$8.6819e-14$
0.6	$1.2515e-07$	$8.4327e-10$	$9.7576e-11$	$2.4250e-11$	$1.5667e-12$	$1.0081e-13$
0.7	$1.5979e-07$	$9.8835e-10$	$1.1436e-10$	$2.8422e-11$	$1.8361e-12$	$1.1680e-13$
0.8	$1.9052e-07$	$1.1608e-09$	$1.3432e-10$	$3.3383e-11$	$2.1563e-12$	$1.3656e-13$
0.9	$2.2509e-07$	$1.3664e-09$	$1.5811e-10$	$3.9295e-11$	$2.5380e-12$	$1.6076e-13$
1	$2.6702e-07$	$1.6120e-09$	$1.8653e-10$	$4.6357e-11$	$3.0183e-12$	$1.9007e-13$

Table 4.10: The maximum absolute errors for Examples 4.5 with different values of N.

N=10	N=50	N=100	N=150	N=200
$1.2376e-07$	$6.8547e-10$	$4.6357e-11$	$9.4127e-12$	$3.0183e-12$

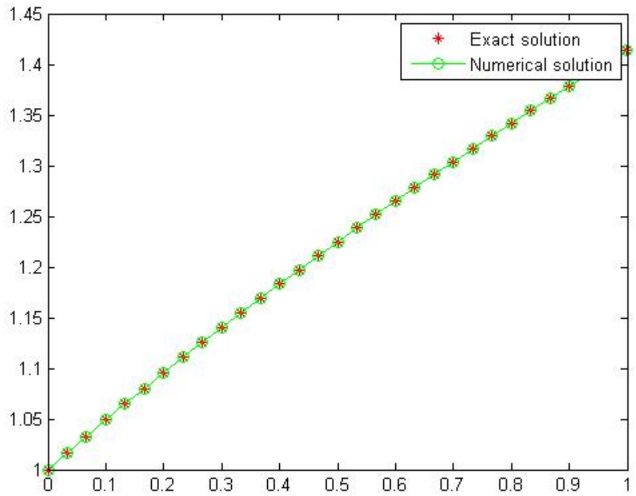

Figure 4.5: The graph of numerical and exact solution of example 4.5 for N=15

4.4 Discussion and Conclusion

In this thesis, eight order predictor-corrector method is presented for solving quadratic Riccati differential equations.To further collaborate the applicability of the proposed method; tables of point wise absolute error and graphs have been plotted for examples 1-5 for exact solution versus the numerical solutions at different values of mesh size h. The stability and convergence of the method have been investigated. The computational results are presented in the Tables. The results obtained by the present method are compared with the results of Gemechis and Tesfaye(2016). Furthermore, from the Tables it is significant that all of the absolute errors decrease rapidly as h decreases, which in turn shows the convergence of the computed solution. This shows that the small step size provides the better approximation. Briefly, the present method is stable, more accurate and effective method for solving quadratic Riccati differential equations

References

Anas, A. B., Milivoj, R. B., and Nikola Z. P. (2010). Special solutions of the Riccati equation with applications to the Gross-Pitaevskii nonlinear PDE. *Electronic Journal Of Differential Equations*, 66, 1-10.

Batiha, B. (2015). A new efcient method for solving quadratic Riccati differential equation. *International Journal of Applied Mathematics Research*, 4(1), 9-24.

Biazar, J. and Eslami, M. (2010). Differential transform method for quadratic Riccati differential equation. *International Journal of Nonlinear Science*, 9 (4), 444-447.

Burden,R.L. and Faries, J.D. (2011). *Numerical Analysis*, 9^{th} edition, Youngstown University.

Butcher, J.C. (1996). A history of Runge-Kutta methods. *Elsevier applied numerical Mathematics*, 20, 247-260.

Butcher, J.C. (2003). *Numerical methods for ordinary differential equations*. New York: John Wiley & Sons.

Curtis,A.R.(1970). An eighth order Runge-Kutta process with eleven function evaluations per step,*Elsevier applied numerical Mathematics*, 16, 268-277.

David, E. (2008). *Stability Analysis for Systems of Differential Equations*. Geometric Tools, LLC,

Fatem, G. and Esmaile, K. (2017). Approximate solution for quadratic Riccati differential equation. *Journal of Taibah University for Science*, 11, 246-250.

Gemadi, R., Gashu, G., and Kefyalew, H. (2017). Fifth order predictor-corrector method for quadratic Riccati differential equations. *International Journal of Engineering & Applied Sciences (IJEAS)*, 9, 51-64.

Gemechis File and Tesfaye Aga (2016). Numerical solution of quadratic Riccati differential equations. *Egyptian journal of basic and applied sciences*, 3, 392-397,

Ghomanjan, F. and Khorram, E. (2015). Approximate solution for quadratic Riccati differential equation. *Journal of Taibah University Science*, 11(2), 246-250.

Ince, E. L. (1956). *Ordinary differential equations*. Dover publications, New York.

John, H. M. and Kurtis, k. F. (2004). *Numerical methods using matlab*. 4^{th} ed. Prentice Hall.

Khalid, M., Mariam, S., Faheem, Z. and Uroosa, A. (2015). An effective perturbation iteration algorithm for solving Riccati differential equations. *International Journal of Computer Applications*, 111 (10), 1-5.

Mukherje, S. and Roy, B. (2012). Solution of Riccati equation with variable coefficient by differential transformation method. *International Journal of Nonlinear Science*, 14(2), 251256.

O'Connor, J. J. and Robertson, E. F. (1996). Jacopo Francesco Riccati. Retrieved from *http://wwwhistory.mcs.st-andrews.ac.uk/Biographies/Riccati.html*.

Opanuga, A. A., Edeki, S. O., Okagbue, H. I. and Akinlabi, G. O. (2015). A novel approach for solving quadratic Riccati differential equations. *International Journal of Applied Engineering Research*, 10 (11), 29121-29126.

Reid, W.T. (1972). *Riccati differential equations*. Elsevier.

Tan, Y. and Abbasbandy, S. (2008). Homotopy analysis method for quadratic Riccati Differential equation. *Communications in Nonlinear Science and Numerical Simulation*, 13(3), 539-546

Vinod, M. and Dimple, R. (2016). Newton-Raphson based modified Laplace Adomian decomposition method for solving quadratic Riccati differential equations. *MATEC web of Conferences*, (Vol. 57, P. 05001). EDP Sciences.