Research Journal of Applied Sciences, Engineering and Technology 7(10): 2065-2073, 2014
DOI:10.19026/ajfst.7.500
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 28, 2013 Accepted: July 19, 2013 Published: March 15, 2014

Research Article

Cubic B-spline for the Numerical Solution of Parabolic Integro-differential Equation with a Weakly Singular Kernel

Shahid S. Siddiqi and Saima Arshed
Department of Mathematics, University of the Punjab, Lahore 54590, Pakistan

Abstract: The aim of study is to solve parabolic integro-differential equation with a weakly singular kernel. Problems involving partial integro-differential equations arise in fluid dynamics, viscoelasticity, engineering, mathematical biology, financial mathematics and other areas. Many mathematical formulations of physical phenomena contain integro-differential equations. Integro-differential equations are usually difficult to solve analytically so, it is required to obtain an efficient approximate solution. A numerical method is developed to solve the partial integro-differential equation using the cubic B-spline collocation method. The method is based on discretizing the time derivative using finite central difference formula and the cubic B-spline collocation method for the spatial derivative. Three examples are considered to illustrate the efficiency of the method developed. It is to be observed that the numerical results obtained by the proposed method efficiently approximate the exact solutions.

Keywords: Central differences, collocation method, cubic B-spline, integro-differential equation, weakly singular kernel

INTRODUCTION

Consider the following partial integro-differential equation with a weakly singular kernel:

$$\int_0^t \beta(t-s)u_t(x,s)ds - u_{xx}(x,t) = f(x,t), \qquad x \in [a,b], \quad t > 0 \tag{1}$$

Subject to the initial condition:

$$u(x,0) = g_0(x), \quad 0 \le x \le 1 \tag{2}$$

and appropriate boundary conditions:

$u(a,t) = f_0(t), \quad u(b,t) = f_1(t), \quad t \ge 0$ Dirichlet conditions

or

$u_x(a,t) = r_0(t), \quad u_x(b,t) = r_1(t), \quad t \ge 0$ Neumann conditions (3)

where, the kernel:

$$\beta(t) = \frac{t^{\alpha-1}}{\Gamma(\alpha)}, \quad 0 < \alpha < 1$$

is a singular kernel at t = 0 and Γdenotes the gamma function, g_0 (x), f_0 (t), f_1 (t), r_0 (t), r_1 (t) are known functions, f (x, t) is a given smooth function and the function u (x, t) is unknown.

The integro-differential Eq. (1) along with the constraints (2) and (3) occurs in applications such as heat conduction in material with memory (Gurtin and Pipkin, 1968; Miller, 1978), compression of poroviscoelastic media, population dynamics, nuclear reactor dynamics etc.

It can be seen that in Eq. (1), the kernel function has a weak singularity at the origin (Tang, 1993). This is particular interesting in viscoelasticity, because it might smooth the solution when the boundary data is discontinuous (Renardy, 1989).

Solution of integro-partial differential equations has recently attracted much attention of research. Chen *et al*. (1992) used finite element method for the numerical solution of a parabolic integro-differential equation with a weakly singular kernel. In Fairweather (1994), spline collocation methods have been applied to obtain the numerical solution for a class of hyperbolic partial integro-differential equations. Huang (1994) used time discretization scheme for solving integro-differential equations of parabolic type. Xu (1993a, b and c) used finite element method to solve parabolic partial integro-differential equation. Wulan and Xu (2010) used finite central difference/finite element approximations for the numerical solution of partial integro-differential equations. Soliman *et al*. (2012) used fourth order finite difference and collocation method for the numerical solution of partial integro-differential equation.

Corresponding Author: Shahid S. Siddiqi, Department of Mathematics, University of the Punjab, Lahore 54590, Pakistan

In this study, the approximate solution of parabolic integro-differential equation with weakly singular kernel is proposed using cubic B-spline collocation method. The collocation method with B-spline basis functions represents an economical alternative, since it only requires the evaluation of the unknown parameters at the grid points. Haixiang *et al*. (2013) used quintic B-spline collocation method for solving fourth order partial integro-differential equation with a weakly singular kernel.

TEMPORAL DISCRETIZATION

Consider a uniform mesh Δwith the grid points λ_{ij} to discretize the region $\Omega = [a,b] \times [0,T]$. Each λ_{ij} is the vertices of the grid point (x_i, t_j) where $x_i = a + ih$ $i = 0,1,2,...,N$ and $t_j = jk, j = 0,1,2,...,M$, $Mk = T$. The quantities h and k are the mesh sizes in the space and time directions, respectively.

A finite difference approximation is used to discretize the time derivative involved in Eq. (1) at time point $t = t_{j+1}$ as:

$$\begin{aligned}\int_0^{t_{j+1}} \frac{(t_{j+1}-s)^{\alpha-1}}{\Gamma(\alpha)} \frac{\partial u(x,s)}{\partial s} ds &= \int_{t_0}^{t_1} \frac{u(x,t_1)-u(x,t_0)}{k} \frac{(t_{j+1}-s)^{\alpha-1}}{\Gamma(\alpha)} ds \\ &+ \sum_{r=1}^{j} \int_{t_r}^{t_{r+1}} \frac{(t_{j+1}-s)^{\alpha-1}}{\Gamma(\alpha)} \frac{u(x,t_{r+1})-u(x,t_{r-1})}{2k} ds \\ &= \frac{u(x,t_1)-u(x,t_0)}{\Gamma(\alpha).k} \int_{t_0}^{t_1} \frac{1}{(t_{j+1}-s)^{1-\alpha}} ds \\ &+ \sum_{r=1}^{j} \frac{u(x,t_{r+1})-u(x,t_{r-1})}{2k\,.\Gamma(\alpha)} \int_{t_r}^{t_{r+1}} \frac{1}{(t_{j+1}-s)^{1-\alpha}} ds \\ &= \frac{u(x,t_1)-u(x,t_0)}{\alpha\Gamma(\alpha).k^{1-\alpha}} [(j+1)^\alpha - j^\alpha] \\ &+ \sum_{r=0}^{j-1} \frac{u(x,t_{j+1-r})-u(x,t_{j-r-1})}{2\alpha\Gamma(\alpha).k^{1-\alpha}} [(r+1)^\alpha - r^\alpha] \\ &= b_j \frac{u(x,t_1)-u(x,t_0)}{\Gamma(\alpha+1).k^{1-\alpha}} + \frac{1}{2\Gamma(\alpha+1)} \sum_{r=0}^{j-1} b_r \frac{u(x,t_{j+1-r})-u(x,t_{j-r-1})}{k^{1-\alpha}}\end{aligned} \tag{4}$$

where, $b_r = (r+1)^\alpha - r^\alpha$, $r = 0,1,2,..., j$.

The discrete differential operator L_t^α can be defined as:

$$L_t^\alpha u(x, t_{j+1}) = b_j \frac{u(x,t_1)-u(x,t_0)}{\Gamma(\alpha+1).k^{1-\alpha}} + \frac{1}{2\Gamma(\alpha+1)} \sum_{r=0}^{j-1} b_r \frac{u(x,t_{j+1-r})-u(x,t_{j-r-1})}{k^{1-\alpha}}$$

The Eq. (4) can be rewritten as:

$$\int_0^{t_{j+1}} \frac{(t_{j+1}-s)^{\alpha-1}}{\Gamma(\alpha)} \frac{\partial u(x,s)}{\partial s} ds \cong L_t^\alpha u(x, t_{j+1})$$

Substituting $L_t^\alpha u(x, t_{j+1})$ as an approximation of:

$$\int_0^{t_{j+1}} \frac{(t_{j+1}-s)^{\alpha-1}}{\Gamma(\alpha)} \frac{\partial u(x,s)}{\partial s} ds$$

leads to the following difference scheme to Eq. (1):

$$L_t^\alpha u(x, t_{j+1}) - u_{xx}(x, t_{j+1}) \cong f(x, t_{j+1})$$

It can further be written as:

$$\frac{b_j}{\Gamma(\alpha+1)} \frac{u(x,t_1)-u(x,t_0)}{k^{1-\alpha}} + \frac{1}{2\Gamma(\alpha+1)} \sum_{r=0}^{j-1} b_r \frac{u(x,t_{j-r+1})-u(x,t_{j-r-1})}{k^{1-\alpha}} - u_{xx}(x,t_{j+1}) \cong f(x,t_{j+1}) \tag{5}$$

The above equation can be rewritten as:

$$\begin{aligned} b_0\, u^{j+1}(x) - 2\Gamma(\alpha+1) k^{1-\alpha} \frac{\partial^2 u^{j+1}}{\partial x^2} &= b_0\, u^{j-1}(x) - \sum_{r=1}^{j-1} b_r (u^{j-r+1}(x) - u^{j-r-1}(x)) \\ &- 2 b_j (u^1(x) - u^0(x)) + 2\Gamma(\alpha+1) k^{1-\alpha} f^{j+1}(x) \end{aligned} \tag{6}$$

where,

$$u^{j+1}(x) = u(x, t_{j+1}),\ b_r = (r+1)^\alpha - r^\alpha,\ r = 0,1,2,..., j.$$

Note that $b_0 = 1$ and let $a_0 = 2\Gamma(\alpha+1) k^{1-\alpha}$, then the right hand side of Eq. (6) can be reformulated as:

$$u^{j+1}(x) - a_0 \frac{\partial^2 u^{j+1}}{\partial x^2} = -b_1 u^j(x) + \sum_{r=1}^{j-1} (b_{r-1} - b_{r+1})\, u^{j-r}(x) - b_j u^1(x) + (b_{j-1} + 2b_j)\, u^0(x) + a_0 f^{j+1}(x), j \geq 1 \tag{7}$$

with the boundary conditions:

$$u^{j+1}(a) = f_0(t_{j+1}), \qquad u^{j+1}(b) = f_1(t_{j+1}) \tag{8}$$

In each time level, there is an ordinary differential equation in the form of Eq. (7) with the boundary conditions Eq. (8), which is solved by cubic B-spline collocation method. The proposed scheme Eq. (7) is a three level scheme. In order to apply the proposed scheme, it is necessary to have the values of *u* at the nodal points at the zeroth (u^0) and first (u^1) level times.

To compute u^1 substitute j = 0 (the special case), in Eq. (5), it can be written as:

$$u^1(x) - \frac{1}{2} a_0 \frac{\partial^2 u^1}{\partial x^2} = u^0(x) + \frac{1}{2} a_0 f^1(x) \tag{9}$$

where, $u^0 = u(x,0) = g_0(x)$ is the value of u at the zeroth level time (the initial condition).

CUBIC B-SPLINE COLLOCATION METHOD

Let $\Delta^* = \{a = x_0 < x_1 < x_2 < ... < x_N = b\}$ be the partition of [a, b]. Let B_i be B-spline basis functions with knots at the points x_i, i = 0, 1,…, N. Thus, an approximation U^{j+1} (x) to the exact solution U^{j+1} (x)

Table 1: Coefficient of cubic B-spline and its derivatives at knots x_i

	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}	else
$B_i(x)$	0	1	4	1	0	0
$B_i^{(1)}(x)$	0	3/h	0	-3/h	0	0
$B_i^{(2)}(x)$	0	$6/h^2$	$-12/h^2$	$6/h^2$	0	0

At j+1 time level, can be expressed in terms of the cubic B-spline basis functions $B_i(x)$ as:

$$U^{j+1}(x) = \sum_{i=-1}^{N+1} c_i(t) B_i(x) \tag{10}$$

where, c_i are unknown time dependent quantities to be determined from the boundary conditions and collocation form of the integro-differential equation.

The cubic B-spline $B_i(x)$, $i = -1, 0, ..., N+1$ can be defined as under:

$$B_i(x) = \frac{1}{h^3}\begin{cases} (x - x_{i-2})^3, & x \in [x_{i-2}, x_{i-1}], \\ h^3 + 3h^2(x - x_{i-1}) + 3h(x - x_{i-1})^2 - 3(x - x_{i-1})^3, & x \in [x_{i-1}, x_i], \\ h^3 + 3h^2(x_{i+1} - x) + 3h(x_{i+1} - x)^2 - 3(x_{i+1} - x)^3, & x \in [x_i, x_{i+1}], \\ (x_{i+2} - x)^3, & x \in [x_{i+1}, x_{i+2}], \\ 0, & otherwise \end{cases}$$

The values of successive derivatives $B_i^{(r)}(x)$, $i = -1, ..., N+1; r = 0,1,2$ at nodes, are listed in Table 1.

Let, $U^{j+1}(x)$ satisfies the boundary conditions:

$$U^{j+1}(a) = f_0(t_{j+1}), \qquad U^{j+1}(b) = f_1(t_{j+1})$$

and the collocation equations:

$$U^{j+1}(x_i) - a_0 \frac{\partial^2 U^{j+1}}{\partial x^2} = -b_1 U^j(x_i) + \sum_{r=1}^{j-1}(b_{r-1} - b_{r+1}) U^{j-r}(x_i)$$
$$-b_j U^1(x_i) + (b_{j-1} + 2b_j) U^0(x_i) + a_0 f^{j+1}(x_i), \quad j \geq 1, \quad i = 0,1,2,...,N$$

The above equation can be rewritten, by omitting the dependence of $U^{j+1}(x)$ on x as:

$$U_i^{j+1} - a_0 \frac{\partial^2 U_i^{j+1}}{\partial x^2} = -b_1 U_i^j + \sum_{r=1}^{j-1}(b_{r-1} - b_{r+1}) U_i^{j-r}$$
$$-b_j U_i^1 + (b_{j-1} + 2b_j) U_i^0 + a_0 f_i^{j+1} \quad j \geq 1, \quad i = 0,1,2,...,N \tag{11}$$

Substituting Eq. (10) into Eq. (11), it can be written as:

$$(c_{i-1}^{j+1} + 4c_i^{j+1} + c_{i+1}^{j+1}) - a_0 \frac{6}{h^2}(c_{i-1}^{j+1} - 2c_i^{j+1} + c_{i+1}^{j+1})$$
$$= -b_1(c_{i-1}^j + 4c_i^j + c_{i+1}^j) + \sum_{r=1}^{j-1}(b_{r-1} - b_{r+1})(c_{i-1}^{j-r} + 4c_i^{j-r} + c_{i+1}^{j-r})$$
$$-b_j(c_{i-1}^1 + 4c_i^1 + c_{i+1}^1) + (b_{j-1} + 2b_j)(c_{i-1}^0 + 4c_i^0 + c_{i+1}^0) + a_0 f_i^{j+1} \tag{12}$$

Simplifying the above relation leads to the following system of (N+1) linear equations in (N+3) unknowns $c_{-1}^{j+1}, c_0^{j+1}, c_1^{j+1}, ..., c_N^{j+1}, c_{N+1}^{j+1}$.

$$\left(1 - a_0\frac{6}{h^2}\right)c_{i-1}^{j+1} + \left(4 + a_0\frac{12}{h^2}\right)c_i^{j+1} + \left(1 - a_0\frac{6}{h^2}\right)c_{i+1}^{j+1} = F_i, \quad j \geq 1, \quad i = 0,1,2,...,N \tag{13}$$

where,

$$F_i = -b_1(c_{i-1}^j + 4c_i^j + c_{i+1}^j) + \sum_{r=1}^{j-1}(b_{r-1} - b_{r+1})(c_{i-1}^{j-r} + 4c_i^{j-r} + c_{i+1}^{j-r})$$
$$-b_j(c_{i-1}^1 + 4c_i^1 + c_{i+1}^1) + (b_{j-1} + 2b_j)(c_{i-1}^0 + 4c_i^0 + c_{i+1}^0) + a_0 f_i^{j+1}$$

To obtain the unique solution of the system (13), two additional constraints are required. These constraints are obtained from the boundary conditions. Imposition of the boundary conditions enables us to eliminate the parameters c_{-1} and c_{N+1} from the system (13).

First the Dirichlet boundary conditions are used in order to eliminate c_{-1} and c_{N+1}, as:

$$u(a,t) = (c_{-1} + 4c_0 + c_1) = f_0(t)$$
$$u(b,t) = (c_{N-1} + 4c_N + c_{N+1}) = f_1(t)$$

$$c_{-1} = -4c_0 - c_1 + f_0(t)$$
$$c_{N+1} = -4c_N - c_{N-1} + f_1(t)$$

After eliminating c_{-1} and c_{N+1}, the system (13) is reduced to a tri-diagonal system of (N+1) linear equations in (N+1) unknowns. This system can be rewritten in matrix form as:

$$A\,C^{j+1} = F, \quad j = 1,2,3,... \tag{14}$$

where,

$$C^{j+1} = [c_0^{j+1}, c_1^{j+1}, ..., c_N^{j+1}]^T, \qquad j = 1,2,3,...$$

The coefficient matrix A is given as under:

$$A = \begin{bmatrix} a_0\frac{36}{h^2} & & & & & & \\ \alpha & \beta & \alpha & & & & \\ & \alpha & \beta & \alpha & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & \alpha & \beta & \alpha \\ & & & & & & a_0\frac{36}{h^2} \end{bmatrix}$$

where,

$$\alpha = \left(1 - a_0\frac{6}{h^2}\right), \beta = \left(4 + a_0\frac{12}{h^2}\right)$$

The Neumann boundary conditions can also be applied in order to eliminate c_{-1} and c_{N+1}, as:

$$u_x(a,t)=\frac{3}{h}(-c_{-1}+c_1)=r_0(t)$$
$$u_x(b,t)=\frac{3}{h}(-c_{N-1}+c_{N+1})=r_1(t)$$
$$c_{-1}=c_1-\frac{h}{3}r_0(t)$$
$$c_{N+1}=c_{N-1}+\frac{h}{3}r_1(t)$$

After eliminating c_{-1} and c_{N+1}, the system (13) is reduced to a tri-diagonal system of (N+1) linear equations in (N+1) unknowns. This system can be rewritten in matrix form as:

$$A\,C^{j+1}=F, \quad j=1,2,3,... \tag{15}$$

where,

$$C^{j+1}=[c_0^{j+1},c_1^{j+1},...,c_N^{j+1}]^T, \qquad j=1,2,3,...$$

The coefficient matrix A is given as under:

$$A=\begin{bmatrix} \left(4+a_0\frac{12}{h^2}\right) & \left(2-a_0\frac{12}{h^2}\right) & & & & \\ \gamma & \delta & \gamma & & & \\ & \gamma & \delta & \gamma & & \\ & & \ddots & \ddots & \ddots & \\ & & & \gamma & \delta & \gamma \\ & & & & \left(2-a_0\frac{12}{h^2}\right) & \left(4+a_0\frac{12}{h^2}\right) \end{bmatrix}$$

where,

$$\gamma=\left(1-a_0\frac{6}{h^2}\right),\ \delta=\left(4+a_0\frac{12}{h^2}\right)$$

Using the system (13), for j = 1, following is the system of ($N+1$) linear equations in ($N+3$) unknowns $c_{-1}^2,c_0^2,c_1^2,...,c_N^2,c_{N+1}^2$.

$$\left(1-a_0\frac{6}{h^2}\right)c_{i-1}^2+\left(4+a_0\frac{12}{h^2}\right)c_i^2+\left(1-a_0\frac{6}{h^2}\right)c_{i+1}^2=F_i, \qquad i=0,1,2,...,N \tag{16}$$

where,

$$F_i=-2b_1(c_{i-1}^1+4c_i^1+c_{i+1}^1)+(b_0+2b_1)(c_{i-1}^0+4c_i^0+c_{i+1}^0)+a_0f_i^2$$

In order to find the value of $C^2=[c_0^2,c_1^2,...,c_N^2]^T$, it is first needed to find the value of $C^1=[c_0^1,c_1^1,...,c_N^1]^T$. The value of C^1 is obtained, solving Eq. (9) using cubic B-spline collocation method, as:

$$\left(1-a_0\frac{3}{h^2}\right)c_{i-1}^1+\left(4+a_0\frac{6}{h^2}\right)c_i^1+\left(1-a_0\frac{3}{h^2}\right)c_{i+1}^1=F_i, \qquad i=0,1,2,...,N \tag{17}$$

where,

$$F_i=(c_{i-1}^0+4c_i^0+c_{i+1}^0)+\frac{1}{2}a_0f_i^1$$

The above Eq. (17) is a system of ($N+1$) linear equations in ($N+3$) unknowns $c_{-1}^1,c_0^1,c_1^1,...,c_N^1,c_{N+1}^1$. To obtain the unique solution of this system, eliminate c_{-1} and c_{N+1}, using Dirichlet and Neumann boundary conditions.

The time evolution of the approximate solution U^{j+1} is determined by the time evolution of the vector C^{j+1}. This is found by repeatedly solving the recurrence relationship, once the initial vector $C^0=[c_0^0,c_1^0,...,c_N^0]^T$, has been computed from the initial condition. The recurrence relationship is tri-diagonal and so can be solved using Thomas algorithm.

The initial state vector: The initial state vector $\mathbf{C}^0$ can be determined from the initial condition $u(x,0)=u^0(x)=g_0(x)$ which gives ($N+1$) equations in ($N+3$) unknowns. For determining these unknowns the following relations at the knots are used:

$$U_x(a,0)=u_x(x_0,0),$$
$$U(x_i,0)=u^0(x_i), \qquad i=1,2,3,...,N-1$$
$$U_x(b,0)=u_x(x_N,0)$$

which gives a tri-diagonal system of equations in the following matrix:

$$G\,C^0=E$$

where,

$$G=\begin{bmatrix} 4 & 2 & 0 & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & 0 & 2 & 4 \end{bmatrix}$$

NUMERICAL RESULTS

The proposed method is tested on the following three problems. Let:

Table 2: The errors $||e_M||_\infty$ and $||e_M||_2$ when N = 60 and k = 0.0001 for example 1

M	$\|\|e_M\|\|_\infty$	$\|\|e_M\|\|_2$
10	4.1248 E-05	3.7654 E-06
20	2.3848 E-05	2.1770 E-06
30	1.2071 E-05	1.1019 E-06
40	3.1259 E-06	2.8536 E-07
50	4.0669 E-06	3.7126 E-07

Table 3: The errors $||e_M||_\infty$ and $||e_M||_2$ when N = 60 and k = 0.001 for example 1

M	$\|\|e_M\|\|_\infty$	$\|\|e_M\|\|_2$
10	8.7764 E-04	8.0117 E-05
20	8.5436 E-04	7.7992 E-05
30	8.4063 E-04	7.6738 E-05
40	8.3093 E-04	7.5853 E-05
50	8.2337 E-04	7.5163 E-05

Table 4: The errors $||e_M||_\infty$ and $||e_M||_2$ when M = 10 and k = 0.0001 for example 1

N	$\|\|e_M\|\|_\infty$	$\|\|e_M\|\|_2$
10	2.0176 E-03	4.5115 E-04
20	4.2892 E-04	6.7818 E-05
30	1.3503 E-04	1.7433 E-05
40	3.2197 E-05	3.5998 E-06
50	1.5396 E-05	1.5396 E-06

Table 5: The errors $||e_M||_\infty$ and $||e_M||_2$ when N = 60 and k = 0.0001 for example 2

M	$\|\|e_M\|\|_\infty$	$\|\|e_M\|\|_2$
10	9.1411 E-05	6.8482 E-06
20	9.1003 E-05	5.3983 E-06
30	9.0654 E-05	5.5083 E-06
40	8.9746 E-05	5.5048 E-06
50	8.8599 E-05	5.4441 E-06

Table 6: The errors $||e_M||_\infty$ and $||e_M||_2$ when N = 60 and k = 0.001 for example 2

M	$\|\|e_M\|\|_\infty$	$\|\|e_M\|\|_2$
10	9.9041 E-04	8.5887 E-05
20	9.8809 E-04	8.5793 E-05
30	9.8464 E-04	8.5511 E-05
40	9.8077 E-04	8.5187 E-05
50	9.7666 E-04	8.4844 E-05

Table 7: The errors $||e_M||_\infty$ and $||e_M||_2$ when M = 10 and k = 0.0001 for example 2

N	$\|\|e_M\|\|_\infty$	$\|\|e_M\|\|_2$
10	8.5855 E-04	1.8436 E-04
20	1.6181 E-04	2.0645 E-05
30	4.0921 E-05	2.9628 E-06
40	1.3212 E-06	1.1620 E-06

$$t_j = jk\,,\, j = 0,1,2,...,M\,,\; h = \frac{1}{N},$$

where M denotes the final time level t_M and N+1 is the number of nodes. In order to check the accuracy of the proposed method, the maximum norm errors and L_2 norm errors between numerical and exact solution are given with the following definitions:

Maximum norm error: $$\|e_M\|_\infty = \max_{0\le i\le N}\left|u(x_i,t_M) - U_i^M\right|$$

L_2 norm error: $$\|e_M\|_2 = \frac{1}{N}\left(\sum_{i=0}^{N}\left|u(x_i,t_M) - U_i^M\right|^2\right)^{\frac{1}{2}}$$

Example 1: Following is the second order parabolic integro-differential equation:

$$\int_0^t \frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)} u_t(x,s)ds - u_{xx}(x,t) = f(x,t), \qquad x\in[0,1],\;\; t>0,\;\; \alpha = 0.5 \tag{18}$$

with the initial condition:

$$u(x,0) = \sin\pi x\,, \qquad x\in[0,1]$$

and boundary conditions:

$$u(0,t) = 0 = u(1,t), \qquad t\ge 0$$

The exact solution of the problem is:

$$u(x,t) = (t+1)\sin\pi x$$

The numerical solutions at N = 60, k = 0.0001 and k = 0.001, with different time levels M, are presented in Table 2 and 3 respectively. The numerical solutions at M = 10 and k = 0.0001 for different values of N are tabulated in Table 4. In Table 2 to 4, the time increment k, the space increment $h = \frac{1}{N}$ and time level M are varied to test the accuracy of the proposed method, which indicates that the proposed method is substantially efficient.

In order to indicate the effect of the proposed method for larger M, the exact solution and the numerical solution are plotted using N = 100, M = 500 and k = 0.0001 as shown in Fig. 1. When N = 100, k = 0.0001 and M = 10 the exact solution and the numerical solution at the M time level are shown in Fig. 2. It can be observed from the Table 2 to 4 and Fig. 1 and 2, that the proposed method approximates the exact solution very efficiently.

Example 2: Following is the parabolic integro-differential equation:

$$\int_0^t \frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)} u_t(x,s)ds - u_{xx}(x,t) = f(x,t), \qquad x\in[0,1],\;\; t>0,\;\; \alpha = 0.5 \tag{19}$$

with the initial condition:

$$u(x,0) = \cos\pi x\,, \qquad x\in[0,1]$$

and Dirichlet boundary conditions:

$$u(0,t) = (t+1),$$
$$u(1,t) = (t+1)\cos(\pi), \qquad t\ge 0$$

The exact solution of the problem is:

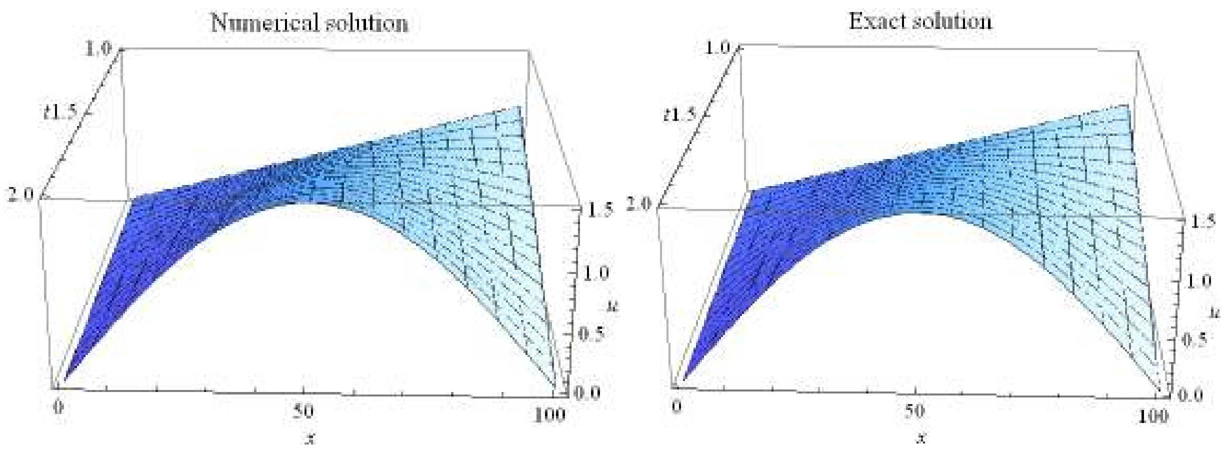

Fig. 1: The results at M = 500 for example 1

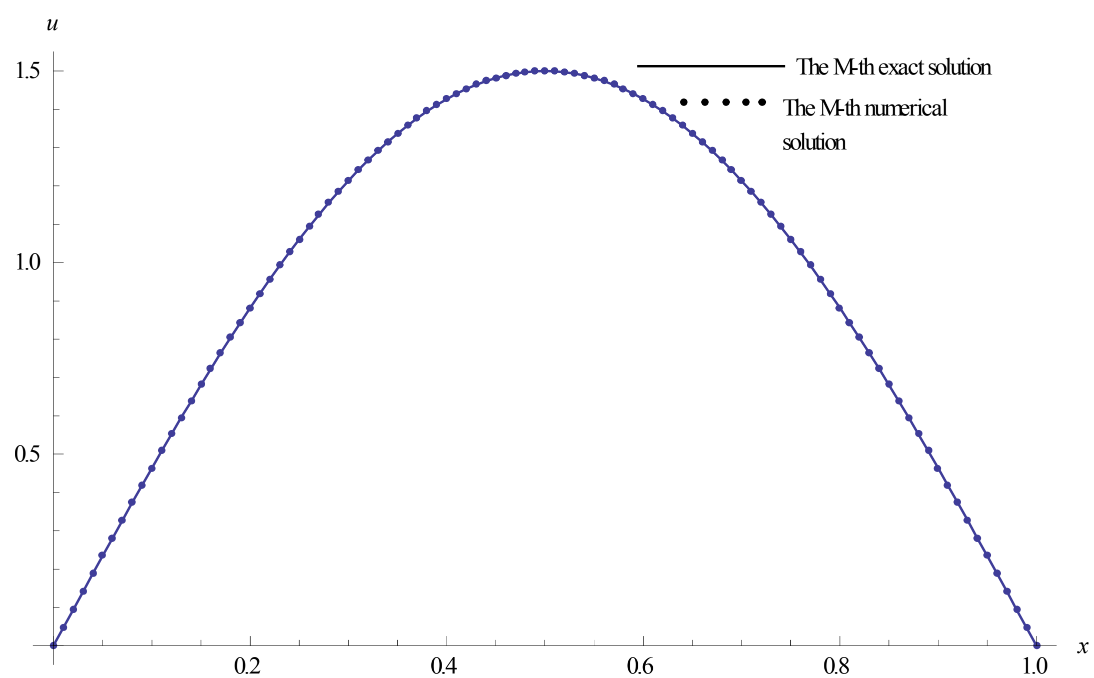

Fig. 2: The exact and numerical solutions at M = 10

$$u(x,t) = (t+1)\cos \pi x$$

The numerical solutions at N = 60, k = 0.0001 and k = 0.001, with different time levels M, are presented in Table 5 and 6 respectively. The numerical solutions at M = 10 and k = 0.0001 for different values of N are tabulated in Table 7. In Table 5 to 7, the time increment k, the space increment $h = \frac{1}{N}$ and time level M are varied to test the accuracy of the proposed method, which indicates that the proposed method is substantially efficient.

In order to indicate the effect of the proposed method for larger M, the exact solution and the numerical solution are plotted using N = 100, M = 500 and k = 0.0001 as shown in Fig. 3. When N = 100, k = 0.0001 and M = 10 the exact solution and the numerical solution at the M time level are shown in Fig. 4. It can be observed from the Table 5 to 7 and Fig. 3 and 4, that the proposed method approximates the exact solution very efficiently.

Example 3: Following is the parabolic integro-differential equation:

$$\int_0^t \frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)} u_t(x,s)ds - u_{xx}(x,t) = f(x,t), \qquad x \in [-1,1], \ \ t>0, \ \ \alpha = 0.5 \tag{20}$$

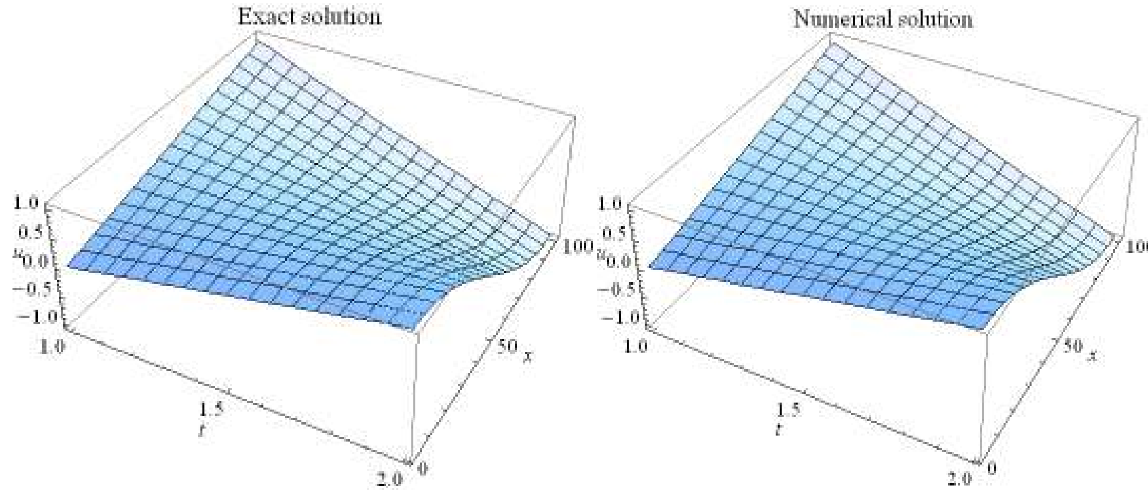

Fig. 3: The results at M = 500 for example 2

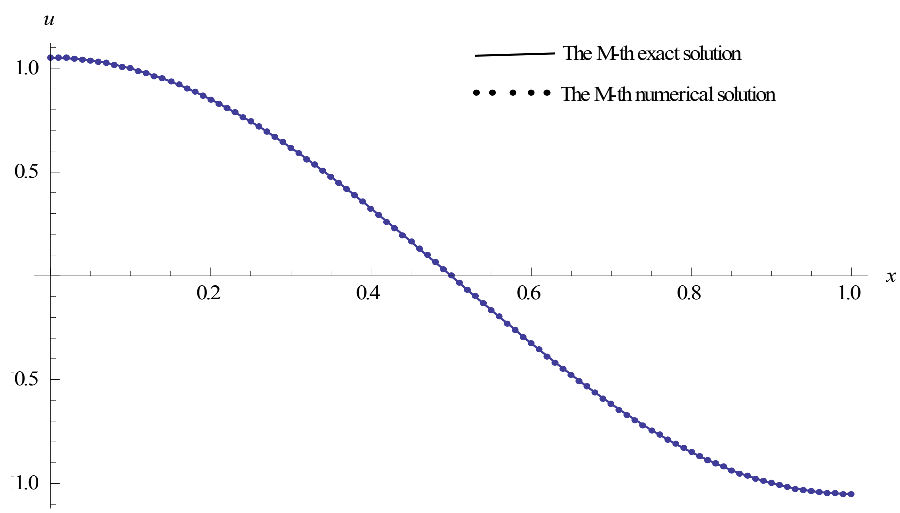

Fig. 4: The exact and numerical solutions at M = 10

with the initial condition:

$$u(x,0) = \sin \pi x, \qquad x \in [-1,1]$$

and Neumann boundary conditions:

$$u_x(-1,t) = \pi(t+1)^2 \cos \pi,$$
$$u_x(1,t) = \pi(t+1)^2 \cos(\pi), \qquad t \geq 0$$

The exact solution of the problem is:

$$u(x,t) = (t+1)^2 \sin \pi x$$

The numerical solutions at $N = 40$, $k = 0.001$ and $k = 0.00125$, with different time levels M, are presented in Table 8 and 9 respectively. In Table 8 and 9, the time increment k and time level M are varied to test the accuracy of the proposed method, which indicates that the proposed method is substantially efficient.

Table 8: Maximum norm errors $||e_M||_\infty$ for N = 40 for example 3

N	M	k = 0.001 $\|\|e_M\|\|_\infty$	k = 0.00125 $\|\|e_M\|\|_\infty$
40	10	5.9948 E-04	1.0018 E-03
	20	4.3331 E-04	7.2357 E-04
	30	7.0620 E-04	1.1162 E-04
	40	1.3169 E-03	1.9388 E-03
	50	2.0042 E-03	2.8735 E-03

Table 9: L_2 norm errors $||e_M||_2$ for N = 40 for example 3

N	M	k = 0.001 $\|\|e_M\|\|_2$	k = 0.00125 $\|\|e_M\|\|_2$
40	10	6.2565 E-05	1.0570 E-04
	20	4.6637 E-05	7.7196 E-05
	30	3.3271 E-05	5.7804 E-05
	40	8.9371 E-05	1.2795 E-04
	50	1.6722 E-04	2.3443 E-04

In order to indicate the effect of the proposed method for larger M, the exact solution and the

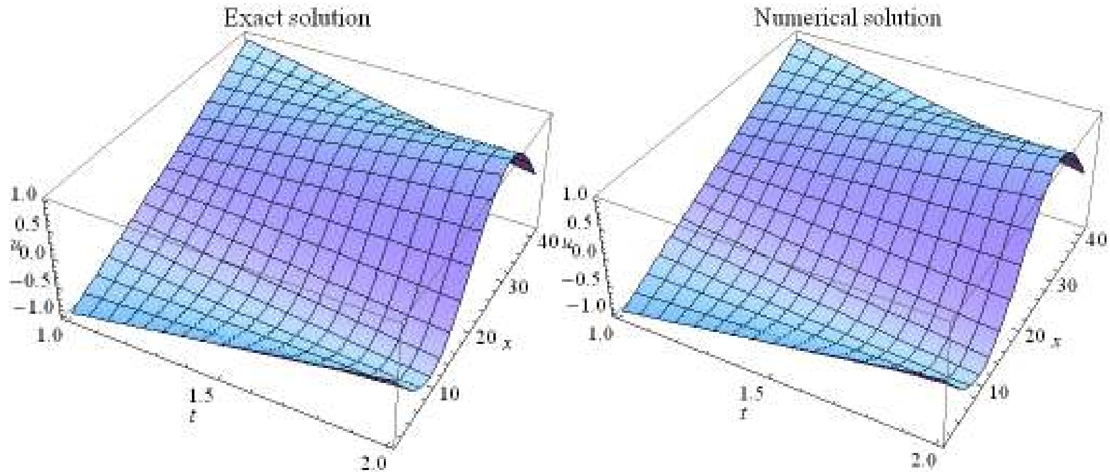

Fig. 5: The results at M = 500 for example 3

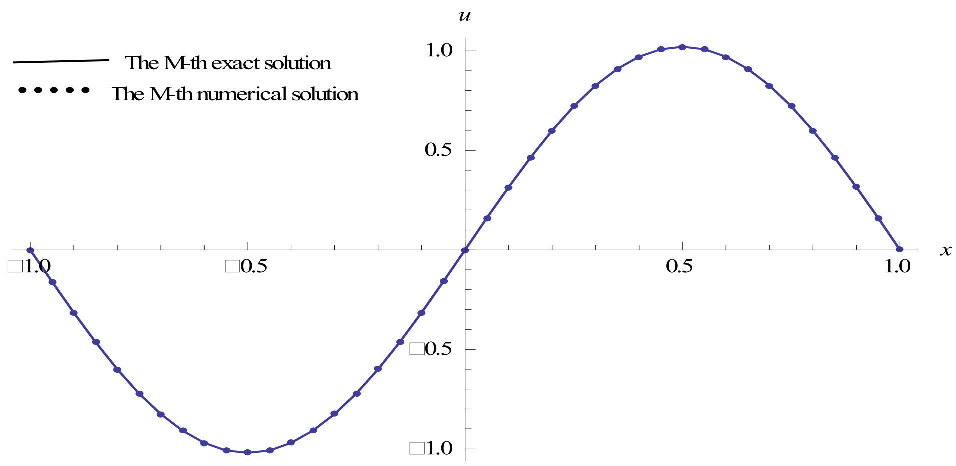

Fig. 6: The exact and numerical solutions at M = 10

numerical solution are plotted using $N = 100$, $M = 500$ and $k = 0.0001$ as shown in Fig. 5. When $N = 100$, $k = 0.0001$ and $M = 10$ the exact solution and the numerical solution at the M time level are shown in Fig. 6. It can be observed from the Table 8 and 9 and Fig. 5 and 6, that the proposed method approximates the exact solution very efficiently.

CONCLUSION

The numerical solution of parabolic integro-differential equation with a weakly singular kernel is studied using cubic B-spline collocation method. The parabolic integro-differential equation is discretized by the finite central difference formula in the time direction and the cubic B-spline collocation method for spatial derivative. The parameters h, k and M are varied in order to test the accuracy of the proposed method. It is observed from the numerical experiments, that the proposed method possesses high degree of efficiency and accuracy. Moreover, the numerical results are in good agreement with the exact solutions. The numerical solutions of non-linear parabolic integro-differential equations are in progress.

REFERENCES

Chen, C., V. Thomée and L.B. Wahlbin, 1992. Finite element approximation of a parabolic integro-differential equation with a weakly singular kernel. Math Comput., 58: 587-602.

Fairweather, G., 1994. Spline collocation methods for a class of hyperbolic partial integro-differential equations. SIAM J. Numer. Anal., 31: 444-460.

Gurtin, M.E. and A.C. Pipkin, 1968. A general theory of heat conduction with finite wave speed. Arch. Ration. Mech. An., 31: 113-126.

Haixiang, Z., H. Xuli and Y. Xuehua, 2013. Quintic B-spline collocation method for fourth order partial integro-differential equations with a weakly singular kernel. Appl. Math. Comput., 219: 6565-6575.

Huang, Y.Q., 1994. Time discretization scheme for an integro-differential equation of parabolic type. J. Comput. Math., 12: 259-263.

Miller, R.K., 1978. An integro-differential equation for rigid heat conductors with memory. J. Math. Anal. Appl., 66: 313-332.

Renardy, M., 1989. Mathematical analysis of viscoelastic flows. Annu. Rev. Fluid Mech., 21: 21-36.

Soliman, A.F., A.M.A. EL-Asyed and M.S. El-Azab, 2012. On the numerical solution of partial integro-differential equations. Math. Sci. Lett., 1: 71-80.

Tang, T., 1993. A finite difference scheme for partial integro-differential equations with a weakly singular kernel. Appl. Numer. Math., 11: 309-319.

Wulan, L. and D. Xu, 2010. Finite central difference/finite element approximations for parabolic integro-differential equations. Computing, 90: 89-111.

Xu, D., 1993a. On the discretization in time for a parabolic integro-differential equation with a weakly singular kernel I: Smooth initial data. Appl. Math. Comput., 58: 1-27.

Xu, D., 1993b. On the discretization in time for a parabolic integrodifferential equation with a weakly singular kernel II: Nonsmooth initial data. Appl. Math. Comput., 58: 29-60.

Xu, D., 1993c. Finite element methods for the nonlinear integro-differential equations. Appl. Math. Comput., 58: 241-273.