
Combination of Laplace Transform and Runge-Kutta Methods for Solving the Fractional Riccati Differential Equation

Zahra Sahraee¹, Maryam Arabameri² and Ali Ahmadian³

^{1,2} Department of Mathematics, Faculty of Mathematics, Statistics and Computer Science, University of Sistan and Baluchestan, Zahedan, Iran

³ Department of Mathematics, Faculty of Engineering and Natural Sciences, Istanbul Okan University, Istanbul, Turkey

ABSTRACT. In this article, a method for solving the fractional Riccati differential equation is presented, which is based on the combination of Laplace transform and Runge-Kutta methods. In this way, first, by using the Laplace transform, the fractional derivative of Caputo in the fractional Riccati equation is converted into the ordinary derivative, and then, the resulting ordinary differential equation of the correct order is solved using the fourth-order Runge-Kutta method. Also, the error estimate and convergence are investigated. In addition, examples are provided to demonstrate the effectiveness of this method in practice. These examples show that the proposed method can give the approximate solution of fractional Riccati differential equations with high accuracy. Also, another advantage of the proposed method is that the approximate solution of fractional Riccati differential equations can be provided with appropriate accuracy in time intervals greater than one (maximum absolute errors 10^{-5} over $t \in [0, 8]$). Additionally, the proposed Laplace-based reformulation removes the need to carry the

²Corresponding author: arabameri@math.usb.ac.ir

Received: 21 June 2025

Revised: 06 September 2025

Accepted: 21 September 2025

How to Cite: Sahraee, Zahra; Arabameri, Maryam; Ahmadian, Ali. Combination of Laplace Transform and Runge-Kutta Methods for Solving the Fractional Riccati Differential Equation, *Casp.J. Math. Sci.*, **14**(2)(2025), 357-374.

This work is licensed under a Creative Commons Attribution 4.0 International License.

 Copyright © 2025 by University of Mazandaran. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution(CC BY) license(<https://creativecommons.org/licenses/by/4.0/>)

full time-history of the solution, leading to a simpler time-domain model that is easier to handle in practice.

Keywords: Fractional order Riccati equation, Caputo fractional derivative, Laplace transform method, Runge-Kutta method.

2000 Mathematics subject classification: 26A33, 44A10, 34A34; Secondary 65L06.

1. INTRODUCTION

The widespread use of fractional differential equations in various fields of science and engineering has been increasingly interesting to a growing number of scientists [1]. Fractional Riccati differential equations are regarded as a significant category of fractional differential equations. These equations, for instance, are used to describe various phenomena in engineering, financial mathematics, economics, optimal control, and transmission lines. The fractional Riccati equation is a part of the characteristic function of the rough Heston model, which is widely used for option pricing in financial markets [2]. In addition, the Riccati differential equation has numerous applications in applied science, including the one dimensional stationary Schrödinger equation and expressing solitary wave solutions of nonlinear partial differential equations via the projective Riccati equation. Such problems are also encountered in the optimal control literature [3]. For this reason, many methods have been proposed to solve these equations [4, 5].

In 2006, the fractional Riccati equation was solved using the Adomian decomposition method [6]. In 2012, Merdan used the modified Riemann-Liouville derivative to solve these equations [7]. In 2013, Yusbas used a technique based on Bernstein polynomials to obtain the approximate solutions to fractional Riccati equation [8]. Abd-Elhameed et al (2014) introduced two spectral wavelet algorithms for solving the fractional Riccati differential equations [9]. In 2017, these equations were solved using the reproducing kernel method [10]. In 2018, Agheli used the method called the trigonometric transform method to solve fractional Riccati equations [11]. In 2019, Chebyshev polynomials of the second kind method and a method based on Bessel, Legendre, and Chelishkov's fractional order basis functions were used to solve fractional Riccati differential equations [12, 13]. In 2021, the Adomian asymptotic decomposition method and fractional-order Boubaker wavelets method were used to obtain an approximate solution to fractional Riccati equations [14, 15]. In 2022, a compact finite difference method was presented to solve these equations [16]. That same year, Toma et al. introduced a method based on the combination of fractional Block-pulse functions and Bernoulli polynomials to solve fractional Riccati differential equations [1]. In 2023, methods like the Mittag-Leffler-Galerkin method, the generalized Bell collocation method, and the operational matrix method based on Lagrange polynomials were introduced for the approximate solution of the fractional Riccati equation [4, 5, 17].

In this paper, we aim to solve the initial value of fractional Riccati equation as follows [14]:

$${}_0^C D_t^\beta u(t) = r(t)u(t)^2 + q(t)u(t) + p(t), \quad t > 0, \quad 0 < \beta \leq 1. \quad (1.1)$$

$$u(0) = \alpha, \quad (1.2)$$

Whereas, $q(t)$, $r(t)$ and $p(t)$ are given functions, and ${}_0^C D_t^\beta$ is the Caputo fractional derivative operator with respect to t of order $0 < \beta \leq 1$, which is defined as follows [18]:

$${}_0^C D_t^\beta u(t) = \frac{1}{\Gamma(1-\beta)} \int_0^t (t-\tau)^{-\beta} \frac{dv(\tau)}{d\tau} d\tau.$$

The method considered in this paper is based on a combination of Laplace transform and Runge-Kutta methods to obtain an approximate solution to a fractional Riccati differential equation. Therefore, the fractional Riccati differential equation is first converted into an ordinary differential equation using Laplace transform, and the resulting ordinary differential equation is then solved using the Runge-Kutta method. In this study, the performance of the proposed method is compared against a set of well-established approaches, including: [1] the fractional-hybrid function based on Block-pulse functions and Bernoulli polynomials, [4] the Mittag-Leffler-Galerkin, [5] the Generalized Bell collocation, [11] Trigonometric basic functions, [12] Third-kind Chebyshev wavelet, [15] fractional-order Boubaker wavelets, [16] the compact finite difference, and [25] the variation of parameters method. The aim of this comparison is to evaluate both the accuracy and computational efficiency of the proposed method relative to these benchmark techniques.

The rest of the article is organized as follows. In Section 2, we first describe the Runge-Kutta method, Laplace transform, and its inverse, along with how to apply them to solve fractional Riccati equations. In Section 3, we explore the introduced method's convergence. In Section 4, we solve several fractional Riccati differential equation examples using the aforementioned combination method. In Section 5, we analyze the results obtained.

2. COMBINATION METHOD BASED ON LAPLACE TRANSFORM AND RUNGE-KUTTA METHOD

2.1. The Laplace transform method. In this subsection, firstly, we introduce the Laplace transform and inverse Laplace transform, and then, using the Caputo fractional derivative Laplace transform, and then, we convert the fractional Riccati differential equation (1.1) with the initial condition (1.2) into the non-linear integer order ordinary differential equation.

Definition 2.1. The Laplace transform of a function $u : [0, \infty) \rightarrow \mathbb{R}$, that is represented by the symbol $\tilde{u}(s)$ is defined as follow [19]:

$$\tilde{u}(s) = \mathcal{L}\{u(t), s\} = \int_0^\infty u(t)e^{-st} dt, \quad s \in \mathbb{C}.$$

Definition 2.2. The inverse Laplace transform of a function $\tilde{u}(s)$, is defined as follow [19]:

$$\mathcal{L}^{-1}\{\tilde{u}(s)\} = u(t) = \frac{1}{2\pi i} \int_{w-i\infty}^{w+i\infty} \tilde{u}(s)e^{st} ds, \quad \forall w \in \mathbb{R}.$$

We illustrate the differential transform of some required functions in the following [19]:

$$\begin{aligned} \mathcal{L}\{ {}^c_0 D_t^\beta u(t), s\} &= s^\beta \tilde{u}(s) - \sum_{k=0}^{n-1} s^{\beta-k-1} u^{(k)}(0), \quad n-1 < \beta \leq n, n \in \mathbb{N}. \\ \mathcal{L}\{u^{(n)}(t), s\} &= s^n \tilde{u}(s) - \sum_{k=0}^{n-1} s^{n-k-1} u^{(k)}(0), \quad n \in \mathbb{N}. \end{aligned}$$

Since $0 < \beta \leq 1$, we can write

$$\mathcal{L}\{ {}^c_0 D_t^\beta u(t), s\} = s^\beta \tilde{u}(s) - s^{\beta-1} u(0) = s^\beta [\tilde{u}(s) - s^{-1} u(0)], \quad (2.1)$$

We can approximate s^β using the linear Lagrange interpolation with respect to β by two support points $(0, 1)$, and $(1, s)$ as follows [22, 21, 20]:

$$s^\beta \approx \beta s^1 + (1-\beta)s^0 = \beta s + (1-\beta), \quad (2.2)$$

By substituting the equation (2.2) into (2.1), we have:

$$\begin{aligned} \mathcal{L}\{ {}^c_0 D_t^\beta u(t), s\} &\approx (\beta s + (1-\beta)) [\tilde{u}(s) - s^{-1} u(0)] \\ &= \beta s [\tilde{u}(s) - s^{-1} u(0)] + (1-\beta) [\tilde{u}(s) - s^{-1} u(0)] \\ &= \beta [s \tilde{u}(s) - u(0)] + (1-\beta) [\tilde{u}(s) - s^{-1} u(0)], \end{aligned} \quad (2.3)$$

Now, by applying the inverse Laplace transform to equation (2.3), we have:

$${}^c_0 D_t^\beta u(t) \approx \beta u'(t) + (1-\beta)[u(t) - u(0)], \quad 0 < \beta \leq 1. \quad (2.4)$$

By substituting the equation (2.4) into (1.1), we have:

$$\beta u'(t) + (1-\beta)[u(t) - u(0)] = r(t)u(t)^2 + q(t)u(t) + p(t), \quad 0 < \beta \leq 1, \quad 0 < t \leq T, \quad (2.5)$$

which

$$u(0) = \alpha. \quad (2.6)$$

Assuming $\frac{1}{\beta} = d \in \mathbb{R}^+$ and $\frac{1-\beta}{\beta} = c \in \mathbb{R}^+$, equation (2.5) can be rewritten as:

$$u'(t) = d[r(t)u(t)^2 + q(t)u(t) + p(t)] - c[u(t) - u(0)] = f(t, u(t)), \quad 0 < t \leq T, \quad (2.7)$$

which

$$u(0) = \alpha. \quad (2.8)$$

As a result, the non-linear integer order ordinary differential equation (2.7) with the initial condition (2.8) is derived from the fractional order equation (1.1) with the initial condition (1.2).

2.2. Runge-Kutta method. In this subsection, we apply the fourth-order Runge-Kutta method to solve the non-linear integer order ordinary differential equation (2.7) with initial condition (2.8). An approximate solution to this equation in the interval $[0, T]$ is what we are looking for. To do this, we divide interval $[0, T]$ into N equal subintervals of step-length h , that is:

$$h = \frac{T}{N}, \quad t_i = ih, \quad i = 0, 1, \dots, N. \quad (2.9)$$

The function $u^*(t)$ is assumed to be an approximation of the $u(t)$. We take into consideration the main formula that provides us with the value of $u^*(t_{i+1})$ in terms of $u^*(t_i)$ in the fourth-order Runge-Kutta method as follow [23]:

$$\begin{cases} k_1 = hf(t_i, u^*(t_i)), \\ k_2 = hf(t_i + \frac{h}{2}, u^*(t_i + \frac{k_1}{2})), \\ k_3 = hf(t_i + \frac{h}{2}, u^*(t_i + \frac{k_2}{2})), \\ k_4 = hf(t_i + h, u^*(t_i + k_3)), \\ u^*(t_{i+1}) = u^*(t_i) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad i = 0(1)N - 1. \end{cases} \quad (2.10)$$

Consequently, the Runge-Kutta method and the approximate solution of Equation (1.1) with initial condition (1.2) are as follows, by relations (2.10) for equation (2.7) and initial condition (2.8):

$$\begin{cases} k_1 = h \left[d \left(r(t_i)(u^*(t_i))^2 + q(t_i)u^*(t_i) + p(t_i) \right) - c(u^*(t_i) - \alpha) \right], \\ k_2 = h \left[d \left(r(t_i + \frac{h}{2})(u^*(t_i + \frac{k_1}{2}))^2 + q(t_i + \frac{h}{2})u^*(t_i + \frac{k_1}{2}) + p(t_i + \frac{h}{2}) \right) - c(u^*(t_i + \frac{k_1}{2}) - \alpha) \right], \\ k_3 = h \left[d \left(r(t_i + \frac{h}{2})(u^*(t_i + \frac{k_2}{2}))^2 + q(t_i + \frac{h}{2})u^*(t_i + \frac{k_2}{2}) + p(t_i + \frac{h}{2}) \right) - c(u^*(t_i + \frac{k_2}{2}) - \alpha) \right], \\ k_4 = h \left[d \left(r(t_i + h)(u^*(t_i + k_3))^2 + q(t_i + h)u^*(t_i + k_3) + p(t_i + h) \right) - c(u^*(t_i + k_3) - \alpha) \right], \\ u^*(t_{i+1}) = u^*(t_i) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad i = 0(1)N - 1. \end{cases} \quad (2.11)$$

3. CONVERGENCE ANALYSIS

The proposed procedure transforms the original fractional problem (1.1) with the initial condition (1.2) into an integer-order nonlinear ODE ((2.7) with the initial condition (2.8)) using the Laplace transform and

linear approximation of s^β [21]. This substitution is an analytical reformulation of the model, carried out directly in the Laplace domain. Therefore, in numerical error analysis (e.g., stability and convergence of discretization like the fourth-order Runge-Kutta), the error from this step is ignored, because once substituted, the resulting system is an exact closed-form problem for the modified model. The discrepancy from the true s^β regarded as modeling error, separate from any numerical discretization error. The integer-order ODE is then solved using the classical fourth-order Runge-Kutta method. The stability of the overall procedure is identical to the absolute stability property of the fourth-order Runge-Kutta method. Since both the Laplace and its inverse transformation are analytical, no further stability condition arises and the total method error solely arises from the Runge-Kutta scheme.

Definition 3.1. [23] A function $f(t, u)$ is said to satisfy a Lipschitz condition in the variable u on a set $D \subset \mathbb{R}^2$ if a constant L exists with

$$|f(t, u) - f(t, u^*)| \leq L|u - u^*|,$$

where (t, u) and (t, u^*) are in D and the constant L is called the Lipschitz constant.

Theorem 3.2. [23] Suppose that $D = \{(t, u) | 0 \leq t \leq T, -\infty < u < \infty\}$ and $f(t, u)$ is continuous on D . If f satisfies a Lipschitz condition on D in the variable u , then the initial-value problem

$$\begin{aligned} u'(t) &= f(t, u(t)), \quad 0 \leq t \leq T, \\ u(0) &= \alpha. \end{aligned} \quad (3.1)$$

has a unique solution $u(t)$ for $0 \leq t \leq T$.

Theorem 3.3. Suppose that $D = \{(t, u) | 0 \leq t \leq T, -\infty < u < \infty\}$ and $r, p, q, u \in C[0, T]$. If the function $f(t, u)$ satisfies the Lipschitz condition with respect to the variable u , then the differential equation (2.7) with the initial condition (2.8) has a unique solution $u(t)$ for $0 \leq t \leq T$.

Proof. To prove, it suffices to show that the function

$$f(t, u(t)) = d[r(t)u^2(t) + q(t)u(t) + p(t)] - c[u(t) - u(0)],$$

satisfies the Lipschitz condition with respect to the variable u . For all $(t, u_1), (t, u_2) \in D \subseteq \mathbb{R}^2$, we have:

$$\begin{aligned} |f(t, u_1) - f(t, u_2)| &= |d[r(t)u_1^2(t) + q(t)u_1(t) + p(t)] - c[u_1(t) - u(0)] \\ &\quad - d[r(t)u_2^2(t) + q(t)u_2(t) + p(t)] + c[u_2(t) - u(0)]| \\ &\leq d|r(t)||u_1^2(t) - u_2^2(t)| + d|q(t)||u_1 - u_2| + c|u_1 - u_2|, \end{aligned} \quad (3.2)$$

Using the first-degree Taylor polynomial of $g(u)$ around the point u_2 , we have:

$$u^2 = u_2^2 + (u - u_2) \frac{\partial g}{\partial u} \Big|_{u=u_2} = u_2^2 + (u - u_2)(2u_2),$$

Thus, we can write

$$u_1^2 = u_2^2 + 2(u_1 - u_2)u_2. \quad (3.3)$$

Substituting (3.3) into (3.2), we have:

$$|f(t, u_1) - f(t, u_2)| \leq 2d|r(t)||u_2||u_1 - u_2| + d|q(t)||u_1 - u_2| + c|u_1 - u_2|, \quad (3.4)$$

Since $r, q, u \in C[0, T]$, there exist positive numbers m_1, m_2 and m_3 such that for all $t \in [0, T]$, $|r(t)| \leq m_1$, $|q(t)| \leq m_2$ and $|u(t)| \leq m_3$.

Consequently, we can write:

$$\begin{aligned} |f(t, u_1) - f(t, u_2)| &\leq 2dm_1m_3|u_1 - u_2| + dm_2|u_1 - u_2| + c|u_1 - u_2| \\ &= (2dm_1m_3 + dm_2 + c)|u_1 - u_2|. \end{aligned} \quad (3.5)$$

□

To achieve the full $\mathcal{O}(h^4)$ convergence order of the fourth-order Runge-Kutta method, the function

$$f(t, u) = d[r(t)u^2 + q(t)u + p(t)] - c[u - \alpha],$$

must be four times continuously differentiable with respect to t ($f \in C^4[0, T]$). Since $r(t)$, $q(t)$, and $p(t)$ are smooth functions derived from physical parameters, and d and c are positive constants, this condition holds over the entire time interval.

Theorem 3.4. [24] Assume $f(t, u) \in C^4[0, T]$ (with respect to u) and its partial derivatives are bounded and assume that there exist positive numbers P and M as follows:

$$|f(t, u)| < M, \quad \left| \frac{\partial^{i+j} f}{\partial t^i \partial u^j} \right| < \frac{P^{i+j}}{M^{j-1}}, \quad i + j \leq 4,$$

then, in the fourth-order Runge-Kutta method, we have:

$$u(t_i) - u^*(t_i) = \frac{73}{720}h^5MP^4 + \mathcal{O}(h^6). \quad (3.6)$$

4. NUMERICAL EXAMPLES

To verify the theoretical convergence of the fourth-order Runge-Kutta method, we computed the solution of the transformed ODE for several decreasing time steps h . For each refinement, we calculated the maximum absolute error with respect to the analytical solution (available for our test problems) and then estimated the ROC (rate of convergence) using the standard formula

$$ROC = \log_2\left(\frac{e_h}{e_{2h}}\right),$$

where e_h and e_{2h} are, respectively, the maximum absolute errors for step size h and $2h$. In all examples, the computed orders were equal to 4 or very close to 4, matching the theoretical fourth-order Runge-Kutta

order and confirming that the method's only numerical error (fourth-order Runge-Kutta discretization error) is fully controlled.

Example 4.1. Consider the following fractional Riccati differential equation with the given initial condition: [25, 16, 15, 11, 5, 4]:

$$\begin{aligned} {}^C D_t^\beta u(t) &= 1 - u^2(t), \quad 0 < \beta \leq 1, \quad 0 < t \leq T, \\ u(0) &= 0. \end{aligned} \quad (4.1)$$

The exact solution to this problem for $\beta = 1$, is equal $u(t) = \frac{e^{2t}-1}{e^{2t}+1}$. We use the method described in Section 2 to solve this problem, and for the range of values of N , β , and T we report the numerical solutions, the absolute error of the solution at the nodes, and the maximum absolute error of this method. Additionally, we compare the results of this method with those of some existing methods for Example 4.1.

Table 1 shows the comparison of the absolute error of the solution obtained by our proposed method with the method introduced in [16], to solve Example 4.1 where the results are obtained for $t \in [0, 10]$, $\beta = 1$ and $N = 100$. Table 2 shows the maximum absolute error, convergence order, and calculation time of the proposed algorithm for Example 4.1 for $T = 1$, $\beta = 1$, and various values of N . The numerical solutions derived from the methods in [25, 16, 15, 11, 5, 4] and associated with Example 4.1 for $T = 1$, $\beta = 1$, and $N = 10$ are shown in Table 3. The results from the methods in [15, 11] and the proposed method associated with Example 4.1 for two different values of $\beta = 0.5$ and $\beta = 0.75$ are shown in Table 4. The comparison diagram between the exact solution and the numerical solutions for various values of β , $T = 1$, and $N = 10$ associated with Example 4.1 is shown in Figure 1. In the Table 5, the absolute errors obtained by the proposed method for Example 4.1 are reported for various values of $N = 50, 100$ and $\beta = 1$ and compared with the results of the methods presented in [4] and [5].

TABLE 1. Comparison of the absolute error related to the numerical solution of Example 4.1, obtained from our proposed method with the presented method in [16], for $T = 10$, $\beta = 1$, and $N = 100$.

t	Error of the proposed method	Error of [16]
1	1.44736×10^{-6}	2.3983554×10^{-6}
2	1.08038×10^{-6}	5.2634336×10^{-6}
3	3.00082×10^{-7}	1.0067173×10^{-5}
4	6.17995×10^{-8}	1.9654761×10^{-5}
5	1.1228×10^{-8}	3.8236123×10^{-5}
6	1.90694×10^{-9}	7.4334413×10^{-5}
7	3.10499×10^{-10}	1.4450373×10^{-4}
8	4.9116×10^{-11}	2.8091453×10^{-4}
9	7.60257×10^{-12}	2.4611733×10^{-4}

TABLE 2. Maximum absolute error, ROC and CPU time(s) corresponding to the approximate solution of Example 4.1 for $\beta = 1$, $T = 1$ and different values $N = 5, 10, 20, 40, 80$, and 160.

N	Maximum Absolute Error	ROC	CPU time(s)
5	2.48×10^{-5}	—	0.05
10	1.44×10^{-6}	4.1062	0.05
20	8.71×10^{-8}	4.04725	0.08
40	5.34×10^{-9}	4.02776	0.06
80	3.31×10^{-10}	4.01194	0.07
160	2.06×10^{-11}	4.00611	0.11

TABLE 3. Comparing the error of different methods, related to Example 4.1 for $\beta = 1$, $T = 1$, and $N = 10$.

t	Exact solution	Numerical solution	Error of [11]	Error of [15]	Error of [16]	Error of [25]	Error of proposed method
0.2	0.1973753203	0.1973751438	3.97×10^{-4}	3.20×10^{-7}	2.67×10^{-6}	4.3×10^{-9}	1.76×10^{-7}
0.4	0.3799489622	0.3799485359	4.73×10^{-4}	3.78×10^{-8}	2.03×10^{-6}	1.97×10^{-6}	4.26×10^{-7}
0.6	0.5370495670	0.53704879	3.99×10^{-4}	5.67×10^{-7}	1.43×10^{-6}	6.61×10^{-5}	7.72×10^{-7}
0.8	0.6640367702	0.664035622	2.48×10^{-4}	2.29×10^{-7}	7.70×10^{-7}	7.35×10^{-4}	1.14×10^{-6}
1	0.7615941560	0.761592708	7.68×10^{-5}	1.56×10^{-7}	2.21×10^{-5}	4.42×10^{-3}	1.44×10^{-6}

Example 4.2. Assume the below fractional Riccati differential equation [12, 11, 1]:

$$\begin{aligned}
 {}^c D_t^\beta u(t) &= u^2(t) - u(t), \quad 0 < \beta \leq 1, \quad 0 < t \leq T, \\
 u(0) &= 0.5.
 \end{aligned}
 \tag{4.2}$$

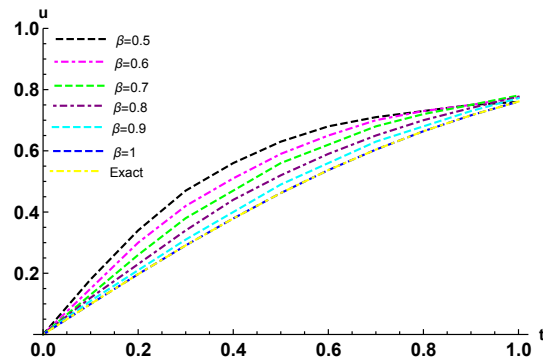
For $\beta = 1$, the exact solution to this problem is equal to $u(t) = \frac{1}{1+e^t}$. By using the proposed methods, we will resolve this example and display the results as tables and figure. A comparison between the absolute

TABLE 4. Comparing the numerical solutions, related to Example 4.1 for β .

t	$\beta = 0.5$			$\beta = 0.75$		
	Our Method	Method of [11]	Method of [15]	Our Method	Method of [11]	Method of [15]
0.2	0.345996	0.334626	0.426843	0.252216	0.260941	0.298583
0.4	0.535308	0.498466	0.543787	0.43079	0.442638	0.471284
0.6	0.680026	0.604588	0.611016	0.611699	0.67693	0.54723
0.8	0.73535	0.677429	0.666009	0.715028	0.749104	0.668492

TABLE 5. Comparison of the absolute error related to the numerical solution of Example 4.1, obtained from our proposed method with the presented method in [4] and [5], for $T = 1$, $\beta = 1$, and $N = 50, 100$.

t	Our Method $N = 50$	Our Method $N = 100$	Method of [4]	Method of [5]
0.1	1.359×10^{-10}	8.511×10^{-12}	2.824×10^{-11}	2.51×10^{-8}
0.2	2.888×10^{-10}	1.809×10^{-11}	3.467×10^{-11}	4.46×10^{-8}
0.3	4.724×10^{-10}	2.960×10^{-11}	3.502×10^{-11}	5.60×10^{-8}
0.4	6.930×10^{-10}	4.338×10^{-11}	2.737×10^{-11}	1.08×10^{-8}
0.5	9.475×10^{-10}	5.922×10^{-11}	9.246×10^{-12}	1.46×10^{-7}
0.6	1.224×10^{-9}	7.386×10^{-11}	1.095×10^{-11}	2.42×10^{-7}
0.7	1.504×10^{-9}	9.374×10^{-11}	2.270×10^{-11}	1.17×10^{-6}
0.8	1.769×10^{-9}	1.100×10^{-10}	2.390×10^{-11}	1.57×10^{-6}
0.9	2.003×10^{-9}	1.242×10^{-10}	1.557×10^{-11}	1.16×10^{-6}

FIGURE 1. Comparing the exact solution and numerical solution, related to Example 4.1 for $N = 10$, $T = 1$ and $\beta = 0.5, 0.6, 0.7, 0.8, 0.9, 1$.

error for Example 4.2 obtained for $\beta = 1$, $t \in [0, 1]$ using the proposed method with $N = 10$ and the method available in [11] with $n = 500$ is shown in Table 6. The maximum absolute error, convergence order and calculation time of the algorithm of the proposed method for $\beta = 1$,

$T = 1$ and various values of $N = 5, 10, 20, 40, 80, 160$ in Example 4.2 are shown in Table 7.

A graph of the exact solution and the numerical solutions for different values of β $N = 10$ and $T = 1$ for Example 4.2 is shown in Figure 2.

Figure 3 includes panel (A) with the absolute errors of the proposed method for Example 4.2, for values of $\beta = 1$, $N = 100$ and $T = 1$, and panel (B) with the absolute errors of the method in [1] for the same example. In the Table 9, the approximate solution obtained by the proposed method for Exmple 4.1 are reported for various values of $N = 10, 20$ and $\beta = 1$ and compared with the results of the method presented in [12].

TABLE 6. Comparison of the absolute error related to the numerical solution of Example 4.2, obtained from our proposed method with the presented method in [11], for $T = 1$, $\beta = 1$.

t	Error of the proposed method	Error of [11]
0	0	0
0.2	2.64162×10^{-9}	1.01×10^{-4}
0.4	5.59648×10^{-9}	1.34×10^{-4}
0.6	9.15537×10^{-9}	1.40×10^{-4}
0.8	1.34665×10^{-8}	1.28×10^{-4}
1	1.84416×10^{-8}	1.04×10^{-4}

TABLE 7. Maximum absolute error, ROC and CPU time(s) corresponding to the approximate solution of Example 4.2 for $\beta = 1$, $T = 1$ and different values $N = 5, 10, 20, 40, 80, 160$.

N	Maximum Absolute Error	ROC	CPU time(s)
5	2.94×10^{-7}	—	0.06
10	1.84×10^{-8}	3.998	0.08
20	1.15×10^{-9}	4	0.08
40	7.22×10^{-11}	3.993	0.09
80	4.51×10^{-12}	4	0.08
160	2.82×10^{-13}	3.999	0.08

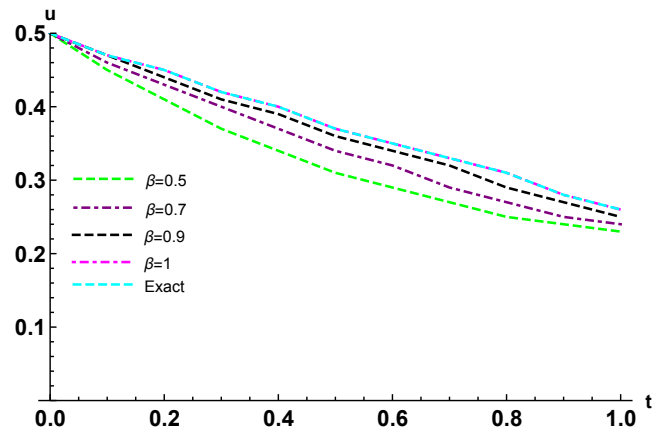


FIGURE 2. Comparing the exact solution and numerical solution, related to Example 4.2 for $N = 10$, $T = 1$ and $\beta = 0.5, 0.7, 0.8, 0.9, 1$.

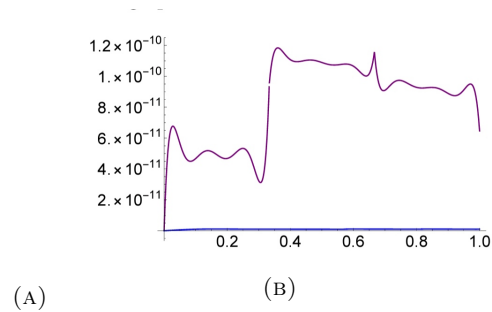


FIGURE 3. (A), The absolute error corresponding to the approximate solution of Example 4.2 for $\beta = 1$ and $N = 100$. (B), The absolute error corresponding to the approximate solution of Example 4.2 by presented method in [1].

TABLE 8. Comparison of exact solution and numerical solution related to the Example 4.2, obtained from our proposed method, for $T = 8$, $\beta = 1$ and $N = 80$.

t	Exact solution	Numerical solution	Error
0.8	0.3100255188	0.3100255323	1.34×10^{-8}
1.6	0.1679816148	0.1679816499	3.50×10^{-8}
2.4	0.0831726964	0.0831727442	4.77×10^{-8}
3.2	0.0391657227	0.0391657663	4.35×10^{-8}
4.0	0.0179862099	0.0179862415	3.15×10^{-8}
4.8	0.0081625711	0.0081625911	1.99×10^{-8}
5.6	0.0036842398	0.0036842515	1.16×10^{-8}
6.4	0.0016588010	0.0016588075	6.43×10^{-9}
7.2	0.0007460288	0.0007460322	3.43×10^{-9}
8.0	0.0003353501	0.0003353519	1.78×10^{-9}

TABLE 9. Comparison of the approximate solution related to the numerical solution of Example 4.2, obtained from our proposed method with the presented method in [12], for $T = 1$, $\beta = 1$, and $N = 10, 20$.

t	Exact solution	Our Method $N = 10$	Our Method $N = 20$	Method of [12]
0	0.500000000000	0.500000000000	0.500000000000	0.500000000690
0.1	0.475020812521	0.475020813824	0.475020812603	0.475020819612
0.2	0.450166002688	0.450166005329	0.450166002853	0.450166016633
0.3	0.425557483188	0.425557487248	0.425557483444	0.425557503350
0.4	0.401312339888	0.401312345484	0.401312340240	0.401312365921
0.5	0.377540668798	0.377540676085	0.377540669257	0.377540700506
0.6	0.354343693774	0.354343702929	0.354343694350	0.354343729236
0.7	0.331812227832	0.331812239046	0.331812228536	0.331812266624
0.8	0.310025518872	0.310025532339	0.310025519718	0.310025559907
0.9	0.289050497375	0.289050513275	0.289050498371	0.289050539822

Example 4.3. Assume the below fractional Riccati differential equation [25, 16, 12, 4]:

$$\begin{aligned} {}^c D_t^\beta u(t) &= 1 + 2u(t) - u^2(t), \quad 0 < \beta \leq 1, \quad 0 < t \leq T, \\ u(0) &= 1. \end{aligned} \quad (4.3)$$

The exact solution to this problem for $\beta = 1$ is

$$u(t) = 1 + \sqrt{2} \tanh\left(\sqrt{2}t + \frac{1}{2} \log\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)\right).$$

The results of applying the aforementioned combined method to solve Example 4.3 are as follows.

The exact and approximate solution of Example 4.3 for $N = 10$, $t \in [0, 1]$ and various values of β are shown in Figure 4. Figure 5 includes panel

(A) with the absolute errors of the proposed method for Example 4.3, for various values of $\beta = 1$, $N = 100$ and $T = 1$, and panel (B) with the absolute errors of the method in [4] for the same example. The maximum absolute error, convergence order and calculation time of the algorithm of the proposed method for $\beta = 1$, $T = 1$ and various values of N in Example 4.3 are shown in Table 10.

We can see the comparison of the absolute error for Example 4.3 obtained for $\beta = 1$, $t \in [0, 8]$ and $N = 80$ using our proposed method and the method available in [16] in Table 11. The numerical solutions and absolute errors for Example 4.3 for $\beta = 1$ obtained using our proposed method and the method available in [25] are compared in Table 12.

In the Table 13, the absolute error obtained by the proposed method for Example 4.3 are reported for various values of $N = 10, 20, 40$ and $\beta = 1$ and compared with the results of the method presented in [12].

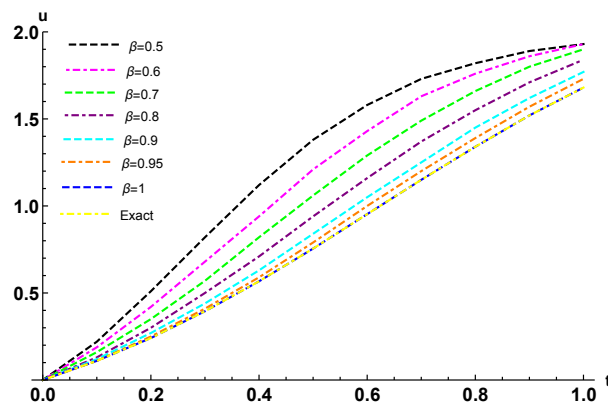


FIGURE 4. Comparing the exact solution and numerical solution, related to Example 4.3 for $N = 10$, $T = 1$ and $\beta = 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 1$.

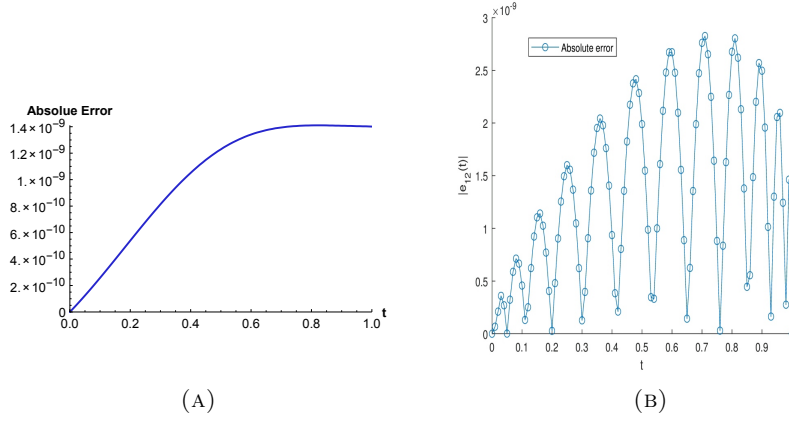


FIGURE 5. (A), The absolute error corresponding to the approximate solution of Example 4.3 for $\beta = 1$ and $N = 100$. (B), The absolute error corresponding to the approximate solution of Example 4.3 in [4].

TABLE 10. Maximum absolute error, ROC and CPU time(s) corresponding to the approximate solution of Example 4.3 for $\beta = 1$, $T = 1$ and different values $N = 5, 10, 20, 40, 80, 160$.

N	Maximum Absolute Error	ROC	CPU time(s)
5	1.97×10^{-4}	—	0.08
10	1.32×10^{-5}	3.9	0.08
20	8.54×10^{-7}	3.95	0.08
40	5.44×10^{-8}	3.973	0.09
80	3.43×10^{-9}	3.99	0.11
160	2.15×10^{-10}	4	0.12

TABLE 11. Comparison of the absolute error related to the numerical solution of Example 4.3, obtained from our proposed method with the presented method in [16], for $T = 8$, $\beta = 1$ and $N = 80$.

t	Exact solution	Our Method	Method of [16]	Error of [16]	Error of Our Method
0.8	1.346363553683	1.34635055517	1.346362994	6.61×10^{-7}	1.31×10^{-5}
1.6	2.246285959891159	2.2462736470889	2.246290755	4.79×10^{-6}	1.23×10^{-5}
2.4	2.395756424796480	2.3957525303123	2.395782816	2.63×10^{-5}	3.84×10^{-6}
3.2	2.412281528797622	2.4122808275535	2.412338083	5.65×10^{-5}	7.01×10^{-7}
4.0	2.4140123826056925	2.414012278819	2.414131848	1.19×10^{-4}	1.03×10^{-7}
4.8	2.41419262531085	2.4141926113068	2.414445422	2.52×10^{-4}	1.40×10^{-8}
5.6	2.414211383547892	2.414211381757	2.414746423	5.35×10^{-4}	1.79×10^{-9}
6.4	2.414213356339754	2.4142133541293	2.415345681	1.13×10^{-3}	2.21×10^{-10}
7.2	2.414213538777537	2.4142135387509	2.416609669	2.39×10^{-3}	2.66×10^{-11}
8.0	2.414213559917628	2.41421355991448	2.418416749	4.20×10^{-3}	3.14×10^{-12}

TABLE 12. Comparison of the absolute error related to the numerical solution of Example 4.3, obtained from our proposed method with the presented method in [25], for $T = 1$, $\beta = 1$ and $N = 10$.

t	Exact solution	Our Method	Method of [25]	Error of [25]	Error of Our Method
0.2	0.2419744004	0.2419720233	0.2419499764	2.44×10^{-5}	4.77×10^{-6}
0.4	0.5678068604	0.5678026028	0.5673979034	4.08×10^{-4}	9.56×10^{-6}
0.6	0.9535582813	0.9535538089	0.9525886597	9.69×10^{-4}	1.24×10^{-5}
0.8	1.346354258	1.346350555	1.345789984	5.64×10^{-4}	1.31×10^{-5}
1.0	1.689488974	1.689485146	1.688651308	8.37×10^{-4}	1.32×10^{-5}

TABLE 13. Comparison of the absolute error related to the numerical solution of Example 4.3, obtained from our proposed method with the presented method in [12], for $T = 1$, $\beta = 1$, and $N = 10, 20, 40$.

t	Our Method $N = 10$	Our Method $N = 20$	Our Method $N = 40$	Error of [12]
0.1	2.25×10^{-6}	1.51×10^{-7}	9.89×10^{-9}	2.14×10^{-6}
0.2	4.77×10^{-6}	3.19×10^{-7}	2.06×10^{-8}	2.54×10^{-6}
0.3	7.30×10^{-6}	4.85×10^{-7}	3.12×10^{-8}	2.80×10^{-6}
0.4	9.56×10^{-6}	6.30×10^{-7}	4.04×10^{-8}	2.86×10^{-6}
0.5	1.13×10^{-5}	7.40×10^{-7}	4.73×10^{-8}	2.66×10^{-6}
0.6	1.24×10^{-5}	8.09×10^{-7}	5.17×10^{-8}	2.34×10^{-6}
0.7	1.29×10^{-5}	8.43×10^{-7}	5.38×10^{-8}	2.34×10^{-6}
0.8	1.31×10^{-5}	8.53×10^{-7}	5.44×10^{-8}	1.41×10^{-6}
0.9	1.32×10^{-5}	8.53×10^{-7}	5.43×10^{-8}	1.01×10^{-6}

5. CONCLUSION

In this paper, an effective methodology for resolving fractional differential equations was introduced and its convergence and error were

investigated. By applying this method to a number of fractional differential equation examples, we obtained results that demonstrate the desirable advantages of the proposed method. Among these advantages, we can mention the aforementioned method's efficiency and simplicity at the same time, it considerably cuts down on calculation execution time and allows results to be reported for an interval larger than 1 with a desired level of accuracy. Additionally, this method's high accuracy in presenting results shows that it performs better than some of the other methods currently in use.

REFERENCES

- [1] A.Toma, F. Dragoi, and O. Postavaru, Enhancing the accuracy of solving Riccati fractional differential equations, *Fractal Fract.* **6**(2022), 275.
- [2] S. W. jeng, and A. Kilicman , Fractional Riccati equation and its applications to Rough Heston model using numerical methods, *Symmetry.* **12**(2020), 959.
- [3] F. Geng, Y. Lin, and M. Cui, A piecewise variational iteration method for Riccati differential equations, *Comput. Math. Appl.* **58**(2009), 2518-2522.
- [4] L. Sadek, A. S. Bataineh, H. Talibi Alaoui, and I. Hashim, The novel Mittag-Leffler–Galerkin method: Application to a Riccati differential equation of fractional order, *Fractal Fract.* **7**(2023), 302.
- [5] R. Jin, and L. Wang, Generalized bell collocation method to solve fractional Riccati differential equations, *IAENG Int. J. Appl. Math.* **53**(2023),1-7.
- [6] S. Momani, and N. Shawagfeh, Decomposition method for solving fractional Riccati differential equations, *Appl. Math. Comput.* **182**(2006),1083-1092.
- [7] M. Merdan, On the solutions fractional Riccati differential equation with modified Riemann-Liouville derivative, *Int. J. Differ. Equ.* **2012**(2012).
- [8] Ş. Yüzbaşı, Numerical solutions of fractional Riccati type differential equations by means of the Bernstein polynomials, *Appl. Math. Comput.* **219**(2013), 6328-6343.
- [9] W. M. Abd-Elhameed, and Y. H.Youssri, New ultraspherical wavelets spectral solutions for fractional Riccati differential equations, *Abstr. Appl. Anal.* **2014**(2014)
- [10] M. G. Sakar, A. Akgül, and D. Baleanu, On solutions of fractional Riccati differential equations, *Adv. Differ. Equ.* **2017**(2017),1-10.
- [11] B. Agheli, Approximate solution for solving fractional Riccati differential equations via trigonometric basic functions, *Trans. A. Razmadze Math. Inst.* **172**(2018), 299-308.
- [12] S. N. Tural-Polat, Third-kind Chebyshev wavelet method for the solution of fractional order Riccati differential equations, *J. Circuits. Syst. Comput.* **28**(2019), 1950247.
- [13] M. Izadi, Fractional polynomial approximations to the solution of fractional Riccati equation, *Punjab Univ. J. Math.* **51**(2020), 123-141.
- [14] B. Hasani Lichae, J. Biazar, and Z. Ayati, Asymptotic decomposition method for fractional order Riccati differential equation, *CMDE.* **9**(2021), 63-78.

- [15] K. Rabiei, and M. Razzaghi, Fractional-order Boubaker wavelets method for solving fractional Riccati differential equations, *Appl. Numer. Math.* **168**(2021), 221-234.
- [16] H. Porki, M. Arabameri, and R. Gharechahi, Numerical solution of nonlinear fractional Riccati differential equations using compact finite difference method, *IJNAO*. **12**(2022), 585-606.
- [17] S. Kumar, V. Gupta, A. Kumar, and J. F. Gómez-Aguilar, An accurate operational matrix method based on Lagrange polynomials for solving fractional-order pantograph delay and Riccati differential equations, *Phys. Scr.* **98**(2023), 044005.
- [18] M. Mohammad, A. Trounev, and M. Alshbool, A novel numerical method for solving fractional diffusion-wave and nonlinear Fredholm and Volterra integral equations with zero absolute error, *Axioms*. **165**(2021).
- [19] I. Podlubny, Fractional differential equations: an introduction to fractional derivatives. fractional differential equations, to methods of their solution and some of their applications, Mathematics in Science and Engineering, 1999.
- [20] M. Bishehniasar, S. Salagshour, A. Ahmadian, F. Ismail, and D. Baleanu, An Accurate Approximate-Analytical Technique for Solving Time-Fractional Partial Differential Equations, *J. Complex.* **2017**(2017), 8718209.
- [21] J. Ren, Z. Z. Sun, and W. Dai, New approximations for solving the Caputo-type fractional partial differential equations, *Appl. Math. Model.* **40**(2016), 2625-2636.
- [22] N. H. Salama, M. Ali, and N. N. A. Hamid, Fast O(N) hybrid Laplace transform-finite difference method in solving 2D time fractional diffusion equation, *J. Math. Comput. Sci.* **23**(2021), 110-123.
- [23] R. L. Burden, and D. J. Faires, Numerical Analysis, Ninth Edition.
- [24] S. Abbasbandy, and T. Allah Viranloo, Numerical solution of Fuzzy differential equation by Runge-Kutta method, *J. Sci (Kharazmi University)*. **1**(2002), 31-43.
- [25] E. UI Haq, M. Ali, and A. Saeed Khan, On the solution of fractional Riccati differential equations with variation of parameters method, *EASL*. **3**(2020), 1-9.