Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

Numerical solution of fractional-order Riccati differential equation by differential quadrature method based on Chebyshev polynomials

Jianhua Hou* and Changqing Yang

*Correspondence: cdruby@126.com
Department of Science, Huaihai Institute of Technology, Cangwu Road, Lianyungang, 222005, China

Abstract

We apply the Chebyshev polynomial-based differential quadrature method to the solution of a fractional-order Riccati differential equation. The fractional derivative is described in the Caputo sense. We derive and utilize explicit expressions of weighting coefficients for approximation of fractional derivatives to reduce a Riccati differential equation to a system of algebraic equations. We present numerical examples to verify the efficiency and accuracy of the proposed method. The results reveal that the method is accurate and easy to implement.

Keywords: Differential quadrature; Riccati differential equation; Chebyshev polynomial; Caputo derivative

1 Introduction

The fractional differential equations have received considerable interest in recent years. In many applications, fractional derivatives and fractional integrals provide more accurate models of the systems than ordinary derivatives and integrals do. Many applications of fractional differential equations in the areas of solid mechanics and modeling of viscoelastic damping, electrochemical processes, dielectric polarization, colored noise, bioengineering, and various branches of science and engineering could be found, among others, in [1].

The existence and uniqueness of solutions of fractional differential equations have been investigated in [2, 3]. In general, most of the fractional differential equations have no exact solutions. Therefore, there has been significant interest in developing approximate methods for solving this kind of equations. Several methods have recently been proposed to solve these equations. These methods include the Adomian decomposition method [4], the homotopy analysis method [5], the Adams-Bashforth-Moulton method [6, 7], the Laplace transform method [8], the Bessel function method [9–11], and so on. However, few papers reported applications of the differential quadrature method to solve fractional-order differential equations.

The differential quadrature method was introduced by Richard Bellman and his associates in the early 1970s, following the idea of integral quadrature [12]. The basic idea of the differential quadrature method is that any derivative at a mesh point can be approximated by a weighted linear sum of all the functional values along a mesh line. The key procedure in the differential quadrature method is the determination of weighting coefficients. Fung [13] introduced a modified differential quadrature method to incorporate initial conditions. He also discussed at length the stability of various grid patterns in the differential quadrature method.

In this study, we use the differential quadrature method to numerically solve the fractional-order Riccati differential equation

$$D^{\alpha}y = B(x) + C(x)y + D(x)y^2, \quad x \in [0,1], \tag{1}$$

with initial condition

$$y(0) = c, \tag{2}$$

where $0 < \alpha \leq 1$, $B(x)$, $C(x)$, and $D(x)$ are known functions, and c is a constant. When $\alpha = 1$, the fractional-order Riccati equation is the classic Riccati differential equation. The importance of the equation usually arises in the optimal control problems.

2 Preliminaries and notation

In this section, we present some notation, definitions, and preliminary facts.

2.1 Basic definitions of fractional integration and differentiation

There are various definitions of fractional integration and derivatives. The widely used definition of a fractional integral is the Riemann-Liouville definition, and that of a fractional derivative is the Caputo definition.

Definition 1 The Riemann-Liouville fractional integral operator of order $\alpha > 0$ of a function $y \in C_{\mu}$, $\mu \geq -1$, is defined as

$$J^{\alpha}y(x) = \frac{1}{\Gamma(\alpha)}\int_0^x (x-s)^{\alpha-1}f(s)\,ds, \quad \alpha > 0.$$

Definition 2 The fractional derivative D^{α} of $y(x)$ in the Caputo sense is defined as

$$D^{\alpha}y(x) = \frac{1}{\Gamma(n-\alpha)}\int_0^x (x-\tau)^{n-\alpha-1}f^{(n)}(\tau)\,d(\tau) \tag{3}$$

for $n-1 < \alpha \leq n$, $n \in N$, $t > 0$, and $y(x) \in C_{-1}^{n}$.

For the Caputo derivative, we have

$$D^{\alpha}x^{\beta} = \begin{cases} 0 & \text{for } \beta \in N_0 \text{ and } \beta < \lceil\alpha\rceil; \\ \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)}x^{\beta-\alpha} & \text{for } \beta \in N_0 \text{ and } \beta \geq \lceil\alpha\rceil \text{ or for } \beta \notin N_0 \text{ and } \beta > \lfloor\alpha\rfloor. \end{cases} \tag{4}$$

We use the ceiling function $\lceil\alpha\rceil$ to denote the smallest integer greater than or equal to α and the floor function $\lfloor\alpha\rfloor$ to denote the largest integer less than or equal to α. Also, $N = \{0,1,2,\ldots\}$.

2.2 Chebyshev polynomials and their properties

The well-known Chebyshev polynomials are defined on the interval $[-1,1]$ and are obtained by expanding the formulae

$$T_n(x) = \cos\bigl(n\arccos(x)\bigr), \quad n = 0,1,\ldots; x \in [-1,1].$$

They have the following properties:

- The three-term recurrence relation

  $$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$$

  with $T_0(x) = 1$ and $T_1(x) = x$.
- The expression of $T_n(x)$ in terms of x is given by [14]

  $$T_n(x) = \sum_{k=0}^{\lfloor n/2\rfloor} c_k^{(n)} x^{n-2k}, \tag{5}$$

  where

  $$c_k^{(n)} = (-1)^k 2^{n-2k-1} \frac{n}{n-k}\binom{n-k}{k}$$

  and

  $$c_k^{(2k)} = (-1)^k \quad (k \geq 0).$$
- Discrete orthogonality relation with the extrema of $T_n(x)$ as nodes. Let $n > 0$, $r, s \leq n$, and $x_i = -\cos(i\pi/n)$, $i = 0,1,\ldots,n$. Then

  $$\sum_{i=0}^{n}{}'' T_r(x_i)T_s(x_i) = K_r\delta_{rs}, \tag{6}$$

  where $K_0 = K_n = n$ and $K_r = \frac{1}{2}n$ for $1 \leq r \leq n-1$. The double prime indicates that the terms with suffixes $i = 0$ and $i = n$ are to be halved.

This discrete orthogonality property leads us to a very efficient interpolation formula. For later use, we write the interpolation polynomial $I_N y(x)$ interpolating $y(x)$ at the points $x_i = -\cos(i\pi/N)$, $i = 0,1,\ldots,N$, as a sum of Chebyshev polynomials in the form

$$I_N y(x) = \sum_{k=0}^{N}{}'' c_k T_k(x). \tag{7}$$

The coefficients c_k in (7) are given by the explicit formula [15]

$$c_k = \frac{2}{N}\sum_{i=0}^{n}{}'' y(x_i)T_k(x_i), \quad i = 0,1,\ldots,N. \tag{8}$$

3 Calculation of weighting coefficients of fractional-order derivatives

To apply the Chebyshev polynomials in the interval $[0,1]$, we used the shifted Chebyshev polynomials $T_n^*(x)$ defined in terms of the Chebyshev polynomials $T_n(x)$ by the relation

$$T_n^*(x) = T_n(2x-1). \tag{9}$$

Applying (7), (8), and (9), a function $y(x) \in L_2[0,1]$ is approximated by means of the shifted Chebyshev polynomials as

$$y(x) = T^*(x) \cdot P \cdot Y, \tag{10}$$

where

$$T^*(x) = \bigl[T_0^*(x), T_1^*(x), \dots, T_{N-1}^*, T_N^*(x)\bigr],$$

$$P = \begin{bmatrix} \frac{1}{2N}T_0^*(x_0) & \frac{2}{2N}T_0^*(x_1) & \frac{2}{2N}T_0^*(x_2) & \cdots & \frac{1}{2N}T_0^*(x_N) \\ \frac{1}{N}T_1^*(x_0) & \frac{2}{N}T_1^*(x_1) & \frac{2}{N}T_1^*(x_2) & \cdots & \frac{1}{N}T_1^*(x_N) \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ \frac{1}{2N}T_N^*(x_0) & \frac{2}{2N}T_N^*(x_1) & \frac{2}{2N}T_N^*(x_2) & \cdots & \frac{1}{2N}T_N^*(x_N) \end{bmatrix},$$

$$Y = \Bigl[y(x_0), y(x_1), \dots, y(x_N)\Bigr]^T,$$

and $x_i = \frac{1}{2}[1-\cos(i\pi/N)]$, $i = 0,1,2,\dots,N$. According to the definition of the Caputo fractional derivative, we can write

$$D^\alpha y(x) = D^\alpha T^*(x) \cdot P \cdot Y, \tag{11}$$

where $\alpha > 0$.

The Caputo fractional derivative of the vector $T^*(x)$ in (10) can be expressed as

$$D^\alpha T^*(x) = D^\alpha X \cdot N, \tag{12}$$

where

$$N = \begin{bmatrix} 1 & -1 & 1 & -1 & \cdots & (-1)^N \\ 0 & 2 & -8 & 18 & \cdots & (-1)^{N-1}2N^2 \\ 0 & 0 & 8 & -48 & \cdots & (-1)^{N-2}\frac{2}{3}N^2(N^2-1) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & 2^{2N-1} \end{bmatrix}$$

and

$$X = \bigl[1, x, x^2, \dots, x^N\bigr].$$

Using (4) with $0 < \alpha < 1$, we have

$$D^\alpha X = \bigl[0, c_1 x^{1-\alpha}, c_2 x^{2-\alpha}, \dots, c_N x^{N-\alpha}\bigr], \tag{13}$$

where

$$c_1 = \frac{\Gamma(2)}{\Gamma(2-\alpha)}, c_2 = \frac{\Gamma(3)}{\Gamma(3-\alpha)}, \dots, c_N = \frac{\Gamma(N+1)}{\Gamma(N+1-\alpha)}.$$

Employing (11) and (13), we get

$$Y^\alpha = \Gamma \cdot N \cdot P \cdot Y, \tag{14}$$

where

$$Y^\alpha = \Bigl[D^\alpha y(x_0), D^\alpha(x_1), \dots, D^\alpha y(x_N)\Bigr]^T$$

and

$$\Gamma = \begin{bmatrix} 0 & c_1 x_0^{1-\alpha} & c_2 x_0^{2-\alpha} & \cdots & c_N x_0^{N-\alpha} & & \\ 0 & c_1 x_1^{1-\alpha} & c_2 x_1^{2-\alpha} & \cdots & c_N x_1^{N-\alpha} & & \\ 0 & c_1 x_2^{1-\alpha} & c_2 x_2^{2-\alpha} & \cdots & c_N x_2^{N-\alpha} & & \\ \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & c_1 x_N^{1-\alpha} & c_2 x_N^{2-\alpha} & \cdots & c_N x_N^{N-\alpha} & & \end{bmatrix}.$$

Then the weighting coefficient of the fractional derivative can be written in the matrix form

$$D^{*(\alpha)} = \Gamma \cdot N \cdot P. \tag{15}$$

The weighting coefficients can be written collectively in the matrix form as

$$D^{*(\alpha)} = \begin{bmatrix} d_{00}^{(\alpha)} & d_{01}^{(\alpha)} & \cdots & d_{0N}^{(\alpha)} \\ d_{10}^{(\alpha)} & d_{11}^{(\alpha)} & \cdots & d_{1N}^{(\alpha)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N0}^{(\alpha)} & d_{N1}^{(\alpha)} & \cdots & d_{NN}^{(\alpha)} \end{bmatrix}. \tag{16}$$

4 Applications to fractional differential equation

To show the fundamental importance of weighting coefficients of fractional-order derivatives in the last section, we apply it for solving fractional-order Riccati differential equations. To solve the problem, we first consider incorporation of initial conditions. With the weighting coefficients $D^{*(\alpha)}, 0 < \alpha \le 1$, the initial condition is incorporated easily into the differential quadrature adopting the strategy of [13]:

$$D^\alpha y(x_i) = \sum_{j=0}^{N} d_{ij}^{(\alpha)} y(x_j) = d_{i0}^{(\alpha)} y(0) + \sum_{j=1}^{N} d_{ij}^{(\alpha)} y(x_j).$$

This equation can be rewritten in the matrix form as follows:

$$\begin{bmatrix} y^{(\alpha)}(x_1) \\ y^{(\alpha)}(x_2) \\ \vdots \\ y^{(\alpha)}(x_N) \end{bmatrix} = \begin{bmatrix} d_{10}^{(\alpha)} \\ d_{20}^{(\alpha)} \\ \vdots \\ d_{N0}^{(\alpha)} \end{bmatrix} \cdot y(0) + \begin{bmatrix} d_{11}^{(\alpha)} & d_{12}^{(\alpha)} & \cdots & d_{1N}^{(\alpha)} \\ d_{21}^{(\alpha)} & d_{22}^{(\alpha)} & \cdots & d_{2N}^{(\alpha)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^{(\alpha)} & d_{N2}^{(\alpha)} & \cdots & d_{NN}^{(\alpha)} \end{bmatrix} \begin{bmatrix} y(x_1) \\ y(x_2) \\ \vdots \\ y(x_N) \end{bmatrix}. \tag{17}$$

In equation (17), the initial condition is naturally incorporated into the differential quadrature rule. By substituting the approximation (17) into (1) and using the initial condition (2) we get the system of algebraic equations

$$\begin{aligned} &\begin{bmatrix} d_{10}^{(\alpha)} \\ d_{20}^{(\alpha)} \\ \vdots \\ d_{N0}^{(\alpha)} \end{bmatrix} \cdot y(0) + \begin{bmatrix} d_{11}^{(\alpha)} & d_{12}^{(\alpha)} & \cdots & d_{1N}^{(\alpha)} \\ d_{21}^{(\alpha)} & d_{22}^{(\alpha)} & \cdots & d_{2N}^{(\alpha)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^{(\alpha)} & d_{N2}^{(\alpha)} & \cdots & d_{NN}^{(\alpha)} \end{bmatrix} \begin{bmatrix} y(x_1) \\ y(x_2) \\ \vdots \\ y(x_N) \end{bmatrix} \\ &\quad = \begin{bmatrix} B(x_1) \\ B(x_2) \\ \vdots \\ B(x_N) \end{bmatrix} + \begin{bmatrix} C(x_1)y(x_1) \\ C(x_2)y(x_2) \\ \vdots \\ C(x_N)y(x_N) \end{bmatrix} + \begin{bmatrix} D(x_1)y^2(x_1) \\ D(x_2)y^2(x_2) \\ \vdots \\ D(x_N)y^2(x_N) \end{bmatrix}. \end{aligned} \tag{18}$$

Solving the system of algebraic equations, we can obtain the vector $[y_i]$. Then, using (10), we can get the approximate solutions

$$y_N(x) = T^*(x) \cdot P \cdot Y. \tag{19}$$

5 Some useful lemmas

In this section, we give some useful lemmas, which later play a significant role in the convergence analysis. We first introduce some notation. Let $I := (-1,1)$, and let $L^2_{\omega^{\alpha,\beta}}(I)$ be the space of measurable functions whose square is Lebesgue integrable in I relative to the weight function $\omega^{\alpha,\beta}(x)$. The inner product and norm of $L^2_{\omega^{\alpha,\beta}}(I)$ are defined by

$$(u,v)_{\omega^{\alpha,\beta},I} = \int_{-1}^{1} u(x)v(x)\omega^{\alpha,\beta}\,dx, \quad u,v \in L^2_{\omega^{\alpha,\beta}}(I),$$

and

$$\|u\|_{\omega^{\alpha,\beta},I} = (u,u)^{\frac{1}{2}}_{\omega^{\alpha,\beta},I}.$$

For a nonnegative integer m, define

$$H^m_{\omega^{\alpha,\beta}},I := \left\{v : \partial^k_x v \in L^2_{\omega^{\alpha,\beta}}(I), 0 \le k \le m\right\}$$

with the seminorm and the norm

$$|v|_{m,\omega^{\alpha,\beta}} = \left\|\partial^m_x v\right\|_{\omega^{\alpha,\beta}}, \quad \|v\|_{m,\omega^{\alpha,\beta}} = \left(\sum_{k=0}^{m} |v|^2_{k,\omega^{\alpha,\beta}}\right)^{\frac{1}{2}} \tag{20}$$

and

$$|v|_{H^{m;N}_{\omega^{\alpha,\beta}} I} = \left(\sum_{k=\min(m,N+1)}^{m} \left\| \partial_x^k v \right\|^2_{L^2_{\omega^{\alpha,\beta}}(I)} \right)^{\frac{1}{2}}. \tag{21}$$

To measure the truncation error, we introduce the nonuniformly weighted Sobolev space

$$B^m_{\alpha,\beta}(I) := \left\{ v : \partial_x^k v \in L^2_{\omega^{\alpha+k,\beta+k}}(I), 0 \le k \le m \right\}, \quad m \in N,$$

equipped with the norm and seminorm

$$\|v\|_{B^m_{\alpha,\beta}} = \left(\sum_{k=0}^{m} \left\| \partial_x^k v \right\|_{\omega^{\alpha+k,\beta+k}} \right)^{\frac{1}{2}} \quad \text{and} \quad |v|_{B^m_{\alpha,\beta}} = \left\| \partial_x^k v \right\|_{\omega^{\alpha+m,\beta+m}}.$$

Particularly, let

$$\omega^c(x) = \omega^{-\frac{1}{2},-\frac{1}{2}}(x)$$

be the Chebyshev weight function.

For a given positive integer N, we denote by $\{x_i\}_{i=0}^N$ the set of $N+1$ Gauss-Lobatto points corresponding to the weight $\omega^{\alpha,\beta}(x)$. By P_N we denote the space of all polynomials of degree not exceeding N. For all $v \in C[-1,1]$, we define the Lagrange interpolating polynomial $I_N^{\alpha,\beta} v \in P_N$ satisfying

$$I_N^{\alpha,\beta} v(x_i) = v(x_i).$$

The Lagrange interpolating polynomial can be written in the form

$$I_N^{\alpha,\beta} v(x) = \sum_{i=0}^{N} v(x_i) F_i(x), \quad 0 \le i \le N,$$

where $F_i(x)$ is the Lagrange interpolation basis function associated with $\{x_i\}_{i=0}^N$.

Lemma 1 ([16]) *Assume that $v \in H^m_{\omega^c}$ and denote by $I_N v$ its interpolation polynomial associated with the Gauss-Lobatto points $\{x_i\}_{i=0}^N$, namely,*

$$I_N v(x_i) = v(x_i).$$

Then we have the estimates

$$\|v - I_N v\|_{L^2_{\omega^c}} \le C N^{-m} |v|_{H^{m;N}_{\omega^c}(I)}.$$

6 Convergence analysis

In this section, we provide an error estimate of the applied method for the smooth solutions of fractional Riccati differential equations. To simplify the notation, without loss of generality, we let $C(x) = 1$ and use the change of variables to convert (1) and the initial conditions to the form

$$\frac{1}{2\Gamma(1-\alpha)}\int_{-1}^{x}(x-t)^{\alpha}u'(t)\,dt = b(x) + c(x)u(x) + d(x)u^2(x), \quad -1 \le x \le 1, \tag{22}$$

with initial conditions

$$u(-1) = c,$$

where

$$u(x) = y\left(\frac{1}{2}(1+x)\right), \qquad b(x) = B\left(\frac{1}{2}(1+x)\right), \qquad d(x) = D\left(\frac{1}{2}(1+x)\right).$$

Theorem 1 *Let $u(x)$ be the exact solution of the Riccati differential equation* (22), *which is assumed to be sufficiently smooth. Let the approximate solution $u_N(x)$ be obtained by using the differential quadrature method together with a polynomial interpolation. If $u(x) \in H^m_{\omega^c}(I)$, then, for sufficiently large N, we have the error estimate*

$$\left\|e(x)\right\|_{\omega^c} \le CN^{-m}\left(|u|_{H^{m;N}_{\omega^c}} + \left|u^2\right|_{H^{m;N}_{\omega^c}}\right) + CN^{1-m}|u|_{B^m_{\omega^c}}. \tag{23}$$

Proof Firstly, equation (22) holds at the Gauss-Lobatto points $\{x_i\}_{i=0}^{N}$ on $[-1,1]$:

$$\frac{1}{2\Gamma(1-\alpha)}\int_{-1}^{x_i}(x_i-t)^{\alpha}u'(t)\,dt = b(x_i) + u(x_i) + d(x_i)u^2(x_i), \quad u(-1) = c. \tag{24}$$

We use u_i, $0 \le i \le N$, to approximate the function value $u(x_i)$, $0 \le i \le N$, and use

$$u_N(x) = \sum_{i=0}^{N} u_i F_i(x) \tag{25}$$

to approximate the function $u(x)$, namely, $u(x_i) \approx u_i$ and $u(x) \approx u_N(x)$. Then, the numerical scheme (18) can be rewritten as

$$\frac{1}{2\Gamma(1-\alpha)}\int_{-1}^{x_i}(x_i-t)^{\alpha}u_N'(t)\,dt = b(x_i) + u_i + d(x_i)u_i^2. \tag{26}$$

Subtracting (26) from (24) gives the error equations

$$u(x_i) - u_i = \frac{1}{2\Gamma(1-\alpha)}\int_{-1}^{x_i}(x_i-t)^{\alpha}\left(u'(t) - u_N'(t)\right)dt + d(x_i)\left(u^2(x_i) - u_i^2(x)\right). \tag{27}$$

Multiplying by $F_i(x)$ both sides of (27) and summing up from $i = 0$ to $i = N$ yield

$$\begin{aligned} u(x) - u_N(x) = {} & u(x) - I_N u(x) + \frac{1}{2\Gamma(1-\alpha)}I_N\left(\int_{-1}^{x}(x-t)^{\alpha}\left(u'(t) - u_N'(t)\right)dt\right) \\ & + I_N\left(d(x)\left[u^2(x) - u_N^2(x)\right]\right). \end{aligned} \tag{28}$$

Let $e(x) = u(x) - u_N(x)$ denote the error function. Then, (28) can be written as

$$e(x) = J_1 + J_2 + J_3, \tag{29}$$

where

$$\begin{aligned} J_1 &= u(x) - I_N u(x), \\ J_2 &= I_N\bigl(D^\alpha e(x)\bigr), \\ J_3 &= I_N\bigl(d(x)\bigl[\bigl(u^2(x) - u_N^2(x)\bigr)\bigr]\bigr). \end{aligned}$$

Then we can write

$$\bigl\|e(x)\bigr\|_{\omega^c} \le \|J_1\|_{\omega^c} + \|J_2\|_{\omega^c} + \|J_3\|_{\omega^c}. \tag{30}$$

Applying Lemma 1 to $u(x)$, we have

$$\|J_1\|_{\omega^c} = \bigl\|u(x) - I_N u(x)\bigr\|_{\omega^c} \le CN^{-m}|u|_{H^{m;N}_{\omega^c}}. \tag{31}$$

Now we begin to estimate $\|J_2\|_{\omega^c}$. Also, from [17, 18] we can conclude that

$$\|J_2\|_{\omega^c} \le CN^{1-m}\bigl\|u^{(m)}\bigr\|_{\omega^{-\frac{1}{2}+m,-\frac{1}{2}+m}} = CN^{1-m}|u|_{B^m_{\omega^c}}. \tag{32}$$

We now estimate the third term $\|J_3\|_{\omega^c}$. By (27) we have

$$J_3 \le \max_{x\in[-1,1]}\bigl|b(x)\bigr|I_N\bigl(\bigl(u^2(x) - u_N^2(x)\bigr)\bigr). \tag{33}$$

By simple calculation we can rewrite J_3 as

$$J_3 \le \max_{x\in[-1,1]}\bigl|b(x)\bigr|\bigl(u^2(t) - u_N^2(t) + I_N u^2(t) - u^2(t)\bigr).$$

Therefore

$$\|J_3\|_{\omega^c} \le \max_{x\in[0,1]}\bigl|b(x)\bigr|\bigl(\bigl\|u^2(x) - u_N^2(x)\bigr\|_{\omega^c} + \bigl\|u^2(t) - I_N u^2(t)\bigr\|_{\omega^c}\bigr).$$

Since $u^2(t) - u_N^2(t) = 2u(t)e(t) - e(t)^2$, we have

$$\bigl\|u^2(t) - u_N^2(t)\bigr\|_{\omega^c} \le C\bigl\|u(t)e(t)\bigr\|_{\omega^c} + \bigl\|e(t)^2\bigr\|_{\omega^c}.$$

As in [18], applying Banach algebra theory, we can obtain

$$\bigl\|u^2(t) - u_N^2(t)\bigr\|_{\omega^c} \le C\bigl\|u(t)\bigr\|_{\omega^c}\bigl\|e(t)\bigr\|_{\omega^c} + \bigl\|e(t)\bigr\|_{\omega^c}^2.$$

Due to Lemma 1, we have

$$\bigl\|u^2(t) - I_N u^2(t)\bigr\|_{\omega^c} \le CN^{-m}\bigl|u^2\bigr|_{H^{m;N}_{\omega^c}}.$$

Consequently,

$$\|J_3\|_{\omega^c} \le CN^{-m}\left|u^2\right|_{H^{m;N}_{\omega^c}}. \tag{34}$$

Therefore, a combination of (31), (32), and (34) yields estimate (23). □

7 Illustrative examples

To illustrate the effectiveness of the proposed method, we carry out some test examples. The results obtained by this method reveal that it is very effective and convenient for fractional differential equations.

Example 1 As the first example, we consider the fractional Riccati differential equation

$$D^{\frac{1}{2}}y(x) = -x^2\left(1 + x^{\frac{5}{2}}\right) + \frac{8}{3\sqrt{\pi}}x^{\frac{3}{2}} + y(x) + \sqrt{x}y^2(x), \quad y(0) = 0. \tag{35}$$

The exact solution of the problem is $y(x) = x^2$. Applying the differential quadrature method with $N = 2$, we approximate $D^{\frac{1}{2}}y(x)$ as

$$\begin{bmatrix} y^{(\frac{1}{2})}(x_0) \\ y^{(\frac{1}{2})}(x_1) \\ y^{(\frac{1}{2})}(x_2) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ -1.329808 & 1.063846 & 0.265962 \\ -0.376127 & -1.504506 & 1.880632 \end{bmatrix} \begin{bmatrix} y(x_0) \\ y(x_1) \\ y(x_2) \end{bmatrix}, \tag{36}$$

where $x_0 = 0, x_1 = 1/2$, and $x_2 = 1$. Therefore, using (18) and (36), we obtain

$$\begin{bmatrix} 1.063846 & 0.265962 \\ -1.504506 & 1.880632 \end{bmatrix} \begin{bmatrix} y(x_1) \\ y(x_2) \end{bmatrix} = \begin{bmatrix} 0.237729 \\ -0.495495 \end{bmatrix} + \begin{bmatrix} y(x_1) \\ y(x_2) \end{bmatrix} + \begin{bmatrix} \sqrt{1/2}y^2(x_1) \\ y^2(x_2) \end{bmatrix}. \tag{37}$$

Finally, by solving (37) we get

$$y(x_1) = 0.2500, \qquad y(x_2) = 1.0000.$$

Then, using (20), we have $y(x) = x^2$, which is the exact solution.

Example 2 In this example, we consider the equation

$$D^\alpha y = 1 - y^2, \quad 0 < \alpha \le 1,$$

subject to the initial condition $y(0) = 0$. In general, the exact solution of the problem is not known. The exact solution for $\alpha = 1$ is

$$y(x) = \frac{e^{2x} - 1}{e^{2x} + 1}.$$

The problem is considered in [19–21]. We applied the differential quadrature method to solving the problem with $N = 4, 8, 12$ and various values of α. The numerical solutions obtained by the present method and some other numerical methods such as the wavelet method [19, 20] and the artificial neural networks [21] are given in Tables 1, 2, and 3.

Table 1 Comparison of the numerical solutions with the other methods for $\alpha = 1/4$ of Example 2

x	Present method $N = 8$	Method in [21]	Adams method in [21]
0.1	0.526735	0.458224	0.487151
0.2	0.515880	0.532957	0.540879
0.3	0.560879	0.563197	0.571773
0.4	0.608031	0.588243	0.593261
0.5	0.612075	0.606988	0.609616
0.6	0.611336	0.620195	0.622749
0.7	0.631897	0.630670	0.633677
0.8	0.648453	0.640302	0.643005
0.9	0.647655	0.649310	0.651121
1.0	0.657946	0.656773	0.658290

Table 2 Comparison of the numerical solutions with the other methods for $\alpha = 2/4$ of Example 2

x	Present method $N = 8$	Method in [19] $k = 6, m = 2$	Method in [21]
0.1	0.330148	0.330159	0.303073
0.2	0.429343	0.436737	0.423370
0.3	0.498133	0.504842	0.495336
0.4	0.553348	0.553802	0.546941
0.5	0.590546	0.591265	0.585991
0.6	0.617818	0.621026	0.616680
0.7	0.643355	0.645480	0.641789
0.8	0.665723	0.666016	0.662967
0.9	0.682298	0.683560	0.681065
1.0	0.697835		0.696506

Table 3 Comparison of the numerical solutions with the other methods for $\alpha = 3/4$ of Example 2

x	Present method $N = 8$	Method in [19] $k = 6, m = 2$	Method in [21]
0.1	0.185553	0.190108	0.178692
0.2	0.305933	0.309886	0.301614
0.3	0.400771	0.404552	0.397897
0.4	0.478985	0.481638	0.476155
0.5	0.543038	0.545178	0.540568
0.6	0.595789	0.597790	0.594017
0.7	0.640095	0.641801	0.638671
0.8	0.677553	0.678835	0.676201
0.9	0.709031	0.710182	0.707923
1.0	0.735843		0.734913

Clearly, the approximations obtained by the differential quadrature method are in agreement with those obtained with the above-mentioned numerical methods. Table 4 shows the approximate solutions obtained by using the present method for $\alpha = 1$, the Chebyshev wavelet operational matrix of the fractional integration [19] for $k = 6$, $m = 2$, and the Bernoulli wavelet method for $k = 2$, $m = 5$. Also, the numerical results with $N = 8$ and $\alpha = 1/4, 2/4, 3/4, 1$ are plotted in Figure 1. The approximate solutions using the present method are in high agreement with the exact solutions for $\alpha = 1$.

Table 4 Absolute errors for $\alpha = 1$ using $N = 4, 12$ for Example 2

x	$N = 4$	$N = 12$	Method in [20] $k = 1, M = 5$	Method in [19] $k = 6, M = 2$
0.1	2.5206×10^{-4}	8.3141×10^{-11}	1.9995×10^{-5}	1.9000×10^{-5}
0.2	2.4144×10^{-4}	9.1576×10^{-11}	4.1680×10^{-5}	5.9000×10^{-5}
0.3	8.3597×10^{-5}	7.5812×10^{-11}	3.3875×10^{-6}	2.7000×10^{-5}
0.4	7.1204×10^{-5}	1.1151×10^{-10}	3.6962×10^{-5}	3.4000×10^{-5}
0.5	1.2313×10^{-4}	5.5890×10^{-11}	3.4157×10^{-5}	8.9000×10^{-5}
0.6	5.8666×10^{-5}	7.8642×10^{-11}	7.4330×10^{-6}	9.0000×10^{-6}
0.7	5.9751×10^{-5}	6.2746×10^{-11}	3.7223×10^{-5}	6.7000×10^{-5}
0.8	1.3955×10^{-4}	5.3920×10^{-11}	1.5230×10^{-5}	4.3000×10^{-5}
0.9	1.2508×10^{-4}	4.8389×10^{-11}	3.3870×10^{-5}	3.6000×10^{-5}

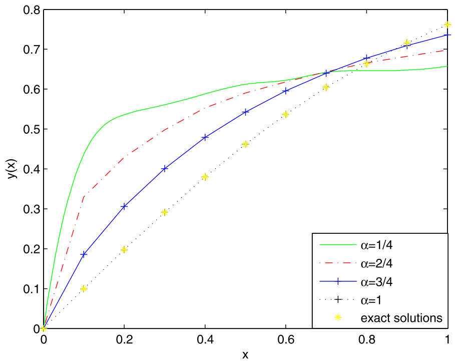

Figure 1 Comparison of $y(x)$ for $N = 8$ and with $\alpha = 1/4, 2/4, 3/4, 1$ for Example 2.

8 Conclusion

A general formulation for the Chebyshev polynomial-based weighting coefficient matrix for approximation of fractional derivatives has been derived. The fractional derivatives are described in the Caputo sense. The matrix is used to get approximate numerical solutions of fractional Riccati differential equations. Our numerical results are compared with the solutions obtained by the wavelet and artificial neural network methods. The solution obtained using the present method shows that this approach can effectively solve the problem.

Acknowledgements
The authors are very grateful to the referees for carefully reading the paper and for their comments and suggestions, which have improved the paper.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
Both authors contributed equally to the writing of this paper. Both authors read and approved the final manuscript.

Publisher's Note

Received: 1 June 2017 Accepted: 20 October 2017 Published online: 17 November 2017

References

1. Podlubny, I: Fractional Differential Equations. Academic Press, San Diego (1999)
2. Amairi, M, Aoun, M, Najar, S, Abdelkrim, MN: A constant enclosure method for validating existence and uniqueness of the solution of an initial value problem for a fractional differential equation. Appl. Math. Comput. **217**(5), 2162-2168 (2010)

3. Deng, J, Ma, L: Existence and uniqueness of solutions of initial value problems for nonlinear fractional differential equations. Appl. Math. Lett. **23**(6), 676-680 (2010)
4. Shawagfeh, NT: Analytical approximate solutions for nonlinear fractional differential equations. Appl. Math. Comput. **131**(2), 517-529 (2002)
5. Hashima, I, Abdulaziz, O, Momani, S: Homotopy analysis method for fractional IVPs. Commun. Nonlinear Sci. Numer. Simul. **14**(3), 674-684 (2009)
6. Diethelm, K, Ford, NJ, Freed, AD: A predictor-corrector approach for the numerical solution of fractional differential equations. Nonlinear Dyn. **29**(1), 3-22 (2002)
7. Diethelm, K, Ford, NJ, Freed, AD: Detailed error analysis for a fractional Adams method. Numer. Algorithms **36**(1), 31-52 (2004)
8. Yang, C, Hou, J: An approximate solution of nonlinear fractional differential equation by Laplace transform and Adomian polynomials. J. Inf. Comput. Sci. **10**(1), 213-222 (2013)
9. Yüzbaşi, Ş: A numerical approximation based on the Bessel functions of first kind for solutions of Riccati type differential difference equations. Comput. Math. Appl. **64**(6), 1691-1705 (2012)
10. Yüzbaşi, Ş: Numerical solution of the Bagley-Torvik equation by the Bessel collocation method. Math. Methods Appl. Sci. **36**(3), 300-312 (2013)
11. Yüzbaşi, Ş: A numerical approximation for Volterra population growth model with fractional order. Appl. Math. Model. **37**(5), 3216-3227 (2013)
12. Bellman, RE, Casti, J: Differential quadrature and long-term integration. J. Math. Anal. Appl. **34**(2), 235-238 (1971)
13. Fung, TC: Solving initial value problems by differential quadrature method part 1: first-order equations. Int. J. Numer. Methods Eng. **50**(6), 1411-1427 (2001)
14. Gil, A, Segura, J, Temme, NM: Numerical Methods for Special Functions. SIAM, Philadelphia (2007)
15. Boyd, JP: Chebyshev and Fourier Spectral Methods, 2nd edn. Dover, New York (1999)
16. Canuto, C, Hussaini, MY, Quarteroni, A, Zang, TA: Spectral Methods: Fundamentals in Single Domains. Springer, Berlin (2006)
17. Ghoreishi, F, Mokhtary, P: Spectral collocation method for multi-order fractional differential equations. Int. J. Comput. Methods **11**(05), 1350072 (2014)
18. Mokhtary, P, Ghoreishi, F: The L^2-convergence of the Legendre spectral tau matrix formulation for nonlinear fractional integro differential equations. Numer. Algorithms **58**(4), 475-496 (2011)
19. Wang, Y, Fan, Q: The second kind Chebyshev wavelet method for solving fractional differential equations. Appl. Math. Comput. **218**(17), 8592-8601 (2012)
20. Keshavarza, E, Ordokhania, Y, Razzaghi, M: Bernoulli wavelet operational matrix of fractional order integration and its applications in solving the fractional order differential equations. Appl. Math. Model. **38**(24), 6038-6051 (2014)
21. Raja, MAZ, Manzar, MA, Samar, R: An efficient computational intelligence approach for solving fractional order Riccati equations using ANN and SQP. Appl. Math. Model. **39**(10-11), 3075-3093 (2015)