HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: http://ees.elsevier.com/ejbas/default.asp

Full Length Article

Numerical solution of quadratic Riccati differential equations

Gemechis File *, *Tesfaye Aga*

Department of Mathematics, Jimma University, Jimma, Ethiopia

A R T I C L E I N F O

Article history:
Received 28 February 2016
Accepted 25 August 2016
Available online 6 September 2016

Keywords:
Riccati differential equation
Runge Kutta method

A B S T R A C T

The quadratic Riccati differential equations are part of non-linear differential equations which have many applications. This paper introduces the classical fourth order Runge Kutta method (RK4) for solving the numerical solution of the quadratic Riccati differential equations. To validate the applicability of the method on the proposed equation, some model examples have been solved for different values of mesh sizes. The numerical results in terms of point wise absolute errors presented in tables and graphs show that the present method approximates the exact solution very well.

1. Introduction

The Riccati differential equation is a well-known non linear differential equation and has different applications in engineering and science domains, such as robust stabilization, stochastic realization theory, network synthesis, optimal control and in financial mathematics [1]. Non linear deferential 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 [2].

Thus, the problem has attracted much attention and has been studied by many authors. Recently, various methods are used like: using the method of Bezier curves, by developing the Bezier polynomial of degree n [3], the multistage variational iteration method is applied as a new efficient method for solving quadratic Riccati differential equation [4], using Legendre scaling functions consisting of expanding the required approximate solution as the elements of Legendre scaling functions and the operational matrix of integral, then reducing the problem to a set of algebraic equations [5] and approximate solution of generalized Riccati differential equations by iterative decomposition algorithm [6]. The solution of Riccati equation with variable co-efficient by differential transformation method (DTM) [7] has presented the absolute error between the approximated solutions which are obtained by DTM.

However, the above mentioned methods have some restrictions and disadvantages. For instance, there is a big difference between the results obtained by DTM, but as we have shown in examples there is a small point wise absolute error between the numerical result obtained by RK4 and the exact value. The classical RK4 is widely used for solving initial value problems and provides approximations which converge to the true solution as h approaches zero [8,9].

* Corresponding author.
E-mail address: gammeef@yahoo.com (G. File).
http://dx.doi.org/10.1016/j.ejbas.2016.08.006

In this paper, we introduce 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. To validate the efficiency of the method, four model examples are solved.

2. Formulation of the method

Consider the quadratic Riccati differential equation of the form:

$$\frac{dy}{dt} = p(t) + q(t)y + r(t)y^2 \tag{1}$$

with initial value condition

$$y(t_0) = \alpha \tag{2}$$

where $p(t), q(t), r(t)$ are continuous with $r(t) \neq 0$, and t_0, α are arbitrary constants for $y(t)$, which is an unknown function. To describe the scheme, we denote the problem in Eq. (1) as:

$$\frac{dy}{dt} = f(t, y) \tag{3}$$

and divide the interval $[t_0, t_f]$ into N equal subintervals of mesh length h and the mesh points given by $t_i = t_0 + ih; i = 1, 2, \ldots n$. To solve the problem, we apply the single step method that requires information about the solution at t_i to calculate t_{i+1} [8]. Let the general numerical solution of Eq. (1) be given as:

$$y_{i+1} = y_i + \sum_{i=1}^{4} w_i k_i \tag{4}$$

where

$$k_i = hf\left(t_i + c_i h, y_i + \sum_{j=1}^{3} a_{ij} k_j\right), \quad \text{for} \quad i = 1, 2, 3, 4 \tag{5}$$

and the parameters c_i, a_{ij} for $i = 2, 3, 4$; and $w_1, \ldots, w_4$ are chosen in such a way that the numerical solution y_{i+1} approximates the exact solution $y(t_{i+1})$ of Eq. (1) very well.

Now, expanding k_2, k_3, k_4 in Taylor series about t_i, substituting in Eq. (4) and matching the coefficients of h, h^2, h^3 and h^4, we obtain the systems of equations which on solving gives us:

$$c_2 = c_3 = 1/2, \quad c_4 = 1, \quad w_2 = w_3 = 1/3, \quad w_1 = w_4 = 1/6,$$
$$a_{21} = 1/2, \quad a_{31} = 0, \quad a_{32} = 1/2, \quad a_{41} = a_{42} = 0, \quad a_{43} = 1.$$

Thus, Eq. (4) becomes an explicit classical fourth order Runge Kutta method and written as:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{6}$$

where

$$k_1 = hf(t_i, y_y)$$
$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right)$$
$$k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right)$$
$$k_4 = hf(t_i + h, y_i + k_3)$$

In the determination of the parameters, since the terms up to $O(h^4)$ are compared, the truncation error is $O(h^5)$ and the order of the method is 4.

3. Stability and convergence analysis

Consider Eq. (1) at the discretized point as:

$$f(t_i, y_i) = p(t_i) + q(t_i)y(t_i) + r(t_i)y^2(t_i) \tag{7}$$

Further, consider the linear first order test differential equation:

$$y' = \lambda y, \quad y(t_0) = y_0$$

Where λ is a constant, and has its solution in the form of

$$y(t) = y(t_0)e^{(\lambda(t-t_0))} \quad \text{which at} \quad t = t_0 + nh.$$

The solution becomes:

$$y(t_n) = y(t_0)e^{\lambda nh} = y_0(e^{\lambda h})^n \tag{8}$$

Let the non-linear quadratic Riccati differential equation of the form given in Eqs. (1)–(3) and Eq. (7) written as:

$$y' = f(t, y); \quad y(t_0) = y_0 = \alpha \tag{9}$$

The non-linear function Eq. (9) can be linearized by expanding the function in Taylor series about the point (t_0, y_0) and truncating it after the first term:

$$y' = f(t_0, y_0) + (t - t_0)\frac{\partial f}{\partial t}(t_0, y_0) + (y - y_0)\frac{\partial f}{\partial y}(t_0, y_0) \tag{10}$$

Using Eq. (7) and by the chain rule differentiation, we have $y' = p_0 + q_0 y_0 + r_0 y_0^2 + (t - t_0)[p_0' + q_0' y_0 + q_0 y_0' + r_0' y_0^2 + 2r_0 y_0 y_0'] + q_0' y_0 + q_0 y_0' + r_0' y_0^2 + 2r_0 y_0 y_0'] + y[q_0 + 2r_0 y_0 y_0'] - q_0 y_0 - 2r_0 y_0^2 y_0'$ For simplicity, consider $p(t_0) = p_0, q(t_0) = q_0, r(t_0) = r_0, y(t_0) = y_0, y'(t_0) = y_0'$ and using Eq. (2) $y(t_0) = y_0 = \alpha = \text{constant}$; Thus, $y_0' = 0$.

$$\Rightarrow y' = q_0 y + p_0 + r_0 y_0^2 + (t - t_0)[p_0' + q_0' y_0 + r_0' y_0^2] \tag{11}$$

This can be written as:

$$y' = \lambda y + c \tag{12}$$

where $\lambda = q_0, c = p_0 + r_0 y_0^2 + (t - t_0)[p_0' + q_0' y_0 + r_0' y_0^2]$

Table 1 – Rate of convergence for some model examples with different mesh sizes.

	N	10	40	70	100	200	400
Rate of convergence	Example 2	3.9522	3.9851	3.9915	3.9941	3.9972	3.9935
	Example 3	3.9584	3.9904	3.9946	3.9964	4.0010	4.0581
	Example 4	4.1313	4.0350	4.0202	4.0144	4.0076	3.9293

Dividing both sides of Eq. (12) by λ, we obtain $\frac{y'}{\lambda} = y + \frac{c}{\lambda}$. If $w = y + \frac{c}{\lambda}$, then we have:

$$y = w - \frac{c}{\lambda} \tag{13}$$

Substituting Eq. (13) into Eq. (12) gives:

$$w' = \lambda w \tag{14}$$

which is called the linear test equation for the non-linear Eq. (1).

The solution of this test equation, Eq. (15), is:

$$w = ke^{\lambda t} \tag{15}$$

Now, by considering Eq. (6), we have:

$$k_1 = hf(t_n, y_n) = \lambda h y_n$$
$$k_2 = \lambda h y_n + \frac{1}{2}(\lambda h)^2 y_n = \left[\lambda h + \frac{(\lambda h)^2}{2}\right] y_n$$
$$k_3 = \lambda h y_n + \frac{1}{2}\lambda h\left[\lambda h + \frac{1}{2}(\lambda h)^2\right] y_n = \left[\lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{4}\right] y_n$$
$$k_4 = \lambda h y_n + \lambda h\left[\lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{4}\right] y_n$$
$$= \left[\lambda h + (\lambda h)^2 + \frac{(\lambda h)^3}{2} + \frac{(\lambda h)^4}{4}\right] y_n$$

On substituting the values for k_1, k_2, k_3 *and* k_4, we obtain:

$$y_{n+1} = y_n + \frac{1}{6}\lambda h y_n + \frac{1}{3}\left[\left(\lambda h + \frac{(\lambda h)^2}{2}\right) y_n\right] + \frac{1}{3}\left[\left(\lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{4}\right) y_n\right] + \frac{1}{6}\left[\left(\lambda h + (\lambda h)^2 + \frac{(\lambda h)^3}{2} + \frac{(\lambda h)^4}{4}\right) y_n\right]$$

$$\Rightarrow y_{n+1} = \left[1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24}\right] y_n$$

$$y_{n+1} = E(\lambda h) y_n \tag{16}$$

Where $E(\lambda h) = 1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24}$

From Eq. (8), it is easily observed that the exact value of $y(t_n)$ increases for the constant $\lambda > 0$ and decreases for $\lambda < 0$ with the factor $e^{\lambda h}$. While from Eq. (16) the approximate value of y_n increases or decreases with the factor of $E(\lambda h)$. If $\lambda h > 0$, then $e^{\lambda h} \geq 1$, so the fourth order RK method is relatively stable. If $\lambda h < 0$, then the interval of absolutely stable is $-2.78 < \lambda h < 0$.

The computational rate of convergence can also be obtained by using the double mesh principle defined below. Consider the numerical solution obtained by Eq. (6) and let $Z_h = \max|y_i^h - y_i^{h/2}|, i = 1, 2, \ldots, N-1$. Where y_i^h is the numerical solution on the mesh $\{t_i\}_1^{N-1}$ at the nodal point $t_i = t_0 + ih, i = 1, 2, \ldots, N-1$ and where $y_i^{h/2}$ is the numerical solution at the nodal point t_1 on the mesh $\{t_i\}_1^{2N-1}$ where, $t_i = t_0 + ih/2, i = 1, 2, \ldots, 2N-1$.

In the same way one can define $Z_{h/2}$ by replacing h by $h/2$ and $N-1$ by $2N-1$, that is, $Z_{h/2} = \max|y_i^{h/2} - y_i^{h/4}|$, $i = 1, 2, \ldots, 2N-1$.

The computed rate of convergence is defined as:

$$\text{Rate} = \frac{\log Z_h - \log Z_{h/2}}{\log 2} \tag{17}$$

Numerical examples are given to illustrate the efficiency and convergence of this method.

In Table 1, the rate of convergence for examples 2, 3 and 4 respectively is given at different mesh sizes.

4. Numerical examples

To validate the applicability of the method, four quadratic Riccati differential equations have been considered. For each N, the point wise absolute errors are approximated by the formula, $\|E\| = |y(t_i) - y_i|$, for $i = 0, 1, 2, \ldots N$ and where, $y(t_i)$ and y_i are the exact and computed approximate solution of the given problem respectively, at the nodal point t_i.

Example 1. Consider the following quadratic Riccati differential equation [3].

$$y'(t) = 16t^2 - 5 + 8ty(t) + y^2(t), \quad 0 \leq t \leq 1$$
$$y(0) = 1$$

where the exact solution is: $y(t) = 1 - 4t$.

The numerical solution in terms of point wise absolute errors by the comparing with the previous method is given in Table 2.

Example 2. Consider the following quadratic Riccati differential equation [3–6].

$$y'(t) = 1 + 2y(t) - y^2(t), \quad 0 \leq t \leq 1$$
$$y(0) = 0,$$

Table 2 – Absolute error for example 1 (mesh size $h = 0.1$ OR N = 10).

t	Method [3]	The present method
0.0	0.0000000000000	0
0.1	0.000233600365141	0
0.3	0.00045422294912	2.2204e-16
0.5	9.375e-11	2.2204e-16
0.7	0.00045422275331	2.2204e-16
0.9	0.00023360043610	4.4409e-16
1.0	0.0000000000000	8.8818e-16

where the exact solution is: $y(t) = 1 + \sqrt{2}\tanh\left(\sqrt{2}t + 0.5\ln\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)\right)$.

The numerical solution in terms of absolute errors is given in Table 3.

Example 3. Consider the following quadratic Riccati differential equation [3].

$$y'(t) = e^t - e^{3t} + 2e^{2t}y(t) - e^t y^2(t), \quad 0 \le t \le 1$$
$$y(0) = 1.$$

where the exact solution is $y(t) = e^t$.

The numerical solution in terms of absolute errors is given in Table 4.

Example 4. Consider the following Riccati differential equation [5].

$$y'(t) = \frac{-1}{1+t} + y(t) - y^2(t), \quad 0 \le t \le 1$$
$$y(0) = 1$$

with analytic solution $y(t) = \frac{1}{1+t}$. Table 5 shows that the numerical solution in terms of absolute error for different step size h.

5. Numerical results

The following graphs (Figs. 1–4) show the numerical solutions obtained by the present method versus its corresponding exact solution.

6. Discussion and conclusion

In this paper, we presented classical KR4 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 1–4 for exact solution versus the numerical solutions at different values of mesh size h. Table 2 shows that the absolute errors obtained by RK4 have been compared with absolute errors obtained by Ref. [3]. Tables 3–5 also show that the point wise absolute error decreases as the mesh size h decreases, which in turn shows the convergence of the computed solution. Generally, the present method is computationally: stable, effective, simple to use, convergent and give accurate solution than some

Table 3 – Absolute error for example 2.

t	N = 10	N = 40	N = 70	N = 100	N = 200	N = 400
0.1	2.2551e-06	9.8491e-09	1.0669e-09	2.8533e-10	1.6233e-11	1.0184e-12
0.2	4.7763e-06	2.0641e-08	2.2327e-09	5.3915e-10	3.3923e-11	2.1275e-12
0.3	7.3083e-06	3.1235e-08	3.3731e-09	8.1402e-10	5.1180e-11	3.2087e-12
0.4	9.5635e-06	4.0441e-08	4.3607e-09	1.0517e-09	6.6078e-11	4.1415e-12
0.5	1.1301e-05	4.7374e-08	5.1021e-09	1.2299e-09	7.7230e-11	4.8390e-12
0.6	1.2408e-05	5.1724e-08	5.5661e-09	1.3414e-09	8.4199e-11	5.2707e-12
0.7	1.2940e-05	5.3815e-08	5.7892e-09	1.3949e-09	8.7546e-11	5.4756e-12
0.8	1.3100e-05	5.4419e-08	5.8528e-09	1.4101e-09	8.8489e-11	5.5316e-12
0.9	1.3141e-05	5.4381e-08	5.8450e-09	1.4079e-09	8.8322e-11	5.5178e-12
1	1.3245e-05	5.4260e-08	5.8236e-09	1.4019e-09	8.7889e-11	5.5029e-12

Table 4 – Absolute errors for example 3.

t	N = 10	N = 40	N = 70	N = 100	N = 200	N = 400
0.1	1.1153e-07	4.5427e-10	4.8711e-11	1.1722e-11	7.3475e-13	4.5963e-14
0.2	2.6297e-07	1.0710e-09	1.1484e-10	2.7633e-11	1.7317e-12	1.0880e-13
0.3	4.6838e-07	1.9073e-09	2.0451e-10	4.9211e-11	3.0835e-12	1.9384e-13
0.4	7.4674e-07	3.0404e-09	3.2600e-10	7.8447e-11	4.9163e-12	3.0909e-13
0.5	1.1237e-06	4.5748e-09	4.9051e-10	1.1803e-10	7.3965e-12	4.6496e-13
0.6	1.6338e-06	6.6511e-09	7.1312e-10	1.7160e-10	1.0753e-11	6.7191e-13
0.7	2.3239e-06	9.4596e-09	1.0142e-09	2.4406e-10	1.5293e-11	9.5124e-13
0.8	3.2569e-06	1.3257e-08	1.4214e-09	3.4203e-10	2.1432e-11	1.3305e-12
0.9	4.5182e-06	1.8390e-08	1.9717e-09	4.7445e-10	2.9730e-11	1.8439e-12
1	6.2225e-06	2.5327e-08	2.7154e-09	6.5340e-10	4.0941e-11	2.5673e-12

Table 5 – Absolute errors for example 4.

t	N = 10	N = 40	N = 70	N = 100	N = 200	N = 400
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-07	2.6865e-09	2.8066e-10	6.6837e-11	4.1374e-12	2.5946e-13
0.9	7.9326e-07	2.7069e-09	2.8284e-10	6.7358e-11	4.1697e-12	2.6190e-13
1	7.9961e-07	2.7304e-09	2.8533e-10	6.7954e-11	4.2070e-12	2.6240e-13

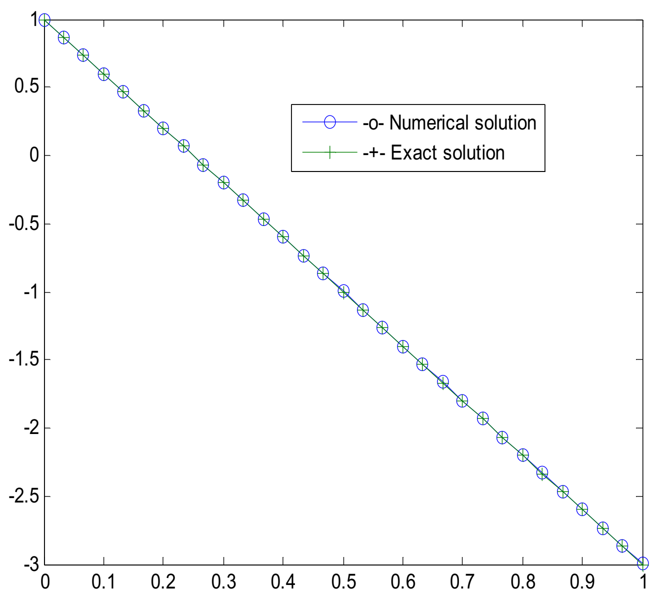

Fig. 1 – The graph of numerical and exact solution of example 1 for N = 30.

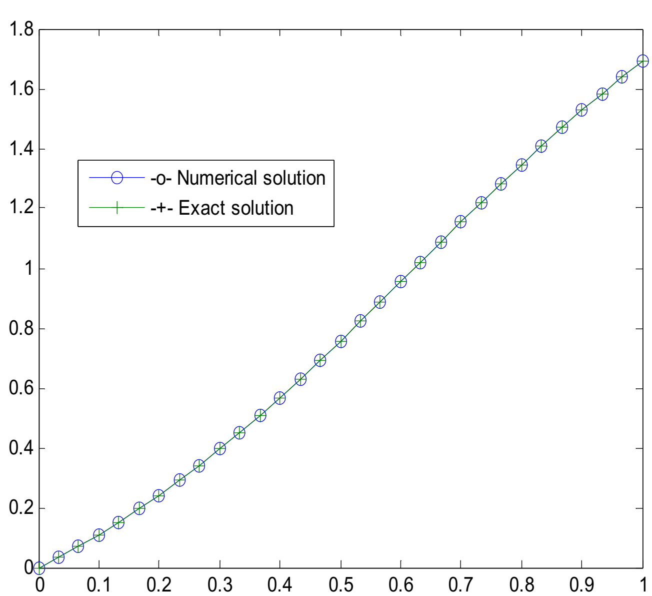

Fig. 2 – The graph of numerical and exact solution of example 2 for N = 30.

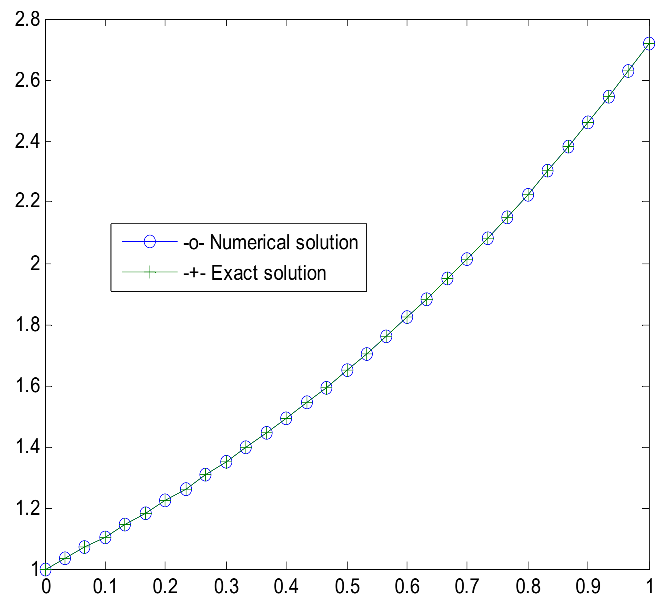

Fig. 3 – The graph of numerical and exact solution of example 3 for N = 30.

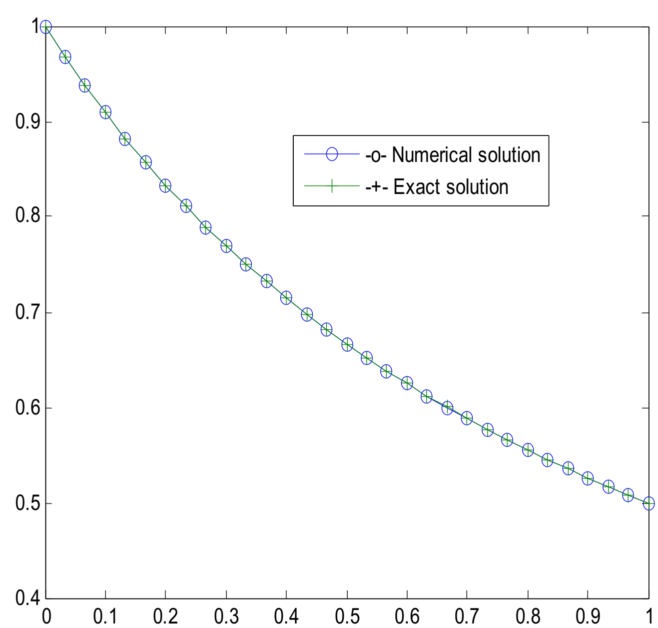

Fig. 4 – The graph of numerical and solution exact solution of example 4 for N = 30.

previously existing methods, including the more recent method in Ref. [3].

REFERENCES

[1] Riaz S, Rafiq M, Ahmad O. Non standard finite difference method for quadratic Riccati differential equation, Pakistan, Punjab University. J Math 2015;47(2).

[2] Vahidi AR Didgar M. Improving the accuracy of the solutions of Riccati equations. Int J Ind Math 2012;4(1):11–20.

[3] Ghomanjani F, Khorram E. Approximate solution for quadratic Riccati differential equation. J Taibah Univ Sci 2015. doi:10.1016/j.jtusci.2015.04.001.

[4] Batiha B. A new efficient method for solving quadratic Riccati differential equation. Int J Appl Math Res 2015;4(1):24–9.

[5] Baghchehjoughi RN, Saray BN, Lakestani M. Numerical solution of Riccati differential equation using Legendre scaling functions, Iran. J Curr Res Sci 2014;2(1):149–53.

[6] Taiwo OA, Osilagun JA. Approximate solution of generalized Riccati differential equations by iterative decomposition algorithm. Int J Eng Innovative Tech 2012;1(2).

[7] Mukherje S, Roy B. Solution of Riccati equation with variable coefficient by differential transformation method. Int J Nonlinear Sci 2012;14(2):251–6.

[8] Jain MK. Numerical solution of differential equations, finite difference and finite element methods. 3rd ed. New Age International Publishers; 2014.

[9] Jain MK, Iyengr SRK, Jain RK. Numerical methods for scientific and engineering computation. 5th ed. 2007.