



# Numerical solution of nonlinear delay differential equations of fractional variable-order using a novel shifted Jacobi operational matrix

H. R. Khodabandehlo<sup>1</sup> · E. Shivanian<sup>1</sup> · S. Abbasbandy<sup>1</sup>

Received: 25 January 2021 / Accepted: 6 May 2021 / Published online: 18 May 2021  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

## Abstract

This paper presents the generalized nonlinear delay differential equations of fractional variable-order. In this article, a novel shifted Jacobi operational matrix technique is introduced for solving a class of multi-terms variable-order fractional delay differential equations via reducing the main problem to an algebraic system of equations that can be solved numerically. The suggested technique is successfully developed for the aforementioned problem. Comprehensive numerical experiments are presented to demonstrate the efficiency, generality, accuracy of proposed scheme and the flexibility of this method. The numerical results compared it with other existing methods such as fractional Adams method (FAM), new predictor–corrector method (NPCM), a new approach, Adams–Bashforth–Moulton algorithm and L1 predictor–corrector method (L1-PCM). Comparing the results of these methods as well as comparing the current method (NSJOM) with the exact solution, indicating the efficiency and validity of this method. Note that the procedure is easy to implement and this technique will be considered as a generalization of many numerical schemes. Furthermore, the error and its bound are estimated.

**Keywords** Nonlinear delay differential equations of fractional variable-order · Jacobi polynomials · Operational matrix technique · Caputo differential operator

**Mathematics subject classifications** 65M99

## 1 Introduction

The analysis and applications of fractional calculus are an active and the fastest growing region for research in the last three decades. It has currently become an important tool owing to its wide applications in various scientific disciplines, such as physics, regular variation in thermodynamics, blood flow phenomenons, biophysics, electro-dynamics of complex medium, capacitor theory, chemistry, polymer rheology, dynamical systems, fitting of experimental data, etc. ([1–3, 7] and references therein). The increasing development of appropriate and efficient method to solve *FDEs* has aroused more interest of researchers in this field. Solving the

nonlinear *FDEs* is relatively difficult compared to its linear type. In this regard, analytical methods such as, homotopy perturbation method (*HPM*), homotopy analysis method (*HAM*), new iterative methods and Adomian decomposition, have been used in the recent literature widely [8, 9, 11–14, 53]. On the other hands, some researchers such as Diethelm et al [2, 16] have developed standard numerical methods, such as Adams–Bashforth methods have been used for numerical integration of *FDEs*. Lakestani et al [54] have presented The construction of operational matrix of fractional derivatives using B-spline functions, Application of Taylor series in obtaining the orthogonal operational matrix is proposed by Eslahchi and Dehghan [56], Kayedi-Bardeh et al have provided A method for obtaining the operational matrix of the fractional Jacobi functions and application [57].

Inclusion of delay in the differential equations of fractional order creates new perspectives especially in the field of bioengineering [6], because in bioengineering, the

✉ E. Shivanian  
shivanian@sci.ikiu.ac.ir

<sup>1</sup> Department of Applied Mathematics, Faculty of Science, Imam Khomeini International University, Qazvin 34149, Iran

understanding of the dynamics that occur in biological tissues, is improved by fractional derivatives [4, 6].

In mathematics, delay differential equations (DDEs) are a type of differential equation in which the derivative of the unknown function at a certain time is given in terms of the values of the function at previous times. The DDEs are also called time-delay systems, systems with aftereffect or deed-time, hereditary systems, equations with deviating argument, or differential–difference equations [17].

Fractional delay differential equations differ from ordinary differential equations in that the derivative at any time depends on the solution (and in the case of neutral equations on the derivative) at prior times. Many events in the natural world can be modeled to form of fractional order of delay differential equations [7]. The fraction order of differential equations have many application in various scientific disciplines by modeling of various problem such as economy, electrodynamics, biology, control, finance, chemistry, physics and so on, for more detail, we refer the interested reader to [17–23].

In recent years, Margado et al in [24] analyzed numerical solution and approximated it for FDDEs. Cermak et al in [25], analyzed stability regions of FDDEs system. Stability for FDDEs systems by means of Grünwalds approach is analyzed by Lazarovic and Spansic [26]. Abbaszadeh and Dehghan have presented Numerical and analytical investigations for neutral delay fractional damped diffusion-wave equation based on the stabilized interpolating element free Galerkin (IEFG) method [55], Daftardar-Gejji et al in [12] have proposed a new predictor–corrector method and based on the realizer developed new iteration method presented by Daftardar-Gejji and Jafari [27], for solving FDEs numerically. Bhalekar and Daftardar-Gejji proposed A predictor–corrector scheme to solve non-linear fractional-order delay differential equations in [10]. In [5], the author generalized the Adams–Bashforth–Moulton algorithm introduced in [2, 16, 28] to the delay FDEs. Varsha et al [6], have presented a new approach to solve non-linear fractional- order differential equations involving delay. Ghasemi et al [17], have employed Reproducing kernel Hilbert Space method to solve nonlinear delay differential equation of fractional order. Jhing and Daftardar-Gejji have presented a new numerical method to solve FDDEs [4].

Moreover, the spectral methods that basically depends on a set of orthogonal polynomials, are used to solve the differential equations of fractional-order. One of the most famous of them, is the classical Jacobi polynomials that shown as follows:

$$P_n^{(\alpha,\beta)}(x) \quad (n \geq 0, \alpha > -1, \beta > -1).$$

These polynomials have been used extensively in mathematical analysis and practical applications because it has

the advantages of gaining the numerical solutions in  $\alpha$  and  $\beta$  parameters. Thus, it would be beneficial to carry out a systematic study via Jacobi polynomials with general indexes  $\alpha$  and  $\beta$  and this clearly considered one of the aims and the novelty of the time interval  $t \in [0, I]$  [15]. Furthermore, recently interest of researchers has increased in this field (field of variable fractional differential equations) [29, 30]. So, many methods are used to find the numerical solution of these equations [31–33].

Now, the aim of this paper is to generalize the orthogonal polynomials in the base of solution. In fact, this technique is introduced in [15] and we present a new shifted Jacobi operational matrix for the fractional derivative to solve the multi-terms variable-order *FDDEs* which as follows:

$$\begin{aligned} & \sum_{j=1}^n \alpha_j D^{\eta_j(t)} w(t) + \alpha_{n+1} w(t - \tau) \\ & = F(t, w(t), D^{\eta_1(t)} w(t), \\ & \quad D^{\eta_2(t)} w(t), \dots, D^{\eta_n(t)} w(t), w(t - \tau)), \quad 0 \leq t \leq T, \\ & w(t) = g(t), \quad t \in [-\tau, 0], \\ & w(0) = w_0, \end{aligned} \tag{1}$$

where  $\alpha_j \in \mathbb{R} (j = 1, 2, \dots, n + 1), \alpha_{n+1} \neq 0, 0 < T$ . and  $D^{\eta_j(t)} w(t) (j = 1, 2, \dots, n)$  are the derivatives of variable-order fractional in the Caputo sense.

**Note 1** If  $\eta_j(t) (j = 1, 2, \dots, n)$  are constants, then equation (1) will be as follows:

$$\begin{aligned} & \sum_{j=1}^n \alpha_j D^{\eta_j} w(t) + \alpha_{n+1} w(t - \tau) \\ & = F(t, w(t), D^{\eta_1} w(t), \\ & \quad D^{\eta_2} w(t), \dots, D^{\eta_n} w(t), w(t - \tau)), \quad 0 \leq t \leq T, \\ & w(t) = g(t), \quad t \in [-\tau, 0], \\ & w(0) = w_0, \end{aligned} \tag{2}$$

Also note that: we can use many polynomials such as Legendre polynomials, Gegenbauer polynomials, all Chebyshev polynomials, Lucas polynomials, Vieta-Lucas polynomials, and Fibonacci polynomials in our new suggestion technique.

The numerical results obtained for the mentioned equation in this study reveals that the present method is of highly accuracy. By focusing on numerical experiments gained by this method with other available methods, and comparing them, we can find that the proposed scheme capable of solving the variable-order fractional delay differential equation, playing role of a powerful effective and practical numerical technique.

## 2 Fundamentals and preliminaries

In the first part of this section, we review some of the basic and most important properties of fractional calculus theory. Then recall some important properties of the Jacobi polynomials which help us for developing the suggested technique. So we refuse to include duplicate concepts in other related articles and unnecessary content but for more detail, we refer the interested reader to [34, 37, 38].

### 2.1 The fractional derivative and integral

There are several used definitions for fractional derivatives but the three most usual are Riemann–Liouville, Grünwald-Letincov and Caputo definitions. This article is based on Caputo definition because, as well as know, only the Caputo sense has the same form as integer-order differential equations in initial conditions.

**Definition 2.1** The left- and right-sided Caputo fractional derivatives of order  $\eta$  ( $n - 1 < \eta \leq n$ ) are defined as

$$\begin{aligned} D_+^\eta w(t) &= \frac{1}{\Gamma(n - \eta)} \int_0^t \frac{U^{(n)}(\tau)}{(t - \tau)^{\eta - n + 1}} d\tau, \\ D_-^\eta w(t) &= \frac{(-1)^\eta}{\Gamma(n - \eta)} \int_t^T \frac{U^{(n)}(\tau)}{(\tau - t)^{\eta - n + 1}} d\tau. \end{aligned} \tag{3}$$

that

$$D_+^\eta t^j = \begin{cases} 0, & \text{for } j \in M_0 \text{ and } j < [\eta], \\ \frac{\Gamma(j + 1)}{\Gamma(j - \eta + 1)} t^{j - \eta}, & \text{for } j \in M_0 \text{ and } j > [\eta]. \end{cases} \tag{4}$$

and

$$\begin{aligned} D_-^\eta (T - t)^j &= \begin{cases} 0, & \text{for } j \in M_0 \text{ and } j < [\eta], \\ \frac{(-1)^j \Gamma(j + 1)}{\Gamma(j - \eta + 1)} (T - t)^{j - \eta}, & \text{for } j \in M_0 \text{ and } j > [\eta]. \end{cases} \end{aligned} \tag{5}$$

where  $[\cdot]$  is the ceiling function and  $M_0 = \{0, 1, 2, \dots\}$ .

Also

$$D_\pm^\eta (\gamma \varphi(t) + \delta \phi(t)) = \gamma D_\pm^\eta (\varphi(t)) + \delta D_\pm^\eta (\phi(t)).$$

where  $\gamma$  and  $\delta$  are constants.

---


$$P_i^{(\alpha, \beta)}(t) = \sum_{k=0}^i \frac{\Gamma(\alpha + i + 1) \Gamma(\alpha + i + 1 + \beta + k)}{\Gamma(\alpha + \beta + i + 1) \Gamma(\alpha + 1 + k) \Gamma(k + 1) \Gamma(i - k + 1)} \left(\frac{t - 1}{2}\right)^k, \tag{12}$$


---

**Definition 2.2** The Caputo fractional derivatives of variable-order  $\eta(t)$  for  $w(t) \in C^m[0, T]$  are given respectively as [30, 35]:

$$\begin{aligned} D^{\eta(t)} w(t) &= \frac{1}{\Gamma(1 - \eta(t))} \int_{0^+}^t \frac{w'(\tau)}{(t - \tau)^{\eta(t)}} d\tau \\ &+ \frac{w(0^+) - w(0^-)}{\Gamma(1 - \eta(t))} t^{-\eta(t)}. \end{aligned} \tag{6}$$

At the beginning time and for  $0 < \eta(t) < 1$ , we have:

$$D^{\eta(t)} w(t) = \frac{1}{\Gamma(1 - \eta(t))} \int_{0^+}^t \frac{w'(\tau)}{(t - \tau)^{\eta(t)}} d\tau. \tag{7}$$

also, if  $a$  and  $b$  are constant then

$$D^{\eta(t)} (a w_1(t) + b w_2(t)) = a D^{\eta(t)} w_1(t) + b D^{\eta(t)} w_2(t). \tag{8}$$

According to Eq. (6), we have:

$$D^{\eta(t)} C = 0, \quad C \text{ is a constant.} \tag{9}$$

On the other hand

$$D^{\eta(t)} t^k = \begin{cases} 0, & \text{for } k = 0, \\ \frac{\Gamma(k + 1)}{\Gamma(k + 1 - \eta(t))} t^{k - \eta(t)} & \text{for } k = 1, 2, \dots \end{cases} \tag{10}$$

### 2.2 Shifted Jacobi polynomials and their properties

Denote  $P_n^{(\alpha, \beta)}(z)$ ;  $\alpha > -1, \beta > -1$  as the  $n$ -th order Jacobi polynomial in  $z$  defined on  $[-1, 1]$ .

As all classic orthogonal polynomials,  $P_n^{(\alpha, \beta)}(z)$  form an orthogonal system with respect to weight function  $\omega^{(\alpha, \beta)}(z) = (1 + z)^\alpha (1 - z)^\beta$ , in other words [34]:

$$\int_{-1}^1 P_i^{(\alpha, \beta)}(z) P_j^{(\alpha, \beta)}(z) \omega^{(\alpha, \beta)}(z) dz = h_j^{(\alpha, \beta)} \delta_{i,j}, \tag{11}$$

where  $\delta_{i,j}$  is the Kronecker function and

$$h_j^{(\alpha, \beta)} = \frac{2^{\alpha + \beta + 1} \Gamma(j + \alpha + 1) \Gamma(j + \beta + 1)}{(2j + \alpha + \beta + 1) j! \Gamma(j + \beta + \alpha + 1)},$$

also the  $i$ -th order Jacobi polynomial has the following analytical form [15]

If we implement the change of variable  $z = (\frac{2t}{T} - 1)$ , then we can use the polynomial of Eq. (12) on the interval  $t \in [0, T]$ . Therefore, we have so-called the shifted Jacobi polynomials  $P_n^{(\alpha,\beta)}(\frac{2t}{T} - 1)$  which denoted by  $P_{T,i}^{(\alpha,\beta)}(t)$ . Then  $P_{T,i}^{(\alpha,\beta)}(t)$  form an orthogonal system with respect to weight function  $\omega_T^{(\alpha,\beta)}(t) = t^\alpha (T - t)^\beta$  in the interval  $[0, T]$  with follow orthogonality trait:

$$\int_0^T P_{T,i}^{(\alpha,\beta)}(t) P_{T,j}^{(\alpha,\beta)}(t) \omega_T^{(\alpha,\beta)} dt = h_{T,j}^{(\alpha,\beta)} \delta_{i,j}, \tag{13}$$

where  $\delta_{i,j}$  is the Kronecker function and

$$h_{T,j}^{(\alpha,\beta)} = \left(\frac{T}{2}\right)^{\alpha+\beta+1} h_j^{(\alpha,\beta)},$$

also the  $i$ -th order Shifted Jacobi polynomial has the following analytical form [15]

$$P_{T,i}^{(\alpha,\beta)}(t) = \sum_{k=0}^i (-1)^{i-k} \frac{\Gamma(\alpha+i+1)\Gamma(\alpha+\beta+k+i+1)}{\Gamma(\alpha+\beta+i+1)\Gamma(\alpha+1+k)\Gamma(k+1)\Gamma(i-k+1)T^k} t^k, \\ = \sum_{k=0}^i \frac{\Gamma(\beta+i+1)\Gamma(\alpha+\beta+k+i+1)}{\Gamma(\alpha+\beta+i+1)\Gamma(\beta+1+k)\Gamma(k+1)\Gamma(i-k+1)T^k} (T-t)^k. \tag{14}$$

And in the endpoint values are given as

$$P_{T,i}^{(\alpha,\beta)}(0) = (-1)^i \frac{\Gamma(\alpha+i+1)}{\Gamma(\alpha+1)\Gamma(k+1)}, \\ P_{T,i}^{(\alpha,\beta)}(T) = \frac{\Gamma(\beta+i+1)}{\Gamma(\beta+1)\Gamma(k+1)}.$$

Also,  $P_{T,i}^{(\alpha,\beta)}(t)$  can be obtained via the recurrence relation, we refer the interested reader to [15].

**Note 2** Furthermore, since the shifted Jacobi polynomials includes unlimited number of orthogonal polynomials such as the shifted Gegenbauer polynomials  $G_{T,i}^{(\alpha,\beta)}(t)$ , the shifted Legendre polynomials  $L_{T,i}^{(\alpha,\beta)}(t)$ , and the shifted Chebyshev polynomials of the first, second, third and fourth kinds  $T_{T,i}^{(\alpha,\beta)}(t)$ ,  $U_{T,i}^{(\alpha,\beta)}(t)$ ,  $V_{T,i}^{(\alpha,\beta)}(t)$  and  $W_{T,i}^{(\alpha,\beta)}(t)$ , respectively. All these orthogonal polynomials are contacted with  $P_{T,i}^{(\alpha,\beta)}(t)$  through the relations:

$$L_{T,i}^{(\alpha,\beta)}(t) = P_{T,i}^{(0,0)}(t), \\ G_{T,i}^{(\alpha,\beta)}(t) = \frac{\Gamma(i+1)\Gamma(\alpha+\frac{1}{2})}{\Gamma(\alpha+\frac{1}{2}+i)} P_{T,i}^{(\alpha-\frac{1}{2},\beta-\frac{1}{2})}(t), \\ T_{T,i}^{(\alpha,\beta)}(t) = \frac{\Gamma(i+1)\Gamma(\frac{1}{2})}{\Gamma(\frac{1}{2}+i)} P_{T,i}^{(\frac{1}{2},-\frac{1}{2})}(t), \\ U_{T,i}^{(\alpha,\beta)}(t) = \frac{\Gamma(i+2)\Gamma(\frac{1}{2})}{2\Gamma(\frac{3}{2}+i)} P_{T,i}^{(\frac{1}{2},\frac{1}{2})}(t), \\ V_{T,i}^{(\alpha,\beta)}(t) = \frac{2^{2i}(\Gamma(i+1))^2}{\Gamma(2i+1)} P_{T,i}^{(-\frac{1}{2},\frac{1}{2})}(t), \\ W_{T,i}^{(\alpha,\beta)}(t) = \frac{2^{2i}(\Gamma(i+1))^2}{\Gamma(2i+1)} P_{T,i}^{(\frac{1}{2},-\frac{1}{2})}(t).$$

### 3 Function approximation by shifted Jacobi polynomials

The function  $w(t)$ , square integrable with respect to  $\omega_T^{(\alpha,\beta)}(t)$  in  $[0, T]$ , can be expanded as the following expression [15, 34]:

$$w(t) = \sum_{i=0}^{\infty} a_i P_{T,i}^{(\alpha,\beta)}(t), \tag{15}$$

where  $a_i$  (the coefficients of the series) are obtained by

$$a_i = \frac{1}{h_{T,i}^{(\alpha,\beta)}} \int_0^T \omega_T^{(\alpha,\beta)} w(t) P_{T,i}^{(\alpha,\beta)}(t) dt, \quad i = 0, 1, \dots. \tag{16}$$

So, we can estimate the approximate solution by taking  $(N + 1)$ -terms of the series in Eq. (15) and we will have

$$w(t) \simeq w_N(t) = \sum_{i=0}^N a_i P_{T,i}^{(\alpha,\beta)}(t) = A^T \Phi_{T,N}(t), \tag{17}$$

where  $A = [a_0, a_1, \dots, a_N]^T$ , and  $\Phi_{T,N}(t) = [P_{T,0}^{(\alpha,\beta)}(t), P_{T,1}^{(\alpha,\beta)}(t), \dots, P_{T,N}^{(\alpha,\beta)}(t)]^T$ . Here, we suppose that

$$S(t) = [1, t, t^2, t^3, \dots, t^N]^T. \tag{18}$$

By Eq. (17), the vector  $\Phi_{T,N}(t)$  can be presented as

$$\Phi_{T,N}(t) = B_{(\alpha,\beta)}S(t). \tag{19}$$

where  $B_{(\alpha,\beta)}$  is a square matrix of order  $(N + 1) \times (N + 1)$  that given as follows

$$b_{i+1,j+1} = \begin{cases} (-1)^{i-j} \frac{(\alpha + i)!(\alpha + \beta + j + i)!}{(\alpha + \beta + i)!(\alpha + j)!(j!)(i - j)!T^j}, & i \geq j, \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

for  $0 \leq i, j \leq N$ .

For example, if  $N = 4, \alpha = \beta = 0$ , then  $B$  as follows

$$B_{(0,0)} = \frac{1}{T^i} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 1 & -6 & 6 & 0 & 0 \\ -1 & 12 & -30 & 20 & 0 \\ 1 & -20 & 90 & -140 & 70 \end{bmatrix}. \tag{21}$$

Hence, using Eq. (19), we get

$$S(t) = B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t). \tag{22}$$

**Note 3** Note that, we obtain this square matrix  $B$  for all other orthogonal polynomials as well. For example, if  $N = 4, \alpha = \frac{1}{2}, \beta = \frac{-1}{2}$  then the square matrix  $B$  for the fourth kind shifted Chebyshev polynomials as follows

$$B\left(\frac{1}{2}, \frac{-1}{2}\right) = \frac{1}{T^i} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 4 & 0 & 0 & 0 \\ 1 & -12 & 16 & 0 & 0 \\ -1 & 24 & -80 & 64 & 0 \\ 1 & -40 & 240 & -448 & 256 \end{bmatrix}. \tag{23}$$

### 4 Shifted Jacobi polynomials operational matrix (SJOM)

Operational matrices, which are used in different areas of numerical analysis and they are especially important to solve a variety of problems in different fields such as integral equations, differential equations, integro-differential equations, ordinary and partial fractional differential equations and etc [31, 34–36, 39–47]. In this section, we investigate the (SJOM) of fractional variable-order to support the numerical solution of Eq. (1). Therefore, we convert the problem into the system of algebraic of equations which solved numerically in collocation points.

At first,  $D^{\eta_i(t)}\Phi_{T,N}(t)$ , ( $i = 1, 2, \dots, n$ ) can be deduced as the following:

since  $\Phi_{T,N}(t) = B_{(\alpha,\beta)}S(t)$ , then we have

$$\begin{aligned} D^{\eta_i(t)}\Phi_{T,N}(t) &= D^{\eta_i(t)}(B_{(\alpha,\beta)}S(t)) \\ &= B_{(\alpha,\beta)}D^{\eta_i(t)}[1, t, \dots, t^N]^T, \\ & \quad i = 1, 2, \dots, n. \end{aligned} \tag{24}$$

Combining Eqs. (10) and (24), it gives

$$\begin{aligned} D^{\eta_i(t)}\Phi_{T,N}(t) &= B_{(\alpha,\beta)}D^{\eta_i(t)}(S(t)) \\ &= B_{(\alpha,\beta)} \left[ 0, \frac{\Gamma(2)t^{1-\eta_i(t)}}{\Gamma(2-\eta_i(t))}, \dots, \frac{\Gamma(N+1)t^{N-\eta_i(t)}}{\Gamma(N+1-\eta_i(t))} \right]^T \\ &= B_{(\alpha,\beta)} \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & \frac{\Gamma(2)t^{-\eta_i(t)}}{\Gamma(2-\eta_i(t))} & 0 & \dots & 0 \\ 0 & 0 & \frac{\Gamma(3)t^{-\eta_i(t)}}{\Gamma(3-\eta_i(t))} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{\Gamma(N)t^{-\eta_i(t)}}{\Gamma(N-\eta_i(t))} \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ \vdots \\ t^N \end{bmatrix} \\ &= B_{(\alpha,\beta)}Q_i(t)S(t), \quad i = 1, 2, \dots, n. \end{aligned} \tag{25}$$

where

$$Q_i(t) = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & \frac{\Gamma(2)t^{-\eta_i(t)}}{\Gamma(2-\eta_i(t))} & 0 & \dots & 0 \\ 0 & 0 & \frac{\Gamma(3)t^{-\eta_i(t)}}{\Gamma(3-\eta_i(t))} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{\Gamma(N)t^{-\eta_i(t)}}{\Gamma(N-\eta_i(t))} \end{bmatrix}, \quad i = 1, 2, \dots, n. \tag{26}$$

Using Eq. (22), then

$$D^{n_i(t)} \Phi_{T,N}(t) = B_{(\alpha,\beta)} Q_i(t) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t), \quad i = 1, 2, \dots, n. \tag{27}$$

The operational matrix of  $D^{n_i(t)} \Phi_{T,N}(t)$ , ( $i = 1, 2, \dots, n$ .) is  $B_{(\alpha,\beta)} Q_i(t) B_{(\alpha,\beta)}^{-1}$ .

Now, we can estimate the variable-order fractional of the approximated function that obtained in Eq. (17) as the following

$$D^{n_i(t)} w(t) \simeq D^{n_i(t)} (A^T \Phi_{T,N}(t)) = A^T D^{n_i(t)} \Phi_{T,N}(t) = A^T B_{(\alpha,\beta)} Q_i(t) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t), \quad i = 1, 2, \dots, n. \tag{28}$$

By using Eq. (28), hence the Eq. (1) is turned into

$$\begin{aligned} & \sum_{i=1}^n \alpha_i (A^T B_{(\alpha,\beta)} Q_i(t) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t)) \\ & + \alpha_{n+1} A^T \Phi_{T,N}(t - \tau) \\ & = F(t, A^T \Phi_{T,N}(t), (A^T B_{(\alpha,\beta)} Q_1(t) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t)), \\ & (A^T B_{(\alpha,\beta)} Q_2(t) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t)), \dots, \\ & (A^T B_{(\alpha,\beta)} Q_n(t) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t)), \\ & A^T \Phi_{T,N}(t - \tau)), \quad 0 \leq t \leq T, \end{aligned} \tag{29}$$

Finally, we use  $t_j$  ( $j = 0, 1, 2, \dots, m$ .) where they are the roots of  $P_{T,m+1}^{(\alpha,\beta)}(t)$ . Then Eq. (29) can be converted into the following algebraic system

$$\begin{aligned} & \sum_{i=1}^n \alpha_i (A^T B_{(\alpha,\beta)} Q_i(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)) \\ & + \alpha_{n+1} A^T \Phi_{T,N}(t_j - \tau) \\ & = F(t_j, A^T \Phi_{T,N}(t_j), (A^T B_{(\alpha,\beta)} Q_1(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), \\ & (A^T B_{(\alpha,\beta)} Q_2(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), \dots, \\ & (A^T B_{(\alpha,\beta)} Q_n(t_j) B_{(\alpha,\beta)}^{-1} \Phi_{T,N}(t_j)), \\ & A^T \Phi_{T,N}(t_j - \tau)), \quad j = 0, 1, 2, \dots, m. \end{aligned} \tag{30}$$

So, the system in Eq. (30) can be solved numerically for determining the unknown vector  $A$ . Therefore, the numerical solution that presented in Eq. (17) can be obtained.

### 5 Error analysis

In this part of paper, we estimate an upper bound of the absolute errors via using the Lagrange interpolation polynomials. Also, by using the current method (*NSJOM*) with error approximation and the residual correction method [48, 49], an efficient error approximation will be obtained for the variable-order fractional differential equations.

### 5.1 Error bound

Now, our aim is to gain an analytic expression of error norm for the best approximation of a smooth function  $w(t) \in [0, T]$  via its expansion in terms of Jacobi polynomials. Let

$$\prod_N^{\alpha,\beta} = \text{span} \left\{ P_{T,i}^{(\alpha,\beta)}(t), \quad i = 0, 1, 2, \dots, N \right\}.$$

In this part of the discussion, we always assume that  $w_N(t) \in \prod_N^{\alpha,\beta}$  is the best approximation of  $w(t)$ , then by definition of the best approximation, we have

$$\forall v_N(t) \in \prod_N^{\alpha,\beta} \quad \|w(t) - w_N(t)\|_\infty \leq \|w(t) - v_N(t)\|_\infty. \tag{31}$$

It's obvious that the above inequality is also true if  $v_N(t)$  be the interpolating polynomials at node points  $t_i$  ( $i = 0, 1, \dots, m$ ), where  $t_i$  are the roots of  $P_{T,m+1}^{(\alpha,\beta)}(t)$ . Then by the Lagrange interpolation polynomials formula and its error formula, we have

$$w(t) - v_N(t) = \frac{w^{(N+1)}(\xi)}{(N+1)!} \prod_{j=0}^N (t - t_j), \tag{32}$$

where  $\xi \in [0, T]$ , and hence we obtain that

$$\begin{aligned} \|w(t) - v_N(t)\|_\infty & \leq \max_{0 \leq t \leq T} |w^{(N+1)}(\xi)| \\ & \frac{\| \prod_{j=0}^N (t - t_j) \|_\infty}{(N+1)!}. \end{aligned} \tag{33}$$

We note that  $w(t)$  is a smooth function on  $[0, T]$ , therefore, there exist a constant  $C_1$ , such that

$$\max_{0 \leq t \leq T} |w^{(N+1)}(\xi)| \leq C_1. \tag{34}$$

By minimizing the factor  $\| \prod_{j=0}^N (t - t_j) \|_\infty$  as follows, we will have:

If use the one-to-one mapping  $t = \frac{T}{2}(z + 1)$  between the interval  $[-1, 1]$  and  $[0, T]$  to deduce that [34, 50]

$$\begin{aligned} & \min_{0 \leq t_j \leq T} \max_{0 \leq t \leq T} \left| \prod_{j=0}^N (t - t_j) \right| \\ & = \min_{-1 \leq z_j \leq 1} \max_{-1 \leq z \leq 1} \left| \prod_{j=0}^N \frac{T}{2} (z - z_j) \right| \\ & = \left(\frac{T}{2}\right)^{N+1} \min_{-1 \leq z_j \leq 1} \max_{-1 \leq z \leq 1} \left| \prod_{j=0}^N (z - z_j) \right| \\ & = \left(\frac{T}{2}\right)^{N+1} \min_{-1 \leq z_j \leq 1} \max_{-1 \leq z \leq 1} \left| \frac{P_{N+1}^{(\alpha,\beta)}(z)}{\lambda_N^{(\alpha,\beta)}} \right|, \end{aligned} \tag{35}$$



where  $\lambda_N^{(\alpha,\beta)} = \frac{\Gamma(2N + \alpha + \beta + 1)}{2^N N! \Gamma(N + \alpha + \beta + 1)}$  is the leading coefficient of  $P_{N+1}^{(\alpha,\beta)}(z)$  and  $z_j$  are the roots of  $P_{N+1}^{(\alpha,\beta)}(z)$ . It is clear that

$$\max_{-1 \leq z \leq 1} |P_{N+1}^{(\alpha,\beta)}(z)| = P_{N+1}^{(\alpha,\beta)}(1) = \frac{\Gamma(\beta + N + 2)}{\Gamma(\beta + 1)(N + 1)!},$$

Using Eqs. (34) and (35), gives the following result

$$\|w(t) - w_N(t)\|_\infty \leq C_1 \frac{(\frac{T}{2})^{N+1} \Gamma(\beta + N + 2)}{\lambda_N^{(\alpha,\beta)} ((N + 1)!)^2 \Gamma(\beta + 1)}. \tag{36}$$

Then, we estimate an upper bound for absolute error of the exact and approximate solutions.

### 5.2 Error function estimation

In this subsection, we have introduced the error estimation based on the residual error function for the presented technique and the approximate solution (17) is refined via the residual correction scheme. The residual error estimation was used for the error estimation of some methods for different equations [36, 49, 51, 52].

At first, we denote  $e_N(t) = w_N(t) - w(t)$  as the error function of the NSJOM method approximation  $w_N(t)$  to  $w(t)$ , where  $w(t)$  is the exact solution of Eq. (1).

Therefore,  $w_N(t)$  satisfies the following equation

$$\begin{aligned} &\sum_{j=1}^n \alpha_j D^{\eta_j(t)} w_N(t) + \alpha_{n+1} w_N(t - \tau) \\ &= F(t, w_N(t), D^{\eta_1(t)} w_N(t), \\ &\quad D^{\eta_2(t)} w_N(t), \dots, D^{\eta_n(t)} w_N(t), w_N(t - \tau)) + R_N(t), \\ &\quad 0 \leq t \leq T, \\ &\quad w_N(0) = w_0, \end{aligned} \tag{37}$$

where  $R_N(t)$  is the residual function of Eq. (1), which is estimated by replacing the  $w_N(t)$  with  $w(t)$  in Eq. (1).

By subtract Eq. (1) from Eq. (37), the error problem is constructed as follows:

$$\begin{aligned} &\sum_{j=1}^n \alpha_j D^{\eta_j(t)} e_N(t) + \alpha_{n+1} e_N(t - \tau) \\ &= R_N(t) + R_{N,F}(t), \quad 0 \leq t \leq T, \\ &\quad e_N(0) = 0, \end{aligned} \tag{38}$$

where

$$\begin{aligned} R_{N,F}(t) &= F(t, w_N(t), D^{\eta_1(t)} w_N(t), \\ &\quad D^{\eta_2(t)} w_N(t), \dots, D^{\eta_n(t)} w_N(t), w_N(t - \tau)) \\ &\quad - F(t, w(t), D^{\eta_1(t)} w(t), \\ &\quad D^{\eta_2(t)} w(t), \dots, D^{\eta_n(t)} w(t), w(t - \tau)). \end{aligned} \tag{39}$$

Thus, the (38) can be solved in the same way as in the previous section and we obtain the following approximation to  $e_N(t)$ .

$$e_N(t) = \sum_{i=0}^N d_i P_{T,i}^{(\alpha,\beta)}(t) = D^T \Phi_{T,N}(t), \tag{40}$$

Then the maximum absolute error can be obtained approximately by

$$E_N(t) = \max\{e_N(t), 0 \leq t \leq T\}. \tag{41}$$

**Note 4** Note that if the exact solution of the problem (1) is unknown, in real practical experiment, we have trouble with computing  $R_{N,F}$ . But, the replacement strategy is to be approximated by its bound, say  $D_1 |e_N(t)| + D_2 |e_N(t - \tau)|$  with  $D_1$  and  $D_2$  as positive constants. In fact, it is possible by supposing that the nonlinear term  $F$  satisfies Lipschitz condition with respect to its all arguments.

The above estimation of error (41), depends on the convergence rates of expansions in Jacobi polynomial. Thus, it provides reasonable convergence rates in temporal discretizations [34, 52].

## 6 Numerical experiments

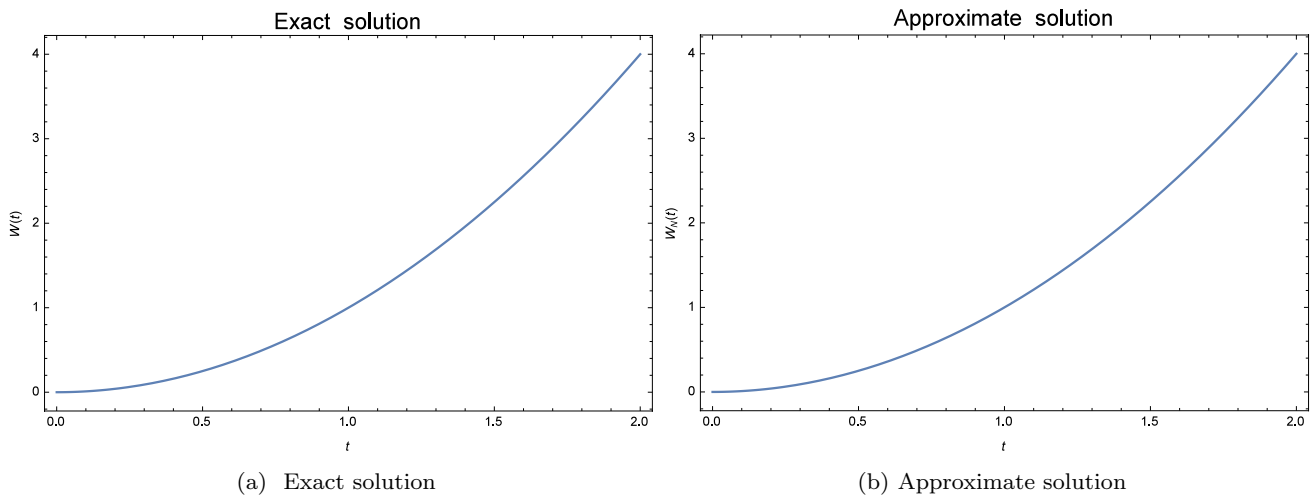
In this section, based on the previous discussion, some numerical examples are given to illustrate the accuracy, efficiency, applicability, generality and validity of the proposed technique. In all examples, the results of the present method are computed by Mathematica 10 software. In order to test our scheme, we compared it with other known methods in terms of absolute errors  $|w_{exact}(t) - w_n(t)|$ , relative errors  $|\frac{w_{exact}(t) - w_n(t)}{w_{exact}(t)}|$  and the CPU time required for solving all examples.

Comparison of the results obtained by this scheme with the exact solution of each example shows that this new technique has a better agreement than other methods. The stability, consistency and easy implementation of this technique cause this method to be more applicable and reliable.

**Example 6.1** [6, 10] Consider the following delay fractional order equation for  $0 < \eta \leq 1, \tau > 0$

**Table 1** Comparison the absolute errors between results in [6, 10] and our results with  $\alpha = 0, \beta = 0$  and  $T = 2$  for Ex.1

$t \in [0, T]$	New technique, $N = 2$	New technique, $N = 7$	Method in [6], $N = 2000$	Method in [10], $N = 2000$
0.2	$5.19525 \times 10^{-7}$	$4.23272 \times 10^{-16}$	$7.81197 \times 10^{-2}$	$7.81550 \times 10^{-2}$
0.4	$1.00211 \times 10^{-6}$	$1.38778 \times 10^{-17}$	$1.29928 \times 10^{-1}$	$1.29978 \times 10^{-1}$
0.6	$1.44777 \times 10^{-6}$	$1.66533 \times 10^{-16}$	$1.90687 \times 10^{-1}$	$1.90760 \times 10^{-1}$
0.8	$1.85649 \times 10^{-6}$	0	$2.48601 \times 10^{-1}$	$2.48694 \times 10^{-1}$
1.0	$2.22827 \times 10^{-6}$	0	$3.07649 \times 10^{-1}$	$3.07763 \times 10^{-1}$
1.2	$2.56312 \times 10^{-6}$	0	$3.66427 \times 10^{-1}$	$3.66563 \times 10^{-1}$
1.4	$2.86104 \times 10^{-6}$	0	$4.25322 \times 10^{-1}$	$4.25479 \times 10^{-1}$
1.6	$3.12202 \times 10^{-6}$	0	$4.84208 \times 10^{-1}$	$4.84387 \times 10^{-1}$
1.8	$3.34607 \times 10^{-6}$	0	$5.43110 \times 10^{-1}$	$5.43312 \times 10^{-1}$
2.0	$3.53318 \times 10^{-6}$	0	$6.02019 \times 10^{-1}$	$6.02243 \times 10^{-1}$
CPU time	0.01560 s	0.63960 s	104.343750 s	215.031250 s



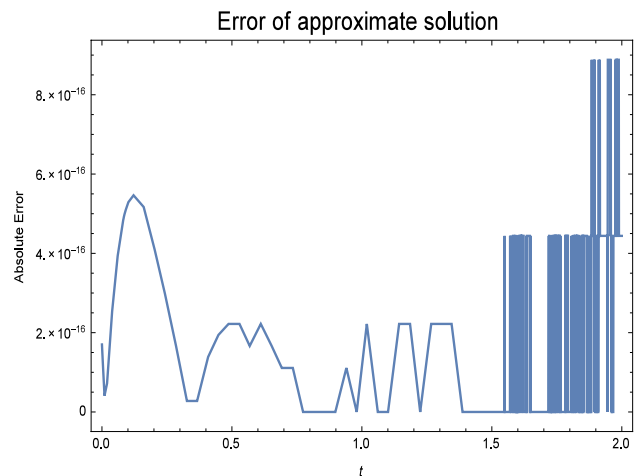
**Fig. 1** Comparison of between approximate solution ( $w_5$ ) of NSJOM method and exact solution for example 1

$$D^\eta w(t) = \frac{2w(t)^{1-\frac{\eta}{2}}}{\Gamma(3-\eta)} + w(t-\tau) - w(t) + 2\tau\sqrt{w(t)} - \tau^2, \tag{42}$$

$w(t) = 0, t \leq 0.$

Note that  $w(t) = t^2$  is the exact solution and  $0 \leq t \leq T, T = 2, \tau = 0.3, \eta = 0.6.$

By the concepts presented in Sect. 4, we consider the approximate solution with  $(N + 1)$  finite terms that presented in Eq. (17) for this problem and substitute in main problem. Then by using Eq. (28), this problem is converted to form of the Eq. (29), finally using  $t_i$ , therefore, a system of algebraic equations emerges which can be solved numerically to determine the unknown vector  $A$  with initial value  $w_0 = 0$ . The solution of the Eq. (42) approximated using new method compared to other methods, is in the



**Fig. 2** The absolute errors comparison between numerical solution ( $w_5$ ) and exact solution for Example 1



best agreement with the exact solution. The absolute errors ( at some nodal points) of this method and methods in [6, 10] are presented and compared in Table 1, also the CPU time needed for these methods is given in this Table. From this Table, it is observed that the numerical results are very close to the exact solution and we achieved an excellent approximation for the exact solution by employing current scheme and it was found the our method in comparison with mentioned methods is better with view to utilization, accuracy and more time efficiency. Figure 1 compares the exact and approximated solution which confirms the reliability of NSJOM method. Moreover, in Fig. 2 we plot the absolute error for this example. Note that the Figs. 1 and 2 show a proper agreement between approximate and exact solutions. In this example for  $N = 2$  and  $N = 7$ , we have

$$A = [1.33333, 2, 0.66667]^T, A = [1.33333, 2, 0.66667, 496 \times 10^{-16}, 8.71848 \times 10^{-17}, 0, 3.55601 \times 10^{-16}, .82553 \times 10^{-16}]^T$$

respectively.

**Example 6.2** [4, 27] Consider the following delay fractional order equation for  $0 < \eta \leq 1, \tau > 0$

$$D^{0.9}w(t) = \frac{2t^{1.1}}{\Gamma(2.1)} - \frac{t^{0.1}}{\Gamma(1.1)} + w(t - 0.1) - w(t) + 0.2t - 0.11, \tag{43}$$

$$w(t) = 0, t \leq 0.$$

In above problem  $w(t) = t^2 - t$  is the exact solution and  $0 \leq t \leq T, T = 10, \tau = 0.1, \eta = 0.9$ .

Using the process mentioned in Ex. 1, we get the solution of this problem. The solution of the Eq. (43) approximated using current method compared to other methods, is in the best agreement with the exact solution. The absolute and relative errors ( at some nodal points) of this method and methods in [4, 27] are presented and compared in Tables 2, 3, also the CPU time needed for these methods is given in these Tables. From these Tables, it is observed that the numerical results are very close to the exact solution and we achieved an excellent approximation for the exact solution by employing new technique and it was found the current method in comparison with mentioned methods is better with view to

**Table 2** Comparison the absolute errors between results in [4, 27] and our results with  $\alpha = 0, \beta = 0$  and  $T = 10$  for Ex.2

$t \in [0, T]$	New technique, $N = 2$	New technique, $N = 3$	Method in [27], $N = 1000$	Method in [4], $N = 1000$
1	$3.33067 \times 10^{-15}$	$3.24185 \times 10^{-14}$	$4.393 \times 10^{-3}$	$4.118 \times 10^{-3}$
2	$9.76996 \times 10^{-15}$	$5.50671 \times 10^{-14}$	$4.117 \times 10^{-3}$	$3.477 \times 10^{-3}$
3	$1.59872 \times 10^{-14}$	$7.10543 \times 10^{-14}$	$3.964 \times 10^{-3}$	$2.986 \times 10^{-3}$
4	$2.13163 \times 10^{-14}$	$8.17124 \times 10^{-14}$	$3.859 \times 10^{-3}$	$2.562 \times 10^{-3}$
5	$2.48390 \times 10^{-14}$	$8.52651 \times 10^{-14}$	$3.780 \times 10^{-3}$	$2.178 \times 10^{-3}$
6	$3.19744 \times 10^{-14}$	$8.17124 \times 10^{-14}$	$3.716 \times 10^{-3}$	$1.824 \times 10^{-3}$
7	$2.84217 \times 10^{-14}$	$7.815973 \times 10^{-14}$	$3.663 \times 10^{-3}$	$1.492 \times 10^{-3}$
8	$4.26326 \times 10^{-14}$	$5.68434 \times 10^{-14}$	$3.618 \times 10^{-3}$	$1.179 \times 10^{-3}$
9	$4.26326 \times 10^{-14}$	$2.84217 \times 10^{-14}$	$3.578 \times 10^{-3}$	$8.820 \times 10^{-4}$
10	$4.26326 \times 10^{-14}$	$1.42109 \times 10^{-14}$	$3.543 \times 10^{-3}$	$6.000 \times 10^{-3}$
CPU time	0 s	0.01560 s	2361.58699 s	1504.27468 s

**Table 3** Comparison the relative errors between results in [4, 27] and our results with  $\alpha = 0, \beta = 0$  and  $T = 10$  for Ex.2

$t \in [0, T]$	New technique, $N = 2$	New technique, $N = 3$	Method in [27], $N = 1000$	Method in [4], $N = 1000$
2	$4.8849 \times 10^{-15}$	$2.7533 \times 10^{-14}$	$2.0585 \times 10^{-3}$	$1.7385 \times 10^{-3}$
3	$2.6645 \times 10^{-15}$	$1.1842 \times 10^{-14}$	$6.6060 \times 10^{-4}$	$4.9770 \times 10^{-4}$
4	$1.7763 \times 10^{-15}$	$6.8093 \times 10^{-15}$	$3.2160 \times 10^{-4}$	$2.1350 \times 10^{-4}$
5	$1.2434 \times 10^{-15}$	$4.2632 \times 10^{-15}$	$1.8900 \times 10^{-4}$	$1.0890 \times 10^{-4}$
6	$1.06580 \times 10^{-15}$	$2.7237 \times 10^{-15}$	$1.2390 \times 10^{-4}$	$6.0800 \times 10^{-5}$
7	$6.7670 \times 10^{-16}$	$1.8609 \times 10^{-15}$	$8.7000 \times 10^{-5}$	$3.5000 \times 10^{-5}$
8	$7.6129 \times 10^{-16}$	$1.0150 \times 10^{-15}$	$6.4600 \times 10^{-5}$	$2.1000 \times 10^{-5}$
9	$5.92116 \times 10^{-16}$	$3.9474 \times 10^{-16}$	$4.9700 \times 10^{-5}$	$1.2250 \times 10^{-5}$
10	$4.7369 \times 10^{-16}$	$1.5789 \times 10^{-16}$	$3.9300 \times 10^{-5}$	$6.6700 \times 10^{-5}$
CPU time	0 s	0.01560 s	2361.58699 s	1504.27468 s

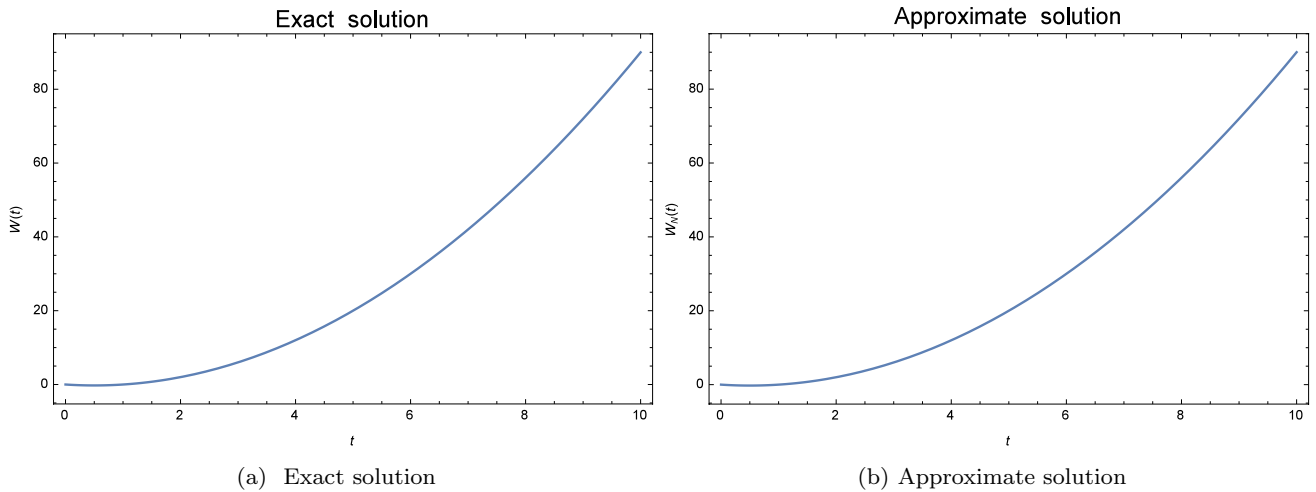


Fig. 3 Comparison of between approximate solution( $w_2$ ) of *NSJOM* method and exact solution for example 2

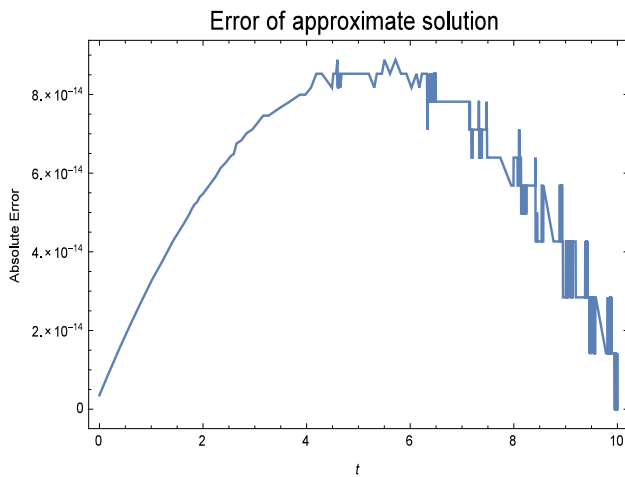


Fig. 4 The absolute errors comparison between numerical solution( $w_2$ ) and exact solution for Example. 2

utilization, accuracy and more time efficiency. Figure 3 compares the exact and approximated solution which confirms the reliability of *NSJOM* method. Moreover, in Fig. 4 we plot the absolute error for this example. Note that the Figs. 3 and 4 show a proper agreement between approximate and exact solution. In this example for  $N = 2$  and  $N = 3$ , we have

$A = [28.33333, 45, 16.66667]^T$ ,  $A = [28.33333, 45, 16.66667, 0]^T$ , respectively.

**Example 6.3** [5] Consider the following delay fractional order equation for  $0 < \eta \leq 1, \tau > 0$

$$D^\eta w(t) = \frac{2t^{2-\eta}}{\Gamma(3-\eta)} - \frac{t^{1-\eta}}{\Gamma(2-\eta)} + w(t-\tau) - w(t) + 2\tau t - \tau^2 - \tau, \tag{44}$$

$w(t) = 0, t \leq 0.$

In this problem  $w(t) = t^2 - t$  is the exact solution and  $0 \leq t \leq T$ .

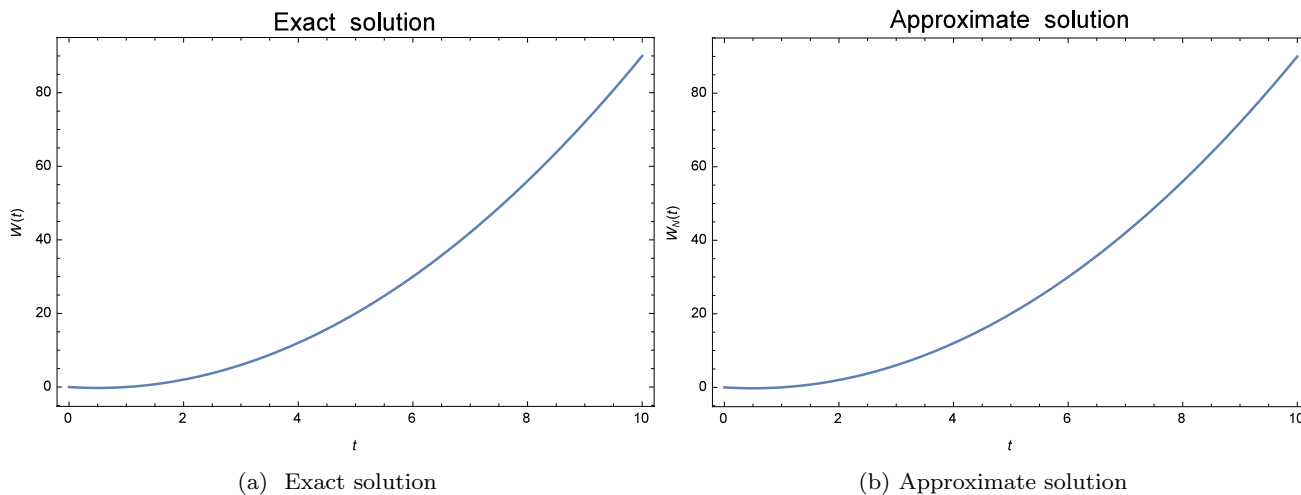
Like pervious examples, we get the solution of this problem in accordance with delay being constant or time varying. The solution of the Eq. (44) approximated using current method compared to other methods, is in the best agreement with the exact solution. The absolute ( $E_A$ ) and relative errors ( $E_R$ ) (at  $t = T$ ) of this method and method in [5] are displayed and compared in Tables 4, 5. From these Tables, it is observed that the numerical results are very close to the exact solution and we gained an excellent approximation for the exact solution by employing new technique and it was found the current technique in comparison with mentioned

**Table 4** Comparison the error treatment between results in [5] and our results with  $\alpha = 0, \beta = 0, \eta = 0.9, \tau = 0.01 \exp(-t)$  at  $t = T$  for Ex.3

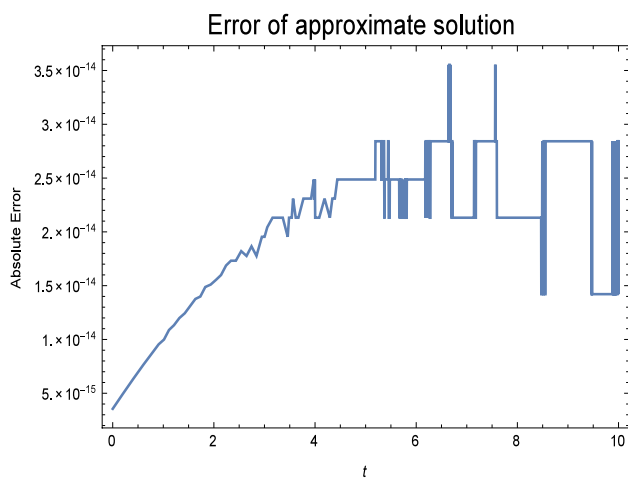
$T$	$E$	New technique, $N = 2$	New technique, $N = 4$	Method in [5] $N = 1600$
5	$E_A$	$1.10371 \times 10^{-15}$	$6.00893 \times 10^{-16}$	$1.0000 \times 10^{-3}$
	$E_R$	$9.20416 \times 10^{-17}$	$4.29713 \times 10^{-17}$	$5.2136 \times 10^{-5}$
10	$E_A$	$1.503711 \times 10^{-14}$	$2.55424 \times 10^{-15}$	$4.7115 \times 10^{-4}$
	$E_R$	$5.99321 \times 10^{-16}$	$1.33251 \times 10^{-16}$	$5.2350 \times 10^{-6}$
50	$E_A$	$2.37010 \times 10^{-14}$	$6.00964 \times 10^{-15}$	$7.0303 \times 10^{-4}$
	$E_R$	$7.00541 \times 10^{-16}$	$8.31663 \times 10^{-17}$	$2.7081 \times 10^{-8}$

**Table 5** Comparison the error treatment between results in [5] and our results with  $\alpha = 0, \beta = 0, \tau = 0.1, \eta = 0.9$  at  $t = T$  for Ex.3

$T$	$E$	New technique, $N = 2$	New technique, $N = 4$	Method in [5], $N = 1600$
5	$E_A$	$7.10543 \times 10^{-15}$	$3.55271 \times 10^{-15}$	$3.3000 \times 10^{-3}$
	$E_R$	$3.552715 \times 10^{-16}$	$1.77636 \times 10^{-16}$	$1.6663 \times 10^{-4}$
10	$E_A$	$4.26326 \times 10^{-14}$	$5.68434 \times 10^{-14}$	$3.1000 \times 10^{-3}$
	$E_R$	$4.73695 \times 10^{-16}$	$6.31594 \times 10^{-16}$	$3.4191 \times 10^{-5}$
50	$E_A$	$3.63798 \times 10^{-12}$	$4.54747 \times 10^{-13}$	$2.3000 \times 10^{-3}$
	$E_R$	$1.48489 \times 10^{-15}$	$1.85611 \times 10^{-16}$	$9.2911 \times 10^{-7}$



**Fig. 5** Comparison of between approximate solution ( $w_2$ ) of *NSJOM* method and exact solution for example. 3. ( $\tau = 0.1, \eta = 0.9$ )



**Fig. 6** The absolute errors comparison between numerical solution ( $w_2$ ) and exact solution for Example. 3. ( $\tau = 0.1, \eta = 0.9$ )

method is better with view to utilization, accuracy and more time efficiency. Figures 5, 7 compare the exact and approximated solution which confirms the reliability of *NSJOM* method. Moreover, in Figs. 6, 8 we plot the absolute errors of this example in accordance with delay being constant or

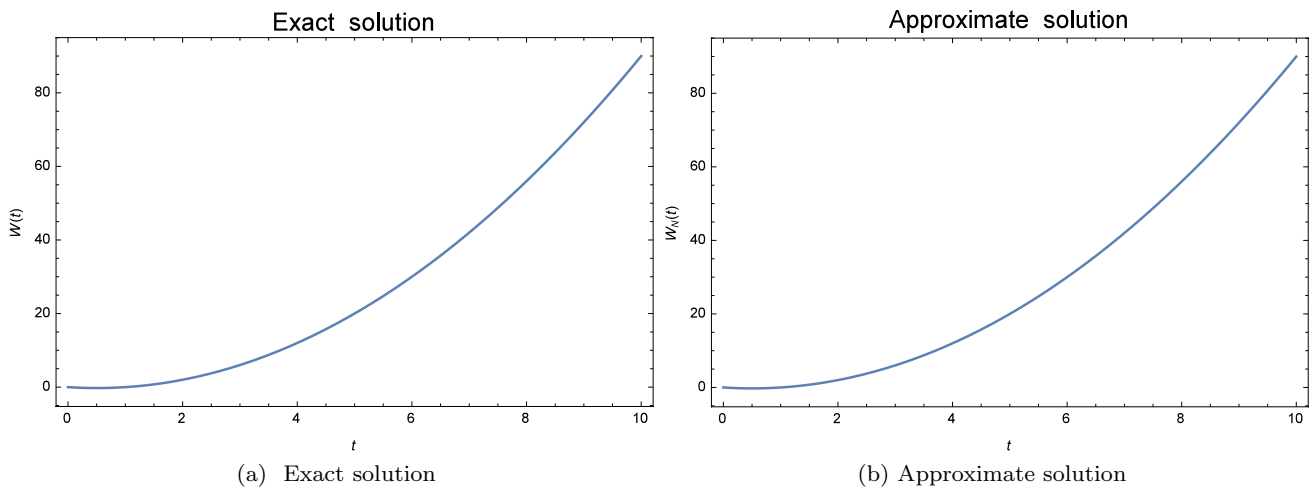
time varying. Note that the Figs. 5, 6, 7 and 8 show a proper agreement between approximate and exact solution.

**Example 6.4** [6, 10] As the last and general example, consider the following delay fractional order equation for  $0 < \eta(t) \leq 1, \tau > 0$

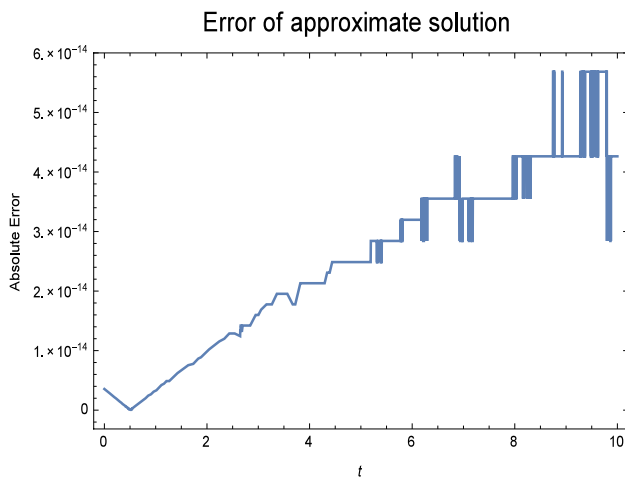
$$\begin{aligned}
 D^{\eta(t)} w(t) &= \frac{2w(t)^{1-\frac{\eta(t)}{2}}}{\Gamma(3-\eta(t))} \\
 &+ w(t-\tau) - w(t) + 2\tau\sqrt{w(t)} - \tau^2, \\
 w(t) &= 0, \quad t \leq 0.
 \end{aligned}
 \tag{45}$$

Note that  $w(t) = t^2$  is the exact solution and  $0 \leq t \leq T, T = 1, \tau = 0.3, \eta(t) = 0.6t$ .

By the concepts presented in Sect. 4, we consider the approximate solution with  $(N + 1)$  finite terms that presented in Eq. (17) for this problem and substitute in main problem. Then by using Eq. (28), this problem is converted to form of the Eq. (29), finally using  $t_i$ , therefore, a system of algebraic equations emerges which can be solved numerically to determine the unknown vector  $A$ . The solutions of the



**Fig. 7** Comparison of between approximate solution ( $w_2$ ) of NSJOM method and exact solution for example. 3. ( $\tau = 0.01 \exp(-t), \eta = 0.9$ )



**Fig. 8** The absolute errors comparison between numerical solution( $w_2$ ) and exact solution for Example. 3. ( $\tau = 0.01 \exp(-t), \eta = 0.9$ )

**Table 6** Absolute errors of  $w(t)$  with  $\alpha = 0, \beta = 0$  and  $T = 1$  for Ex.4

$t \in [0, T]$	New technique $N = 5$	New technique $N = 2$
0.1	$6.80012 \times 10^{-16}$	$8.01146 \times 10^{-9}$
0.2	$6.93889 \times 10^{-16}$	$1.86878 \times 10^{-8}$
0.3	$7.21645 \times 10^{-16}$	$3.2.2292 \times 10^{-8}$
0.4	$8.88178 \times 10^{-16}$	$4.80354 \times 10^{-8}$
0.5	$8.88178 \times 10^{-16}$	$6.67066 \times 10^{-8}$
0.6	$1.11022 \times 10^{-15}$	$8.80427 \times 10^{-8}$
0.7	$1.11022 \times 10^{-15}$	$1.12044 \times 10^{-7}$
0.8	$8.88178 \times 10^{-16}$	$1.38710 \times 10^{-7}$
0.9	$4.44089 \times 10^{-16}$	$1.6804 \times 10^{-7}$
1.0	$4.44089 \times 10^{-16}$	$2.00036 \times 10^{-7}$

**Table 7** Relative errors of  $w(t)$  with  $\alpha = 0, \beta = 0$  and  $T = 1$  for Ex.4

$t \in [0, T]$	New technique $N = 5$	New technique $N = 2$
0.1	$1.70030 \times 10^{-14}$	$2.00286 \times 10^{-7}$
0.2	$4.33681 \times 10^{-15}$	$1.16799 \times 10^{-7}$
0.3	$2.00457 \times 10^{-15}$	$8.89699 \times 10^{-8}$
0.4	$1.38778 \times 10^{-15}$	$7.50553 \times 10^{-8}$
0.5	$8.88178 \times 10^{-16}$	$6.67066 \times 10^{-8}$
0.6	$7.70988 \times 10^{-16}$	$6.11407 \times 10^{-8}$
0.7	$5.66440 \times 10^{-16}$	$5.41834 \times 10^{-8}$
0.8	$3.46945 \times 10^{-16}$	$5.18643 \times 10^{-8}$
0.9	$1.37065 \times 10^{-16}$	$5.18643 \times 10^{-8}$
1.0	$1.11022 \times 10^{-16}$	$5.00091 \times 10^{-8}$

**Table 8** Absolute errors of  $w(t)$  with  $\alpha = 1, \beta = 1$  and  $T = 1$  for Ex.4

$t \in [0, T]$	New technique $N = 5$	New technique $N = 2$
0.1	$3.88578 \times 10^{-16}$	$8.01146 \times 10^{-9}$
0.2	$2.22045 \times 10^{-16}$	$1.86878 \times 10^{-8}$
0.3	$5.55112 \times 10^{-16}$	$3.2.2292 \times 10^{-8}$
0.4	$1.11022 \times 10^{-16}$	$4.80354 \times 10^{-8}$
0.5	0	$6.67066 \times 10^{-8}$
0.6	$4.44089 \times 10^{-15}$	$8.80427 \times 10^{-8}$
0.7	$2.22045 \times 10^{-15}$	$1.12044 \times 10^{-7}$
0.8	0	$1.38710 \times 10^{-7}$
0.9	0	$1.68040 \times 10^{-7}$
1.0	$4.44089 \times 10^{-16}$	$2.00036 \times 10^{-7}$

**Table 9** Relative errors of  $w(t)$  with  $\alpha = 1, \beta = 1$  and  $T = 1$  for Ex.4

$t \in [0, T]$	New technique $N = 5$	New technique $N = 2$
0.1	$9.71445 \times 10^{-15}$	$2.00286 \times 10^{-7}$
0.2	$1.38778 \times 10^{-15}$	$1.16799 \times 10^{-7}$
0.3	$1.54198 \times 10^{-16}$	$8.89699 \times 10^{-8}$
0.4	$1.73472 \times 10^{-16}$	$7.50553 \times 10^{-8}$
0.5	0	$6.67066 \times 10^{-8}$
0.6	$3.08395 \times 10^{-16}$	$6.11407 \times 10^{-8}$
0.7	$1.13288 \times 10^{-16}$	$5.71651 \times 10^{-8}$
0.8	0	$5.41834 \times 10^{-8}$
0.9	0	$5.18643 \times 10^{-8}$
1.0	$1.11022 \times 10^{-16}$	$5.00091 \times 10^{-8}$

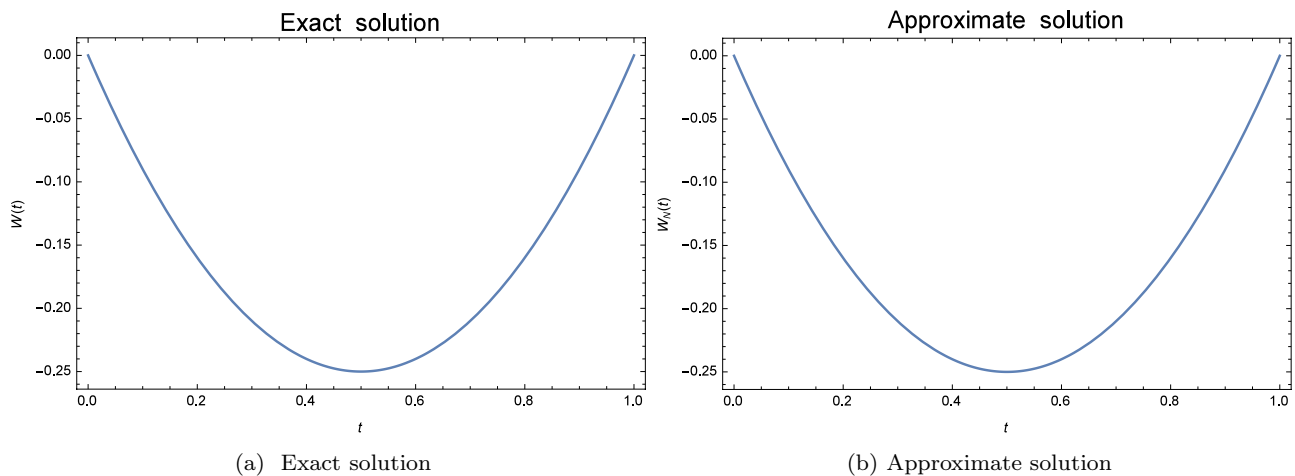
**Table 11** Relative errors of  $w(t)$  with  $\alpha = \frac{1}{2}, \beta = \frac{1}{2}$  and  $T = 1$  for Ex.4

$t \in [0, T]$	New technique, $N = 5$	New technique, $N = 2$
0.1	$9.54098 \times 10^{-15}$	$2.00286 \times 10^{-7}$
0.2	$1.56125 \times 10^{-15}$	$1.16799 \times 10^{-7}$
0.3	$3.08395 \times 10^{-15}$	$8.89699 \times 10^{-8}$
0.4	$2.77556 \times 10^{-15}$	$7.50553 \times 10^{-8}$
0.5	$1.99740 \times 10^{-15}$	$6.67066 \times 10^{-8}$
0.6	$7.70988 \times 10^{-16}$	$6.11407 \times 10^{-8}$
0.7	$7.93016 \times 10^{-16}$	$5.71651 \times 10^{-8}$
0.8	$2.60209 \times 10^{-15}$	$5.41834 \times 10^{-8}$
0.9	$4.38607 \times 10^{-15}$	$5.18643 \times 10^{-8}$
1.0	$6.55032 \times 10^{-15}$	$5.00091 \times 10^{-8}$

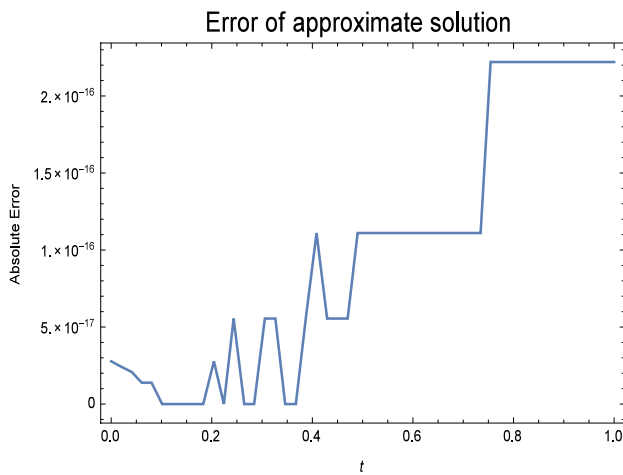
**Table 10** Absolute errors of  $w(t)$  with  $\alpha = \frac{1}{2}, \beta = \frac{1}{2}$  and  $T = 1$  for Ex.4

$t \in [0, T]$	New technique, $N = 5$	New technique, $N = 2$
0.1	$3.81639 \times 10^{-16}$	$8.01146 \times 10^{-9}$
0.2	$2.49800 \times 10^{-16}$	$1.86878 \times 10^{-8}$
0.3	$1.11022 \times 10^{-15}$	$3.20292 \times 10^{-8}$
0.4	$1.77336 \times 10^{-15}$	$4.80354 \times 10^{-8}$
0.5	$1.99840 \times 10^{-15}$	$6.67066 \times 10^{-8}$
0.6	$1.11022 \times 10^{-15}$	$8.80427 \times 10^{-8}$
0.7	$1/55431 \times 10^{-15}$	$1.12044 \times 10^{-7}$
0.8	$6.66134 \times 10^{-15}$	$1.38710 \times 10^{-7}$
0.9	$1.42109 \times 10^{-14}$	$1.68040 \times 10^{-7}$
1.0	$2.62013 \times 10^{-14}$	$2.00036 \times 10^{-7}$

Eq. (45) approximated for various values of  $\alpha, \beta$ , via new method compared to other methods, is in the best agreement with the exact solution. The absolute and relative errors ( at some nodal points) of this method are shown in Tables 6, 7, 8, 9, 10, 11. From these Tables, it is observed that the numerical results are very close to the exact solution and we achieved an excellent approximation for the exact solution by employing current scheme and it was found the our method in comparison with mentioned methods is better with view to utilization, accuracy and more time efficiency. Figure 9 compares the exact and approximated solution which confirms the reliability of *NSJOM* method. Moreover, in Fig. 10 we plot the absolute error with  $\alpha = 1, \beta = 1$  for this example . Note that the Figs. 9 and 10 show a proper



**Fig. 9** Comparison of between approximate solution ( $w_h$ ) of *NSJOM* method and exact solution for example. 4. ( $\tau = 0.3, \eta(t) = 0.6t$ )



**Fig. 10** The absolute errors comparison between numerical solution ( $w_s$ ) and exact solution for Example. 4. ( $\tau = 0.3, \eta(t) = 0.6t$ )

agreement between approximate and exact solution. In this example, we have

If  $\alpha = 0, \beta = 0$  and  $N = 2$ , then

$$A = [1.33333, 2, 0.66667]^T, \text{ CPU time} = 0 \text{ s},$$

if  $\alpha = 0, \beta = 0$  and  $N = 5$ , then

$$A = [1.33333, 2, 0.66667, 1.77479 \times 10^{-16}, 0, -1.44702 \times 10^{-16}]^T, \text{ CPU time} = 0.01560 \text{ s},$$

if  $\alpha = \frac{1}{2}, \beta = \frac{1}{2}$  and  $N = 2$ , then

$$A = [1.25, 1.33333, 0.4]^T, \text{ CPU time} = 0.01560 \text{ s},$$

if  $\alpha = \frac{1}{2}, \beta = \frac{1}{2}$  and  $N = 5$ , then

$$A = [1.25, 1.33333, 0.4, -2.66269 \times 10^{-15}, -3.52736 \times 10^{-16}, -1.04701 \times 10^{-16}]^T, \\ \text{CPU time} = 0.21840 \text{ s},$$

if  $\alpha = 1, \beta = 1$  and  $N = 2$ , then

$$A = [1.2, 1, 0.26667]^T, \text{ CPU time} = 0.01560 \text{ s},$$

if  $\alpha = 1, \beta = 1$  and  $N = 5$ , then

$$A = [1.2, 1, 0.26667, 7.94864 \times 10^{-17}, 5.29022 \times 10^{-17}, -7.42853 \times 10^{-17}]^T, \text{ CPU time} = 0.21840 \text{ s}.$$

## 7 Conclusions

In the current paper, we have proposed the novel shifted Jacobi operational matrix (NSJOM) scheme for variable-order fractional delay differential equations via reducing the

main problem to an algebraic system of equations that can be solved numerically. We have shown that the proposed method has good convergence, is easy to implement and its concepts are simple. The obtained results are excellent compared to other methods. Finally, the numerical results have been presented to clarify the effectiveness and accuracy of this technique.

**Acknowledgements** We are grateful to two anonymous reviewers for their helpful comments, which undoubtedly led to the definite improvements in the paper.

## References

- Baleanu D, Magin RL, Bhalekar S, Daftardar-Gejji V (2015) Chaos in the fractional order nonlinear Bloch equation with delay. *Commun Nonlinear Sci Numer Simul* 25(1–3):41–49
- Diethelm K, Ford NJ, Freed AD (2004) Detailed error analysis for a fractional Adams method. *Numer Algorithms* 36(1):31–52
- Kuang Y (1993) *Delay differential equations: with applications in population dynamics*, vol 191. Academic Press, London
- Jhinga A, Daftardar-Gejji V (2019) A new numerical method for solving fractional delay differential equations. *Comput Appl Math* 38:166
- Wang Z (2013) A numerical method for delayed fractional-order differential equations, Hindawi Publishing Corporation *Journal of Applied Mathematics* Volume 2013
- Daftardar-Gejji V, Sukale Y, Bhalekar S (2015) Solving fractional delay differential equations: a new approach. *Int J Theory Appl* 18:2
- SaedsHoarHeris M, Javidi M (2017) On fractional backward differential formulas for fractional delay differential equations with periodic and anti-periodic conditions. *Appl Numer Math* 118:203–220
- Adomian G (1988) A review of decomposition method in applied mathematics. *J Math Anal Appl* 135:501–544
- Bhalekar S, Daftardar-Gejji V (2011) Convergence of the new iterative method. *Int J Differ Equ*
- Bhalekar S, Daftardar-Gejji V (2011) A predictor-corrector scheme for solving non-linear delay differential equations of fractional order. *J Fract Calculus Appl* 1(5):1–8
- Daftardar-Gejji V, Jafari H (2005) Adomian decomposition: a tool for solving a system of fractional differential equations. *J Math Anal Appl* 301(2):506–518
- Daftardar-Gejji V, Jafari H (2006) An iterative method for solving non linear functional equations. *J Math Anal Appl* 316:753–763
- Hemeda A (2012) Homotopy perturbation method of solving systems of nonlinear coupled equations. *Appl Math Sci* 6:4787–4800
- Wazwaz A (1999) A reliable modification in Adomian decomposition method. *Appl Math Comput* 102:77–86
- El-Sayed AA, Baleanu D, Agarwal P (2020) A novel Jacobi operational matrix for numerical solution of multi-term variable-order fractional differential equations. *J Taibah Univ Sci* 14(1):963–974
- Diethelm K, Ford NJ, Freed AD (2002) A predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dyn* 29:3–22
- Ghasemi M, Fardi M, Ghaziani R, Khoshsiar (2015) Numerical solution of nonlinear delay differential equations of fractional order in reproducing kernel Hilbert space. *Appl Math Comput* 268:815–831

18. Ockendon JR, Tayler AB (1971) The dynamics of a current collection system for an electric locomotive. *Proc R Soc Lond Ser A* 322:447–468
19. Buhmann MD, Iserles A (1993) Stability of the discretized pantograph differential equation. *J Math Comput* 60:575–589
20. Shakeri F, Dehghan M (2008) Solution of delay differential equations via a homotopy perturbation method. *Math Comput Model* 48:486–498
21. Shakeri F, Dehghan M (2008) The use of the decomposition procedure of a domain for solving a delay diffusion equation arising in electrostatics. *Phys Scr Phys Scr* 78:065004, 11pp
22. Sedaghat S, Ordokhani Y, Dehghan M (2012) Numerical solution of the delay differential equations of pantograph type via Chebyshev polynomials. *Commun Nonlin Sci Numer Simul* 17:4125–4136
23. Ajello WG, Freedman HI, Wu J (1992) A model of stage structured population growth with density dependent time delay. *SIAM J Appl Math* 52:855–869
24. Morgado ML, Ford NJ, Lima P (2013) Analysis and numerical methods for fractional differential equations with delay. *J Comput Appl Math* 252:159–168
25. Čermák J, Horníčková J, Kisela T (2016) Stability regions for fractional differential systems with a time delay. *Commun Nonlinear Sci Numer Simul* 31(1):108–123
26. Lazarević MP, Spasić AM (2009) Finite-time stability analysis of fractional order time-delay systems: Gronwall's approach. *Math Comput Model* 49(3):475–481
27. Daftardar-Gejji V, Sukale Y, Bhalekar S (2014) A new predictor–corrector method for fractional differential equations. *Appl Math Comput* 244:158–182
28. Diethelm K, Ford NJ (2002) Analysis of fractional differential equations. *J Math Anal Appl* 265(2):229–248
29. Tavares D, Almeida R, Torres DFM (2016) Caputo derivatives of fractional variable order: numerical approximations. *Commun Nonlinear Sci Numer Simul* 35:69–87
30. Liu J, Li X, Wu L (2016) An operational matrix of fractional differentiation of the second kind of Chebyshev polynomial for solving multi-term variable order fractional differential equation. *Math Probl Eng*, 10 pages
31. Nagy AM, Sweilam NH, El-Sayed AA (2018) New operational matrix for solving multi-term variable order fractional differential equations. *J Comp Nonlinear Dyn* 13:011001–011007
32. El-Sayed AA, Agarwal P (2019) Numerical solution of multi-term variable-order fractional differential equations via shifted Legendre polynomials. *Math Meth Appl Sci* 42(11):3978–3991
33. Mallawi F, Alzaidy JF, Hafez RM (2019) Application of a Legendre collocation method to the space-time variable fractional-order advection-dispersion equation. *J Taibah Univ Sci* 13(1):324–330
34. Bhrawy AH, Zaky MA (2014) A method based on the Jacobi tau approximation for solving multi-term time-space fractional partial differential equations. *J. Comput, Phys*
35. Chen YM, Liu LQ, Li BF et al (2014) Numerical solution for the variable-order linear cable equation with Bernstein polynomials. *Appl Math Comput* 238:329–341
36. Abbasbandy S, Taati A (2009) Numerical solution of the system of nonlinear Volterra integrodifferential equations with nonlinear differential part by the operational Tau method and error estimation. *J Comput Appl Math* 231(1):106–113
37. Szegő G (1985) Orthogonal polynomials, *Am. Math. Soc. Colloq. Pub.* 23
38. Doha EH, Bhrawy AH, Ezz-Eldien SS (2012) A new Jacobi operational matrix: an application for solving fractional differential equations. *Appl Math Model* 36:4931–4943
39. Yousefi SA, Behroozifar M (2010) Operational matrices of Bernstein polynomials and their applications. *Int J Syst Sci* 32:709–716
40. Labacca W, Guimaraes O, Piqueira JRC (2014) Dirac's formalism combined with complex Fourier operational matrices to solve initial and boundary value problems. *Commun Nonlinear Sci Numer Simul* 19(8):2614–2623
41. Razzaghi M, Yousefi S (2005) Legendre wavelets method for the nonlinear Volterra-Fredholm integral equations. *Math. Comput. Simul.* 70:1–8
42. Danfu H, Xufeng S (2007) Numerical solution of integro-differential equations by using CAS wavelet operational matrix of integration. *Appl Math Comput* 194:460–466
43. Behiry SH (2014) Solution of nonlinear Fredholm integro-differential equations using a hybrid of block pulse functions and normalized Bernstein polynomials. *J Comput Appl Math* 260:258–265
44. Saadatmandi A, Dehghan M (2010) A new operational matrix for solving fractional-order differential equations. *Comput Math Appl* 59:1326–1336
45. Saadatmandi A (2014) Bernstein operational matrix of fractional derivatives and its applications. *Appl Math Model* 38:1365–1372
46. Atabakzadeh MH, Akrami MH, Erjaee GH (2013) Chebyshev operational matrix method for solving multi-order fractional ordinary differential equations. *Appl Math Model* 37:8903–8911
47. Bhrawy AH, Alofi AS (2013) The operational matrix of fractional integration for shifted Chebyshev polynomials. *Appl Math Lett* 26:25–31
48. Oliveira (1980) Collocation and residual correction. *Numer Math* 6:27–31
49. Shahmorad S (2005) Numerical solution of the general form linear Fredholm-Volterra integrodifferential equations by the Tau method with an error estimation. *Appl Math Comput* 167:1418–1429
50. de Villiers J (2012) *Mathematics of Approximation*. Atlantis Press, New York
51. Yöüzbaşı S (2012) An efficient algorithm for solving multi-pantograph equation systems. *Comput Math Appl* 64(4):589–603
52. Zlatev Z, Faragó I, Havasi Á (2012) Richardson extrapolation combined with the sequential splitting procedure and  $\theta$ -method. *Central Eur J Math* 10(1):159–172
53. Dehghan M, Manafian J, Saadatmandi A (2010) Solving nonlinear fractional partial differential equations using the homotopy analysis method. *Numer Methods Partial Diff Equ* 26(2):448–479
54. Lakestani M, Dehghan M, Irandoust-pakchin S (2012) The construction of operational matrix of fractional derivatives using B-spline functions. *Commun Nonlinear Sci Numer Simul* 17(3):1149–1162
55. Abbaszadeh M, Dehghan M (2019) Numerical and analytical investigations for neutral delay fractional damped diffusion-wave equation based on the stabilized interpolating element free Galerkin (IEFG) method. *Appl Numer Math* 145:488–506
56. Eslahchi MR, Dehghan M (2011) Application of Taylor series in obtaining the orthogonal operational matrix. *Comput Math Appl* 61(9):2596–2604
57. Kayedi-Bardeh A, Eslahchi MR, Dehghan M (2014) A method for obtaining the operational matrix of the fractional Jacobi functions and applications. *J Vib Control* 20(5):736–748

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.