Alexandria Engineering Journal (2018) 57, 2157–2165

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

ORIGINAL ARTICLE

Cubic B-splines collocation method for a class of partial integro-differential equation

M. Gholamian, J. Saberi-Nadjafi *

Department of Applied Mathematics, Ferdowsi University of Mashhad, International Campus, Mashhad, Iran

Received 18 July 2016; revised 8 May 2017; accepted 10 June 2017
Available online 12 July 2017

KEYWORDS

Cubic B-splines;
Partial integro-differential equation;
Backward Euler method

Abstract In this paper, a numerical method is proposed to estimate the solution of initial-boundary value problems for a class of partial integro-differential equations. This is based on the cubic B-splines method for spatial derivatives while the backward Euler method is used to discretize the temporal derivatives. Detailed discrete schemes are investigated. Next, we proved the convergence and the stability of the proposed method. The method is applied to some test examples and the numerical results have been compared with the exact solutions. The obtained results show the computational efficiency of the method. It can be concluded that computational efficiency of the method is effective for the initial-boundary value problems.

1. Introduction

This paper has used cubic B-spline collocation method for the following second order partial integro-differential equation

$$\frac{\partial u(x,t)}{\partial t} = \int_0^t (t-s)^{-\alpha} \frac{\partial^2 u(x,s)}{\partial x^2} ds + f(x,t), \quad x \in [a,b], \ t \geqslant 0, \ 0 < \alpha < 1 \tag{1}$$

subject to the initial condition

$$u(x,0) = v(x) \tag{2}$$

and the boundary conditions

$$u(a,t) = 0, u(b,t) = 0, \quad t \geqslant 0. \tag{3}$$

The problem (1)–(3) can be found in the applications such as heat conduction in materials with memory, compression of visco-elastic media, biological models and chemical kinetics, fluid dynamics and nuclear reactor dynamics [1–20].

Given that the analytical solution can only be available for a few limited cases, we have used numerical solution as a major approach to compute the solution of these problems. For this goal, some numerical methods have been applied for partial integro- differential equations. These methods include finite-element methods [1–13], finite difference methods [14,15], orthogonal spline collocation methods [16,17], spectral collocation methods [18], Galerkin methods [19] and quasi wavelet methods [20].

There is a challenge in developing accurate numerical methods for partial integro-differential equations; this is because of the possible singularities of the kernel inducing sharp transitions in the solution. Thus, the effective way is to apply collocation method using the B-splines functions in handling the sharp transitions caused by the singularities of the kernel. This is because of two useful features of B-splines in numerical

* Corresponding author.
E-mail addresses: gholamianm2010@gmail.com (M. Gholamian), najafi141@gmail.com (J. Saberi-Nadjafi).
Peer review under responsibility of Faculty of Engineering, Alexandria University.

http://dx.doi.org/10.1016/j.aej.2017.06.004
1110-0168

work. One feature is that the continuity conditions are inherent. Compared with other piecewise polynomial interpolation function, the B-spline is the smoothest interpolation function. Second feature is the B-splines have small local support property, i.e., each B-spline function is only non-zero over a few mesh subintervals. Therefore, the resulting matrix is tightly banded to discretize the equation. B-splines offer special advantages, because they have smoothness and capability to handle local phenomena. When it combined with the collocation, these advantages can significantly simplify the solution procedure of differential equations. The numerical efforts have remarkably decreased, for example, the Regularized Long Wave (RLW) equation can be solved by B-splines in [21,22]. Caglar [23,24] used the B-splines to solve the boundary value problems. As long as we know, the success of the B-splines collocation method is depended on the choice of B-spline basis. In special cases, the cubic B-splines have been used to obtain the numerical solution of the Klein-Gordon equation [25], the RLW equation [22] can be solved by quartic B-splines and the quantic B-splines have been used to build up the numerical solution of the Burgers equation in [26], the KdVB equation [27], the RLW equation [28], the Kuramoto-Sivashinsky equation [29] and cubic spline quasi-interpolation and multi-node higher order expansion have been used to solve the Burgers equation in [30]. Most recently, the quantic B-spline collocation method is applied to obtain the numerical solution of fourth order partial integro-differential equations in [31].

The paper is organized as follows. Detailed description of Section 2 is explained about the cubic B-splines collocation method. In Section 3, we formulate our cubic B-spline collocation method for solving the problem, e.g. (1)–(3). The stability analysis is carried out via Von-Neumann stability as given in Section 4. The convergence analysis of the method is described in Section 5. In Section 6, numerical experiments are tested to demonstrate the viability of the proposed method and this paper ends with a conclusion in Section 7.

2. Description of the method

In this section we use the cubic B-spline collocation method to compute the approximate solution of Eqs. (1)–(3). It can be written as a linear combination of cubic B-splines basis functions. Consider a mesh $a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$ as a uniform partition of the solution domain $a \leqslant x \leqslant b$ by the knots x_j with $h = x_{j+1} - x_j = (b-a)/N, j = 0, 1, \ldots, N-1$.

The numerical treatment for solving (1)–(3) using the collocation method with cubic B-spline is to find an approximate solution $U^N(x,t)$ for the exact solution $u(x,t)$ in the form

$$U^N(x,t) = \sum_{j=-1}^{N+1} \alpha_j(t) B_j(x), \tag{4}$$

where values of $\alpha_j(t)$ are unknown time dependent quantities to be determined by the boundary conditions and collocation of the partial integro-differential equation.

The cubic B-splines $B_j(x)$ for $j = -1, 0, \ldots, N+1$ at the knots are given as follows [32,33]:

$$B_j(x) = \frac{1}{h^3} \begin{cases} (x - x_{j-2})^3, & x \in [x_{j-2}, x_{j-1}) \\ (x - x_{j-2})^3 - 4(x - x_{j-1})^3, & x \in [x_{j-1}, x_j) \\ (x_{j+2} - x)^3 - 4(x_{j+1} - x)^3, & x \in [x_j, x_{j+1}) \\ (x_{j+2} - x)^3, & x \in [x_{j+1}, x_{j+2}) \\ 0, & otherwise, \end{cases} \tag{5}$$

where $\{B_{-1}, B_0, B_1, \ldots, B_{N-1}, B_N, B_{N+1}\}$ form a basis over the region $a \leqslant x \leqslant b$. Each cubic B-spline covers four elements, so that each element is covered by four cubic B-splines. The values of $B_j(x)$ and its first and second derivatives are given as in Table 1. Using the approximation solution (4) and the cubic B-spline functions (5), the approximate values of $U^N(X)$ and its two derivatives at the knots are determined in terms of the time parameters α_j as follows

$$\begin{cases} U_j = \alpha_{j-1} + 4\alpha_j + \alpha_{j+1} \\ U'_j = \frac{3}{h}(\alpha_{j+1} - \alpha_{j-1}) \\ U''_j = \frac{6}{h^2}(\alpha_{j-1} - 2\alpha_j + \alpha_{j+1}) \end{cases} \tag{6}$$

From (4), the boundary conditions (3), reads

$$\begin{cases} U^N(x_0,t) = \sum_{j=-1}^{N+1} \alpha_j(t) B_j(x_0) = 0 \\ U^N(x_N,t) = \sum_{j=-1}^{N+1} \alpha_j(t) B_j(x_N) = 0 \end{cases} \tag{7}$$

In view of Table 1 and from (8), we get

$$\begin{cases} \alpha_{-1} + 4\alpha_0 + \alpha_1 = 0 \\ \alpha_{N-1} + 4\alpha_N + \alpha_{N-1} = 0 \end{cases} \tag{8}$$

We will use the results of this description in next section.

3. Numerical scheme

In this section, the numerical scheme is proposed for solving the one dimensional partial integro-differential equation with the weak singularity kernel, i.e.

$$u_t(x,t) = \int_0^t (t-s)^{-\alpha} u_{xx}(x,s) ds + f(x,t), \quad x \in [0,1], \quad t \geqslant 0, \ 0 < \alpha < 1 \tag{9}$$

subject to the initial condition

$$u(x,0) = v(x) \tag{10}$$

and the boundary conditions

$$u(0,t) = 0, \quad u(1,t) = 0, \quad t \geqslant 0. \tag{11}$$

In this method, the time derivative is dealt with the first order backward Euler scheme and the cubic B-splines are employed to approximate the space derivative. First, let the time level is denoted by $t_n = n\Delta t, n = 0, 1, \ldots$, where Δt is the time step. To apply the proposed method on (9) at time point $t = t_{n+1}$, the first expression in the left of(9) is approximated by

$$u_t(x, t_{n+1}) \approx \frac{u^{n+1}(x) - u^n(x)}{\Delta t}, \quad 0 \le x \le 1, \ n \ge 1. \tag{12}$$

Table 1 Coefficients of cubic B-splines and its derivatives at knots x_j.

x_j	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}
$B_j(x)$	0	1	4	1	0
$B'_j(x_j)$	0	$\frac{3}{h}$	0	$-\frac{3}{h}$	0
$B''_j(x_j)$	0	$\frac{6}{h^2}$	$-\frac{12}{h^2}$	$\frac{6}{h^2}$	0

Therefore, for every $x \in [0,1]$, we have

$$\frac{u^{n+1}(x) - u^n(x)}{\Delta t} = \int_0^{t_{n+1}} (t_{n+1} - s)^{-\alpha} u_{xx}(x,s)ds + f(x, t_{n+1}) \tag{13}$$

The first expression in the right side of (13), yields

$$\begin{aligned}\int_0^{t_{n+1}} (t_{n+1} - s)^{-\alpha} u_{xx}(x,s)ds &= \int_0^{t_{n+1}} s^{-\alpha} u_{xx}(x, t_{n+1} - s)ds \\ &= \sum_{j=0}^{n} \int_{t_j}^{t_{j+1}} s^{-\alpha} u_{xx}(x, t_{n+1} - s)ds \\ &= \sum_{j=0}^{n} u_{xx}(x, t_{n-j+1}) \int_{t_j}^{t_{j+1}} s^{-\alpha} ds \\ &= \frac{\Delta t^{1-\alpha}}{1-\alpha} \sum_{j=0}^{n} u_{xx}(x, t_{n-j+1}) \left[(j+1)^{1-\alpha} - j^{1-\alpha}\right].\end{aligned} \tag{14}$$

Let $b_j = (j+1)^{1-\alpha} - j^{1-\alpha}$, by substituting (14) into (13) and rearranging, we have

$$\begin{aligned}&u^{n+1}(x) - \frac{\Delta t^{2-\alpha}}{1-\alpha} b_0 u_{xx}(x, t_{n+1}) \\ &= u^n(x) + \frac{\Delta t^{2-\alpha}}{1-\alpha} \sum_{j=1}^{n} b_j u_{xx}(x, t_{n-j+1}) + \Delta t f(x, t_{n+1})\end{aligned} \tag{15}$$

Now, the space discretization of (15) is carried out using (4) and the collocation method is implemented by identifying the collocation points as nodes. Therefore, for $i = 0, 1, \ldots, N$, the following formula obtain

$$\begin{aligned}&U_i^{n+1} - \frac{\Delta t^{2-\alpha}}{1-\alpha} b_0 \sum_{j=-1}^{N+1} \alpha_j^{n+1} B''_j(x_i) \\ &= U_i^n + \frac{\Delta t^{2-\alpha}}{1-\alpha} \sum_{k=1}^{n} b_k \sum_{j=-1}^{N+1} \alpha_j^{n-k+1} B''_j(x_i) + \Delta t f_i^{n+1}\end{aligned} \tag{16}$$

where $\alpha_j^{n+1} = \alpha_j(t_{n+1}), f_i^{n+1} = f(x_i, t_{n+1})$. U_i^{n+1} is the approximate solution of $u(x_i)$ at (x_i, t_{n+1}) in (13) and $B''_j(x_i)$ is the second order partial derivative with respect to the space variable x of B_j at x_i. Let $T = \frac{\Delta t^{2-\alpha}}{1-\alpha}$ and

$$F_i = U_i^n + T \sum_{k=1}^{n} b_k \sum_{j=-1}^{N+1} \alpha_j^{n-k+1} B''_j(x_i) + \Delta t f_i^{n+1}, \quad i = 0, 1, \ldots, N, \tag{17}$$

The Eq. (16) yields to the following simplified version

$$\sum_{j=-1}^{N+1} \left[B_j(x_i) - T B''_j(x_i)\right] \alpha_j^{n+1} = F_i, \quad i = 0, 1, \ldots, N. \tag{18}$$

The system (18) consists of $N+1$ linear equations with $N+3$ unknowns, $\alpha_{-1}^{n+1}, \alpha_0^{n+1}, \ldots, \alpha_N^{n+1}$. To obtain the unique solution for this system, the parameters α_{-1}^{n+1} and α_{N+1}^{n+1} are eliminated by imposition the boundary conditions (8). Thus, the system (18) is reduced to a tri-diagonal system of $N+1$ linear equations and $N+1$ unknowns. For the sake of simplification, the system (18) is denoted by the following matrix form

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{F}, \tag{19}$$

where the matrices $A, \boldsymbol{\alpha}$ and $\mathbf{F}$ as follow

$$\mathbf{A} = \begin{bmatrix} \gamma - 4\beta & 0 & 0 & 0 & \cdots & 0 \\ \beta & \gamma & \beta & 0 & \cdots & 0 \\ 0 & \beta & \gamma & \beta & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \beta & \gamma & \beta \\ 0 & \cdots & 0 & 0 & 0 & \gamma - 4\beta \end{bmatrix},$$

$$\boldsymbol{\alpha} = [\alpha_0^{n+1}, \alpha_1^{n+1}, \ldots, \alpha_N^{n+1}]^T, \; n = 0, 1, 2, \ldots, \; F = [F_0, F_1, \ldots, F_N]^T$$

where

$$\beta = 1 - \frac{6T}{h^2}, \quad \gamma = 4 + \frac{6T}{h^2}. \tag{20}$$

We choose

$$u(x, 0) = u^0(x), \quad x \in [0, 1]$$

and by partitioning $[0,1]$ into $N+1$ points, namely $x_0, x_1, \ldots, x_N$ and evaluate $u^0(x)$ at the points, we get the initial vector $U^0 = [u_0^0, u_1^0, \ldots, u_{N-1}^0, u_N^0]$.

4. The stability of the method

The stability of the proposed method is proved by Von-Neumann method. We use Table 1 and set $f(x_i, t_{n+1}) = 0$ in (18) for any $x_i, i = 0, 1, \ldots, N$, to get

$$\beta\alpha_{i-1}^{n+1} + \gamma\alpha_i^{n+1} + \beta\alpha_{i+1}^{n+1} = \widetilde{F}_i, \tag{21}$$

where

$$\widetilde{F}_i = (\alpha_{i-1}^n + 4\alpha_i^n + \alpha_i^{n+1}) + \frac{6T}{h^2} \sum_{l=1}^{n} b_l (\alpha_{i-1}^{n-l+1} - 2\alpha_i^{n-l+1} + \alpha_{i+1}^{n-l+1})$$

Next, we assume that the solution of (18) is presented as

$$\alpha_i^n = \xi^n e^{ki\eta h}$$

where ξ represents the time dependence of the solution, the exponential function shows the spatial dependence such that ηh represents the position along the grid and k is $\sqrt{-1}$. By substituting α_i^n into the (21), we have

$$P\xi^{n+1} e^{ki\eta h} = Q\xi^n e^{ki\eta h} + \frac{6T}{h^2} \sum_{l=1}^{n} b_l R \xi^{n-l+1} e^{ki\eta h}, \tag{22}$$

where

$$\begin{cases} P = \beta(e^{k\eta h} + e^{-k\eta h}) + \gamma, \\ Q = e^{k\eta h} + e^{-k\eta h} + 4, \\ R = e^{k\eta h} + e^{-k\eta h} - 2. \end{cases} \tag{23}$$

On the other hand, substituting the values of γ and β from (20) into (23) and using $e^{k\eta h} + e^{-k\eta h} = 2cos\eta h$, (23) becomes as follows:

$$\begin{cases} P = 2(1 - \frac{6T}{h^2})cos\eta h + (4 + \frac{12T}{h^2}), \\ Q = 2cos\eta h + 4, \\ R = 2cos\eta h - 2. \end{cases} \tag{24}$$

Now, dividing both sides of (22) by $P\xi e^{ik\eta h}$ and after rearranging the results equation, we have

$$\xi^n - \left(\frac{Q}{P} + \frac{6T}{h^2}\frac{R}{P}\right)\xi^{n-1} - \frac{6T}{h^2}\frac{R}{P}\sum_{l=2}^{N} b_l \xi^{n-l} = 0. \tag{25}$$

We choose

$$\begin{cases} a_1 = -\frac{Q}{P} - \frac{6T}{h^2}\frac{R}{P} \\ a_l = -\frac{6T}{h^2}\frac{R}{P} b_l, \quad l = 2, \ldots, n. \end{cases} \tag{26}$$

Using (26), the Eq. (25) becomes as follows

$$\xi^n + a_1\xi^{n-1} + a_2\xi^{n-2} + \cdots + a_{n-1}\xi + a_n = 0 \tag{27}$$

It is easy to see that in (24), $P, Q > 0$ and $R \geqslant 0$. Therefore, all of the coefficients $a_1, a_2, \ldots, a_n$ are positive in the Eq. (27).

In the rest of the procedure, we also need to use the following theorem:

Theorem 4.1. *For all values of roots x_i of an arbitrary polynomial as*

$$P(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n,$$

we have

$$|\xi_i| \leqslant \max\left\{1, \sum_{j=1}^{n}\left|\frac{a_j}{a_0}\right|\right\} \tag{28}$$

Proof. For proof see [34].

Now, for stability, it must be proved that all roots ξ_i of the Eq. (27) satisfy $|\xi_i| \geqslant 1$. According to Theorem 4.1, $a_0 = 1$ and $a_j > 0, j = 1, \ldots, n$. So, we have

$$\sum_{j=1}^{n}\left|\frac{a_j}{a_0}\right| = \frac{-(Q + \frac{6T}{h^2}R\sum_{j=1}^{n} b_j)}{P} = \frac{-(Q + \frac{6T}{h^2}R[(n+1)^{1-\alpha} - 1])}{P}, \tag{29}$$

where

$$\sum_{j=1}^{n} b_j = \sum_{j=1}^{n}[(j+1)^{1-\alpha} - j^{1-\alpha}] = (n+1)^{1-\alpha} - 1. \tag{30}$$

Suppose $N_\alpha = (n+1)^{1-\alpha} - 1$. From (29) we obtain

$$\sum_{j=1}^{n} a_j = \frac{-(Q + \frac{6T}{h^2}RN_\alpha)}{P}. \tag{31}$$

In definition of R in (23), if we set $R = 0$, then $h = 0$. As a result, the Eq. (25) can be rewritten as

$$\xi^n - \frac{Q}{P}\xi^{n-1} = 0 \Rightarrow \xi = 0 \ or \ \xi = -\frac{1}{3}.$$

Therefore, stability condition in this case is proved.

If $R \neq 0 (R < 0)$ and from (31), if we have

$$-\left(Q + \frac{6T}{h^2}RN_\alpha\right) < P, \tag{32}$$

then the stability condition, namely $|\xi_i| \leqslant 1$, is fulfilled.

Now, using Eq. (24), from (32) we obtain

$$-4cos\eta h + \frac{12T}{h^2}(N_\alpha - 1)(-cos\eta h + 1) < 0 \tag{33}$$

or

$$cos\eta h > \frac{\frac{3T}{h^2}(N_\alpha - 1)}{1 + \frac{3T}{h^2}(N_\alpha - 1)}. \quad \square \tag{34}$$

Therefore, (34) is the condition for conditional stability of the method.

5. Convergence analysis

For proving the convergence of the proposed method, we estimate the convergency in space and temporal direction separately. First, we describe the convergency in the space direction with proving the following theorem:

Theorem 5.1. *Let $\widehat{u}(x)$ be the exact solution of Eqs. (1)–(3) and $\widehat{s}(x)$ be the B-spline collocation approximation to $\widehat{u}(x)$, then the method has second order convergence and*

$$\|\widehat{u}(x) - \widehat{s}(x)\|_\infty \leqslant \omega h^2, \tag{35}$$

where $\omega = \lambda_0 Lh^2 + k$ is finite constant.

Proof. Suppose $\widehat{u}(x)$ be the exact solution of Eq. (1)–(3) and also $\widehat{s}(x)$ be the cubic B-splines collocation approximation to $\widehat{u}(x)$. Therefore, we have

$$\widehat{s}(x) = \widehat{u}(x) = \sum_{j=-1}^{N+1}\widehat{\alpha}_j(t)B_j(x) \tag{36}$$

where $\widehat{\alpha} = (\widehat{\alpha}_{-1}, \widehat{\alpha}_0, \ldots, \widehat{\alpha}_{N+1})$.

Also, we assume that $\widetilde{s}(x)$ be the computed cubic B-splines approximation to $\widehat{s}(x)$, namely

$$\widetilde{s}(x) = \sum_{j=-1}^{N+1}\widetilde{\alpha}_j(t)B_j(x) \tag{37}$$

where $\widetilde{\alpha} = (\widetilde{\alpha}_{-1}, \widetilde{\alpha}_0, \ldots, \widetilde{\alpha}_{N+1})$.

To approximate the error $\|\widehat{u}(x) - \widehat{s}(x)\|_\infty$, we have to estimate the error $\|\widehat{u}(x) - \widetilde{s}(x)\|_\infty$ and $\|\widetilde{s}(x) - \widehat{s}(x)\|_\infty$ separately.

Due to Eq. (18), to compute $\widetilde{s}(x)$ and $\widehat{s}(x)$, we must obtain the values of vectors $\widehat{\boldsymbol{\alpha}}$ and $\widetilde{\boldsymbol{\alpha}}$ from two systems of linear equations as follows

$$\mathbf{A}\widehat{\boldsymbol{\alpha}} = \widehat{\mathbf{F}}, \tag{38}$$

and

$$\mathbf{A}\widetilde{\boldsymbol{\alpha}} = \widetilde{\mathbf{F}}. \tag{39}$$

Now, by subtracting (38) and (39), we obtain

$$\mathbf{A}(\widetilde{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}) = \widetilde{\mathbf{F}} - \widehat{\mathbf{F}}. \tag{40}$$

On the other hand, due to the definition of the matrix $\mathbf{A}$ and Eq. (19), $\mathbf{A}$ is a strictly diagonally dominant matrix. Thus, it is nonsingular. Hence, we can write

$$\widetilde{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}} = \mathbf{A}^{-1}(\widetilde{\mathbf{F}} - \widehat{\mathbf{F}}) \tag{41}$$

Taking infinity norm from (41), we obtain

$$\|\widetilde{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}\|_\infty \leqslant \|\mathbf{A}^{-1}\|_\infty \|\widetilde{\mathbf{F}} - \widehat{\mathbf{F}}\|_\infty \tag{42}$$

Now, assume that $\eta_i (0 \leqslant i \leqslant N)$ is the summation of the ith row of matrix $\mathbf{A} = [a_{ij}]_{(N+1)\times(N+1)}$. Therefore, we have

$$\eta_0 = \sum_{j=0}^{N} a_{0j} = \gamma - 4\beta, \tag{43}$$

$$\eta_i = \sum_{j=0}^{N} a_{ij} = \gamma + 2\beta, \quad i = 1, \ldots, N-1 \tag{44}$$

$$\eta_N = \sum_{j=0}^{N} a_{Nj} = \gamma - 4\beta. \tag{45}$$

From the theory of the matrices, we have:

$$\sum_{i=0}^{N} a_{kj}^{-1} = 1, \quad k = 0, 1, \ldots, N, \tag{46}$$

where a_{kj}^{-1} are the entries of $\mathbf{A}^{-1}$. Thus,

$$\|\mathbf{A}^{-1}\|_\infty = \sum_{i=0}^{N} |a_{kj}^{-1}| \leqslant \frac{1}{\eta}, \tag{47}$$

where $\eta = \min_{0\leqslant i\leqslant N} \eta_i = \min(\alpha - 4\beta, \gamma + 2\beta) = \min(\frac{18T}{h^2}, 6)$.

Substituting (47) into (42) we can find

$$\|\widetilde{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}\|_\infty \leqslant \frac{1}{\eta} \|\widetilde{\mathbf{F}} - \widehat{\mathbf{F}}\|_\infty \tag{48}$$

For computing the upper bound of $\|\widetilde{\mathbf{F}} - \widehat{\mathbf{F}}\|_\infty$, from Eq. (17) for all values of $0 \leqslant i \leqslant N$, we have

$$\begin{aligned} |\widetilde{F}_i - \widehat{F}_i| \leqslant\ & |\widetilde{u}_i - \widehat{u}_i| + \Delta t |\widetilde{f}_i^{n+1} - \widehat{f}_i^{n+1}| \\ & + \frac{6T}{h^2} \sum_{k=1}^{n} |b_k| (|\widetilde{\alpha}_{i-1}^{n-k+1} - \widehat{\alpha}_{i-1}^{n-k+1}| + 2|\widetilde{\alpha}_i^{n-k+1} \\ & - \widehat{\alpha}_i^{n-k+1}| + |\widetilde{\alpha}_{i+1}^{n-k+1} - \widehat{\alpha}_{i+1}^{n-k+1}|) \end{aligned} \tag{49}$$

Now, we need to recall the following theorem. □

Theorem 5.2. *If $f(x) \in c^4[a,b], |f^{(4)}(x)| \leqslant L, \forall x \in [a,b]$ and $\Delta = \{a = x_0 < x_1 < \cdots < x_N = b\}$ be the equally spaced partition of $[a,b]$ with step size h. If $s(x)$ is the unique spline function interpolate $f(x)$ at knots $x_0, x_1, \ldots, x_N$, then there exit a constant λ_j such that,*

$$\|f^{(j)} - s^{(j)}\|_\infty \leqslant \lambda_j L h^{4-j}, \quad j = 0, 1, 2, 3.$$

Proof. See [35,36]. Now, due to the above theorem, we have

$$|\widetilde{u}_i - \widehat{u}_i| = |\widetilde{s}(x_i) - \widehat{s}(x_i)| \leqslant \lambda_0 L h^4 \tag{50}$$

In addition, $\{b_k\}_{k=1}^{n}$ is a sequence of positive terms descending and $b_k \leq 1$ for $1 \leqslant k \leqslant n$. Thus, from (50) and by supposing $\widetilde{f}_i^{n+1} = \widehat{f}_i^{n+1}$, we can rewrite (49) as follows

$$\|\widetilde{F} - \widehat{F}\|_\infty \leqslant \lambda_0 L h^4 + \frac{6T}{h^2} \sum_{k=1}^{n} m_k, \tag{51}$$

where

$$(|\widetilde{\alpha}_{i-1}^{n-k+1} - \widehat{\alpha}_{i-1}^{n-k+1}| + 2|\widetilde{\alpha}_i^{n-k+1} - \widehat{\alpha}_i^{n-k+1}| + |\widetilde{\alpha}_{i+1}^{n-k+1} - \widehat{\alpha}_{i+1}^{n-k+1}|) \leqslant m_k.$$

By assuming $\sum_{k=1}^{n} m_k = M_n$ and $\lambda_0 L h^4 + \frac{6T}{h^2} M_n = K_n$, we get

$$\|\widetilde{F} - \widehat{F}\|_\infty \leqslant K_n \tag{52}$$

Using (52), from (48) we have

$$\|\widetilde{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}\|_\infty \leqslant K h^2 \tag{53}$$

where $Kh^2 = \frac{1}{\eta} K_n = \max(\frac{1}{6}, \frac{h^2}{18T}) K_n$. □

To proceed the rest we also need to note the following theorem.

Theorem 5.3. *The B-splines $\{B_{-1}, B_0, B_1, \ldots, B_{N-1}, B_N, B_{N+1}\}$ satisfy the following inequality*

$$\left| \sum_{j=-1}^{N+1} B_j(x) \right| \leqslant 1, \quad 0 \leqslant x \leqslant 1 \tag{54}$$

Proof. See [37].

Now, by subtracting (37) from (36), we have

$$\widetilde{s}(x) - \widehat{s}(x) = \sum_{j=-1}^{N+1} (\widetilde{\alpha}_j - \widehat{\alpha}_j) B_j(x). \tag{55}$$

Using the above theorem and after taking the norm from (55), we obtain

$$\begin{aligned} \|\widetilde{s}(x) - \widehat{s}(x)\|_\infty &= \left\| \sum_{j=-1}^{N+1} (\widetilde{\alpha}_j - \widehat{\alpha}_j) B_j(x) \right\|_\infty \\ &\leqslant \left| \sum_{j=-1}^{N+1} B_j(x) \right| \|\widetilde{\alpha}_j - \widehat{\alpha}_j\|_\infty \leqslant k h^2. \end{aligned} \tag{56}$$

On the other hand, from Theorem 5.3and after taking the norm from (50), we have

$$\|\widehat{u}(x) - \widetilde{s}(x)\|_\infty \leqslant \lambda_0 L h^4. \tag{57}$$

Therefore, from (56) and (57) we get

$$\begin{aligned} \|\widehat{u}(x) - \widehat{s}(x)\|_\infty &\leqslant \|\widehat{u}(x) - \widetilde{s}(x)\|_\infty + \|\widetilde{s}(x) - \widehat{s}(x)\|_\infty \\ &\leqslant \lambda_0 L h^4 + k h^2 = \omega h^2, \end{aligned} \tag{58}$$

where $\omega = \lambda_0 L h^2 + k$. □

Now, for estimate the convergence in temporal direction, we applied Taylor expansion. Therefore, we have from (15)

$$\left(u^n(x)+\Delta t u_t^n+\frac{\Delta t^2}{2!}u_{tt}^n+\cdots\right) - \frac{\Delta t^{2-\alpha}}{1-\alpha}b_0\left(u_{xx}(x,t_n)+\Delta t u_{xxt}(x,t_n)+\frac{\Delta t^2}{2!}u_{xxtt}(x,t_n)+\cdots\right) = u^n(x)+\frac{\Delta t^{2-\alpha}}{1-\alpha}\sum_{j=1}^{n}b_j u_{xx}(x,t_{n-j+1}) + \Delta t(f(x,t_n)+\Delta t f_t(x,t_n)) \tag{59}$$

And from rearranging we get

$$\Delta t\left(u_t^n - f(x,t_n)\right) - \frac{\Delta t^{2-\alpha}}{1-\alpha}\left(b_0 u_{xx}(x,t_n)+\sum_{j=1}^{n}b_j u_{xx}(x,t_{n-j+1})\right) + \Delta t u_{xxt}(x,t_n)+\frac{\Delta t^2}{2!}\left(u_{xxtt}(x,t_n)+\cdots\right) + \frac{\Delta t^2}{2!}\left(u_{tt}^n+f_t(x,t_n)\right)+\cdots = O(\Delta t) \tag{60}$$

Finally, if we suppose that $u(x,t)$ is the exact solution of the Eqs. (1)–(3) and $u^N(x,t)$ is the numerical approximation to this solution by applying the numerical method, from (58) and (60), we will have

$$\|u(x,t)-u^N(x,t)\| \leqslant \rho(k+h^2). \tag{61}$$

where ρ is finite constant.

6. Numerical experiments

In this section, some numerical examples are considered to demonstrate the efficiency and accuracy of the proposed method. All calculations are run with Matlab R2014a software on a Pentium PC Laptop with Core i3-350M Processor 2.26 GHz of CPU and 4G RAM. We have solved the various examples based on a variety of temporal and spatial divisions.

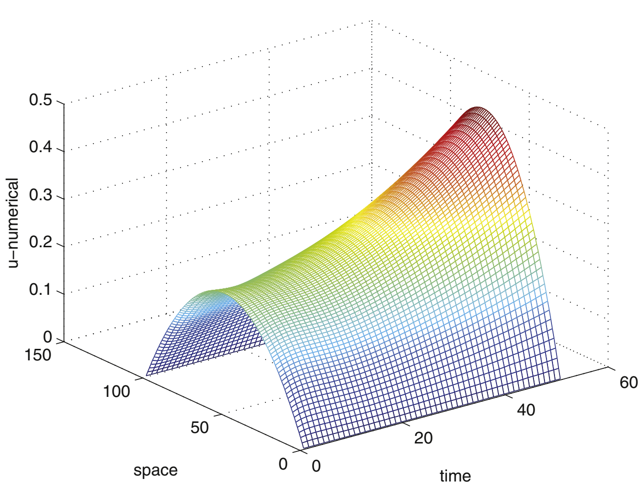

Figure 1 The numerical solutions of example 6.1 for $M=50$ and $N=100$.

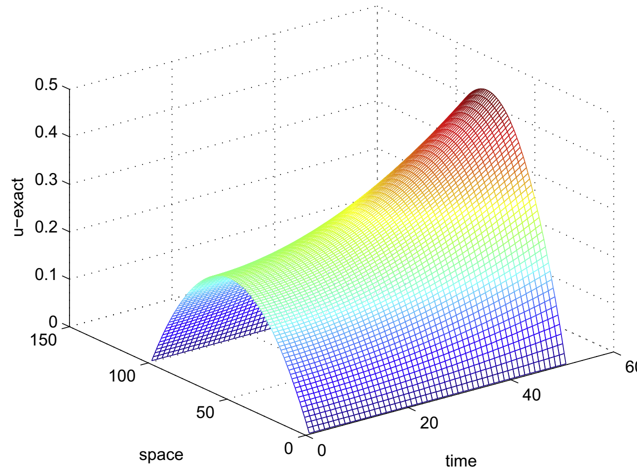

Figure 2 The exact solutions of example 6.1 for $M=50$ and $N=100$.

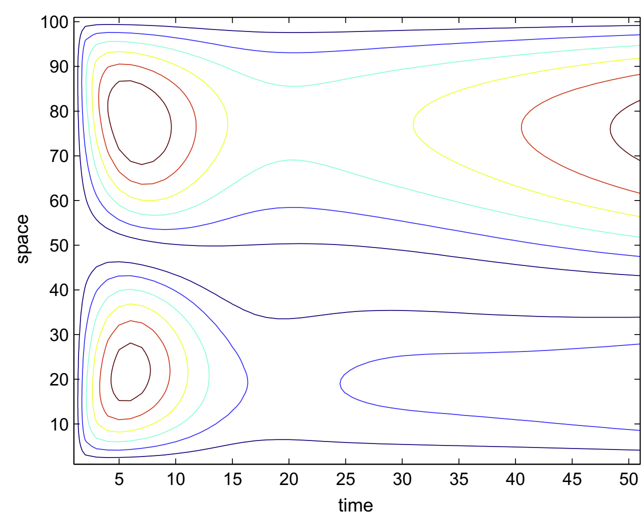

Figure 3 The L_2 error of example 6.1 for $M=50$ and $N=100$.

In all of examples, we have used the variables M and N for temporal and spatial divisions, respectively.

Furthermore, because of conditionally stability, we have applied the stability condition for temporal and spatial divisions and obtained the Error of computations in L_2 and L_∞ error norms as follows

$$L_2 = \|u^{exact}-u^{num}\|_2 = \left(\sum_{i=0}^{N}|u_i^{exact}-u_i^{num}|^2\right)^{1/2}$$

$$L_\infty = \|u^{exact}-u^{num}\|_\infty = \max_{0\leqslant i\leqslant N}|u_i^{exact}-u_i^{num}|$$

Example 6.1. In Eqs. (1)–(3), consider $f(x,t)=2t(x-x^2)+4t^{\frac{1}{2}}(\frac{8}{15}t^2+1)$, for $\alpha=\frac{1}{2}$ the exact solution is $u(x,t)=(x-x^2)(t^2+1), x\in[0,1], 0\leqslant t\leqslant 1$. The initial condition is $u(x,0)=(x-x^2), x\in[0,1]$ and the boundary conditions are $u(0,t)=u(1,t)=0, 0\leqslant t\leqslant 1$. We have solved the problem

Table 2 The L_∞ error norm for some cases of divisions of N and M for example 6.1.

	$N = 100$	$N = 200$	$N = 500$	$N = 2000$
$M = 50$	0.0014	0.008	0.005	0.004
$M = 100$	0.003	0.0015	0.0006	0.0003
$M = 200$	0.006	0.003	0.001	0.0003

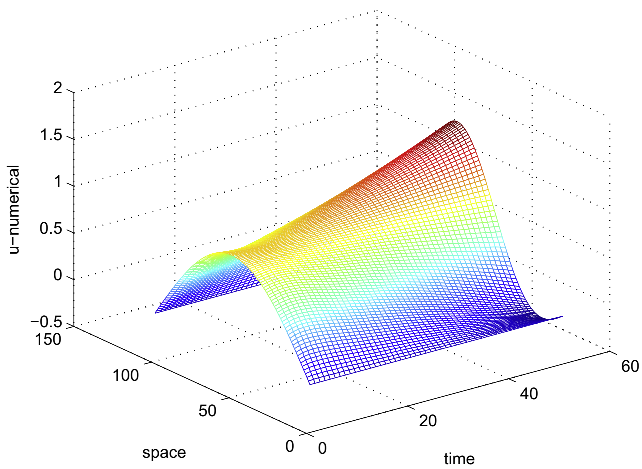

Figure 4 The numerical solutions of example 6.2 for $M = 50$ and $N = 100$.

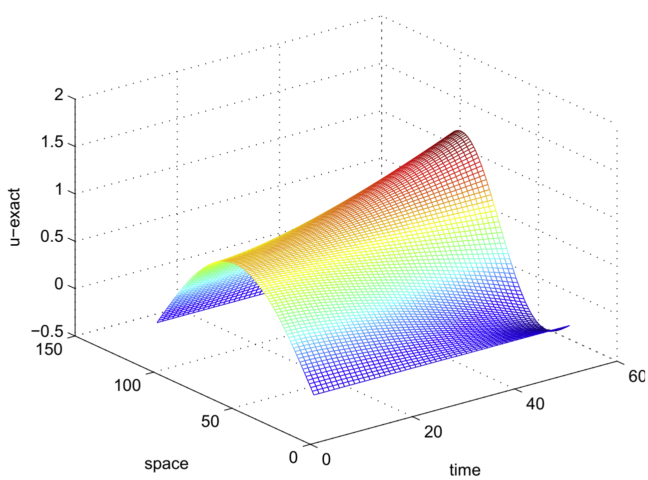

Figure 5 The exact solutions of example 6.2 for $M = 50$ and $N = 100$.

based on a variety of temporal and spatial divisions and applied the stability condition for temporal and spatial divisions. We have obtained the error of computations in L_2 and L_∞. The numerical and exact solutions are shown for a case of divisions as $M = 50$ and $N = 100$ in Figs. 1 and 2.

On Fig. 3. the L_2 error norm is shown for the case of $M = 50$ and $N = 100$.

On Table 2, the L_∞ error norm is shown for some cases of divisions of N and M as follows.

Example 6.2. The following example is taken from [38]. In Eq. (1), we assume that

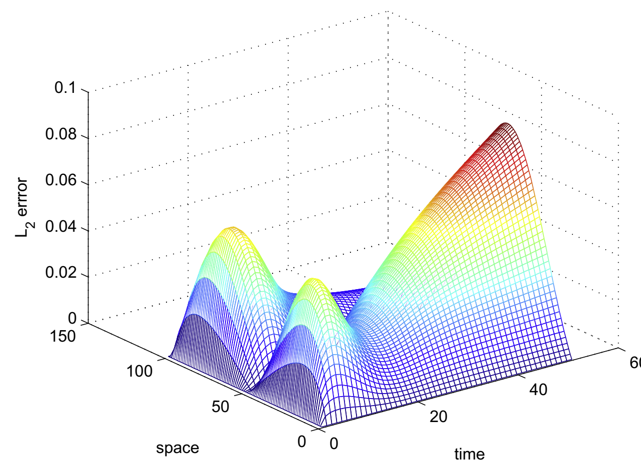

Figure 6 The L_2 error of example 6.2 for $M = 50$ and $N = 100$.

Table 3 The L_∞ error norm for some cases of divisions of N and M for example 6.2.

	$N = 100$	$N = 200$	$N = 500$	$N = 2000$
$M = 50$	0.0089	0.0047	0.0023	0.00013
$M = 100$	0.017	0.0088	0.0036	0.003
$M = 200$	0.035	0.017	0.0071	0.0018

$$f(x,t) = \frac{2t^{\frac{1}{2}}}{\sqrt{\pi}}\left(\pi^{\frac{5}{2}}\sin(\pi x) - \frac{4t^{\frac{3}{2}}}{\sqrt{\pi}}\sin(2\pi x)\right) - 2\pi^{\frac{5}{2}}t^2 sin(2\pi x)$$

The exact solution with $\alpha = \frac{1}{2}$ is $u(x,t) = \sin(\pi x) - \frac{4t^{\frac{3}{2}}}{\sqrt{\pi}}\sin(2\pi x)$. The initial condition is $u(x,0) = \sin(\pi x)$, $x \in [0,1]$ and the boundary conditions are $u(0,t) = u(1,t) = 0$, $0 \leqslant t \leq 1$. The numerical and exact solution are shown for a case of divisions as $M = 50$ and $N = 100$ in Figs. 4, 5 and on Fig. 6 the L_2 error norm is shown for the case of $M = 50$ and $N = 100$. On Table 3, the L_2 error norm is shown for some cases of divisions of M and N as follows.

Example 6.3. The following example is taken from [39], which considered

$$f(x,t) = (1 - x^2)\left(1 - \frac{t^2}{2}\right) + 2t.$$

The exact solution is $u(x,t) = (1 - x^2)t, x \in [-1,1]$. The initial condition is $u(x,0) = 0, x \in [-1,1]$ and the boundary conditions are $u(-1,t) = u(1,t) = 0, 0 \leqslant t \leq 1$. The numerical and exact solution and the L_2 error norm are shown for a case of divisions as $M = 75, N = 100$ in Figs. 7–9.

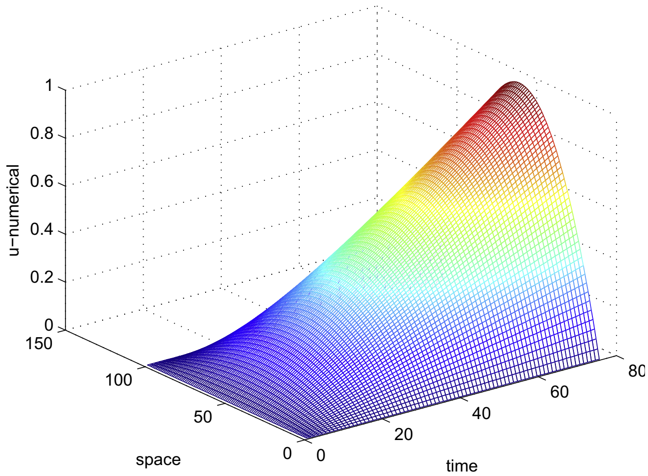

Figure 7 The numerical solutions of example 6.3 for $M = 75$ and $N = 100$.

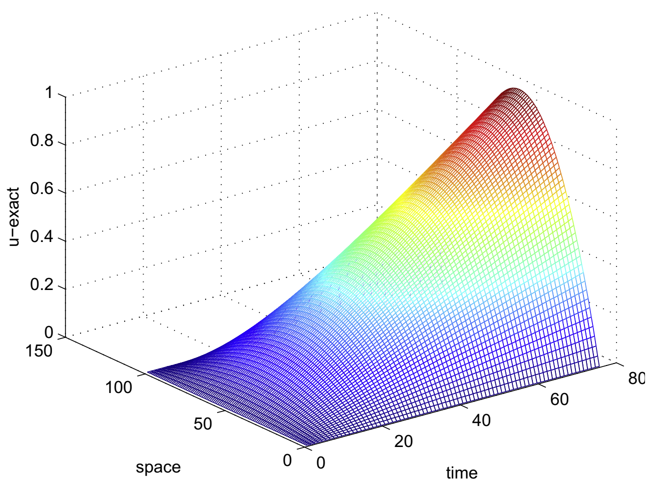

Figure 8 The exact solutions of example 6.3 for $M = 75$ and $N = 100$.

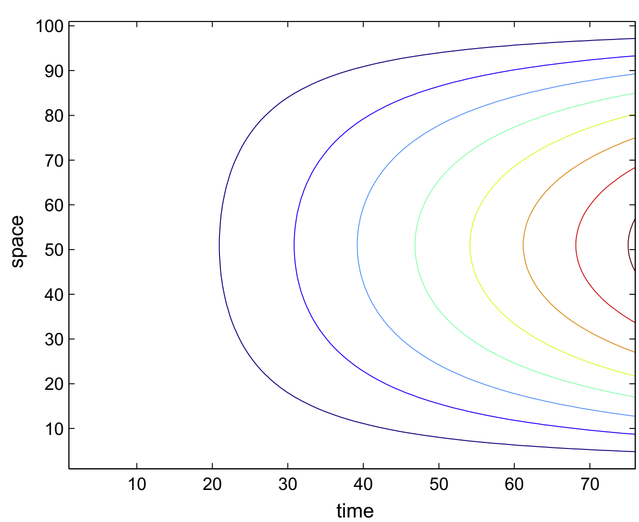

Figure 9 The L_2 error of example 6.3 for $M = 75$ and $N = 100$.

7. Conclusion

The results which we obtained in this research indicated that the discrete schemes are developed in this study and the stability and convergence of the method is confirmed by the analysis. The numerical results have indicated the accuracy of the method. In other words, to validate the proposed method of this study, we have presented some numerical examples. The results have also demonstrated that the proposed computational method is efficient for these problems.

Acknowledgments

The authors would like to express their deep appreciation to the editor and the referees for their valuable comments and suggestions.

References

[1] E. Yanik, G. Fairweather, Finite element methods for parabolic and hyperbolic partial integro-differential equations, Non-Linear Anal. 12 (1988) 809.

[2] W. McLean, V. Thome, Numerical solution of an evolution equation with a positive-type memory term, J. Aust. Math. Soc. Ser. B 35 (1993) 23–70.

[3] C. Chen, V. Thome, L. Wahlbin, Finite element approximation of a parabolic integro-differential equation with a weakly singular kernel, Math. Comput. 58 (1992) 587–602.

[4] I. Sloan, V. Thome, Time discretization of an integro-differential equation of parabolic type, SIAM J. Numer. Anal. (1986) 1052–1061.

[5] S. Larsson, V. Thome, L. Wahlbin, Numerical solution of parabolic integro-differential equations by the discontinuous Galerkin method, Math. Comput. 67 (1998) 45–71.

[6] Y. Huang, Time discretization scheme for an integro-differential equation of parabolic type, J. Comput. Math. 12 (1994) 259–263.

[7] W. McLean, V. Thome, L. Wahlbin, Discretization with variable time steps of an evolution equation with a positive-type memory term, J. Comput. Appl. Math. 69 (1996) 49–69.

[8] C. Lubich, Discretized fractional calculus, SIAM J. Math. Anal. 17 (1986) 704.

[9] D. Xu, The long-time global behavior of time discretization for fractional order Volterra equations, Calcolo 35 (1998) 93–116.

[10] D. Xu, Non-smooth initial data error estimates with the weight norms for the linear finite element method of parabolic partial differential equations, Appl. Math. Comput. 54 (1993) 1–24.

[11] D. Xu, On the discretization in time for a parabolic integro-differential equation with a weakly singular kernel I: smooth initial data, Appl. Math. Comput. 57 (1993) 29–60.

[12] D. Xu, On the discretization in time for a parabolic integro-differential equation with a weakly singular kernel II: nonsmooth initial data, Appl. Math. Comput. 57 (1993) 1–27.

[13] D. Xu, The global behavior of time discretization for an abstract Volterra equation in Hilbert space, Calcolo 34 (1997) 71–104.

[14] Z. Sun, X. Wu, A fully discrete difference scheme for a diffusion-wave system, Appl. Numer. Math. 56 (2006) 193–209.

[15] T. Tang, Superconvergence of numerical solutions to weakly singular Volterra integro-differential equations, Numer. Math. 61 (1992) 373–382.

[16] G. Fairweather, Spline collocation methods for a class of hyperbolic partial integro-differential equations, SIAM J. Numer. Anal. (1994) 444–460.

[17] B. Bialecki, G. Fairweather, Orthogonal spline collocation methods for partial differential equations, J. Comput. Appl. Math. 128 (2001) 55–82.

[18] Y. Lin, C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, J. Comput. Phys. 225 (2007) 1533–1552.
[19] T. Lin, Y. Lin, M. Rao, S. Zhang, Petrov-Galerkin methods for linear Volterra integro-differential equations, SIAM J. Numer. Anal. (2001) 937–963.
[20] W. Long, D. Xu, X. Zeng, Quasi wavelet based numerical method for a class of partial integro-differential equation, Appl. Math. Comput. 218 (2012) 11842–11850.
[21] Blent Saka, Idris Dag ?, Dursun Irk, Quintic B-spline collocation method for numerical solution of the RLW equation, ANZIAM J. 49 (2008) 389–410.
[22] Blent Saka, Idris Dag ?, A numerical solution of the RLW equation by Galerkin method using quartic B-splines, Commun. Numer. Methods Eng. 24 (2008) 1339–1361.
[23] N. Caglar, H. Caglar, B-spline solution of singular boundary value problems, Appl. Math. Comput. 182 (2006) 1509–1513.
[24] N. Caglar, H. Caglar, B-spline method for solving linear system of second-order boundary value problems, Comput. Math. Appl. 57 (2009) 757–762.
[25] M. Lakestani, M. Dehghan, Collocation and finite difference-collocation methods for the solution of nonlinear Klein-Gordon equation, Comput. Phys. Commun. 181 (2010) 1392–1401.
[26] B. Sepehrian, M. Lashani, A numerical solution of the Burgers equation using quintic B-splines, in: Proceedings of the World Congress on Engineering, vol. III, WCE, 2008, London, UK.
[27] S.I. Zaki, A quintic B-spline finite elements scheme for the KdVB equation, Comput. Methods Appl. Eng. 188 (2000) 121–134.
[28] Idris Dag, Blent Saka, Dursun Irk, Galerkin method for the numerical solution of the RLW equation using quintic B-splines, Comput. Appl. Math. 190 (2006) 532–547.
[29] R.C. Mittal, Geeta Arora, Quintic B-spline collocation method for numerical solution of the Kuramoto-Sivashinsky equation, Commun. Nonlinear Sci. Numer. Simul. 15 (2010) 2798–2808.
[30] Min Xu, Ren-Hong Wang, Ji-Hong Zhang, A novel numerical scheme for solving Burgers' equation, Appl. Math. Comput. 217 (2011) 4473–4482.
[31] Haixiang Zhang, Xuli Han, Xuehua Yang, Quintic B-spline collocation method for fourth order partial integro-differential equations with a weakly singular kernel, Appl. Math. Comput. 219 (2013) 6565–6575.
[32] R.C. Mittal, Geeta Arora, Numerical solution of the coupled viscous Burgers' equation, Commun. Nonlinear Sci. Numer. Simul. 16 (2011) 1304–1313.
[33] R.C. Mittal, R.K. Jain, Cubic B-spline collocation method for solving nonlinear parabolic partial differential equations with Neumann boundary conditions, Commun. Nonlinear Sci. Numer. Simul. 17 (2012) 4616–4625.
[34] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, second ed., Springer-Verlag, New York, 1991.
[35] C. de Boor, On the convergence of odd degree spline interpolation, J. Approx. Theory 1 (1968) 452–463.
[36] C.A. Hall, On error bounds for spline interpolation, J. Approx. Theory 1 (1968) 209–218.
[37] P.M. Prenter, Spline and Variational Methods, Wiley, New York, 1975.
[38] Xuehua Yang, Da Xu, Haixiang Zhang, Crank-Nicolson/quasi-wavelets method for solving fourth order partial integro-differential equation with a weakly singular kernel, J. Comput. Phys. 234 (2013) 317–329.
[39] Farhad Fakhar-Izadi, Mehdi Dehghan, The spectral methods for parabolic Volterra integro-differential equations, J. Comput. Appl. Math. 235 (2011) 4032–4046.