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:

\[
\int_0^t \beta(t-s)u_s(x,s)ds - u_t(x,t) = f(x,t), \quad x \in [a,b], \quad t > 0
\]

Subject to the initial condition:

\[
u(x,0) = g_0(x), \quad 0 \leq x \leq 1
\]

and appropriate boundary conditions:

\[
u(a,t) = f_0(t), \quad u(b,t) = f_1(t), \quad t \geq 0 \quad \text{Dirichlet conditions}
\]

or

\[
u_0(a,t) = r_0(t), \quad u_1(b,t) = r_1(t), \quad t \geq 0 \quad \text{Neumann conditions}
\]

where, the kernel:

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

is a singular kernel at \( t = 0 \) and \( \Gamma \) denotes the gamma function, \( g_0(x), f_0(t), f_1(t), r_0(t), r_1(t) \) are known functions, \( \Gamma(x,t) \) 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).

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 $\Delta t$ with the grid points $\lambda_{ij}$ to discretize the region $\Omega=[a, b] \times [0, T]$. Each $\lambda_{ij}$ is the vertices of the grid point $(x_i, t_j)$ where $x_i = a + ih$ and $t_j = j \Delta t$, $j = 0, 1, 2, ..., N$. 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:

$$
\left( \sum_{i=0}^{N_j} \frac{(t_{j+1} - s)^{\alpha_i}}{\Gamma(\alpha_i)} \frac{d u(x,s)}{ds} \right)_{t=s^j} = \sum_{i=0}^{N_j} \frac{(t_{j+1} - s)^{\alpha_i}}{\Gamma(\alpha_i)} \frac{d^2 u(x,s)}{ds^2} + \sum_{i=0}^{N_j} \frac{(t_{j+1} - s)^{\alpha_i}}{\Gamma(\alpha_i)} u(x,s) - \sum_{i=0}^{N_j} \frac{(t_{j+1} - s)^{\alpha_i}}{\Gamma(\alpha_i)} u(x,s) = \sum_{i=0}^{N_j} \frac{(t_{j+1} - s)^{\alpha_i}}{\Gamma(\alpha_i)} \int_0^{t_{j+1}} (t_{j+1} - s)^{\alpha_i} \frac{d u(x,s)}{ds} ds \equiv L^a_t u(x, t_{j+1})
$$

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

$$
L^a_t u(x, t_{j+1}) - u_{xx}(x, t_{j+1}) = f(x, t_{j+1})
$$

It can further be written as:

$$
b_j u^{i+1}_{j+1} - 2 \Gamma(\alpha + 1) k^{\alpha} \frac{\partial u}{\partial x} = b_0 u^{i+1}_{j+1} - \sum_{i=0}^{N_j} b_i (u^{i+1}(x) - u^{i+1}(x)) - 2 b_j (u^i(x) - u^i(x)) + 2 \Gamma(\alpha + 1) k^{\alpha} f^{i+1}(x)
$$

where,

$$
u^{i+1}(x) = u(x, t_{j+1}), \quad b_j = (r+1)^a - r^a, \quad r = 0, 1, 2, ..., j.
$$

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

$$
u^{i+1}(x) - a_0 \frac{\partial^2 u}{\partial x^2} = -b_0 v^{i+1}(x) + \sum_{r=1}^{N_j} (b_r - b_{r-1}) u^r(x) + (b_j + 2 b_0) v^0(x) + a_0 f^{i+1}(x), j \geq 1
$$

with the boundary conditions:

$$
u^{i+1}(a) = f_0(t_{j+1}), \quad u^{i+1}(b) = f_j(t_{j+1})
$$

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:

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

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

**CUBIC B-SPLINE COLLOCATION METHOD**

Let $\Delta t' = |a = x_0 < x_1 < x_2 < ... < x_N = b|$ 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 $U^{i+1}(x)$ to the exact solution $U^{i+1}(x)$
At \( j+1 \) time level, can be expressed in terms of the cubic B-spline basis functions \( B_i(x) \) as:

\[
U^{j+1}(x) = \sum_{i=1}^{N+1} c_i(t) B_i(x)
\]  

(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 \( B_i(x) \), \( i = -1, 0, \ldots, N+1 \) can be defined as under:

\[
B_i(x) = \begin{cases} 
(x-x_i)^3, & x \in [x_i, x_{i+1}] \\
\frac{h}{6}[(x-x_i)^3 + 3h(x-x_i)^2 + 3h(x-x_i)x], & x \in [x_i, x_{i+1}] \\
\frac{h}{6}[(x-x_i)^3 - 3h(x-x_i)^2 + 3h(x-x_i)x], & x \in [x_i, x_{i+1}] \\
0, & \text{otherwise}
\end{cases}
\]

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

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

\[
U^{j+1}(a) = f_o(t_{j+1}), \quad U^{j+1}(b) = f_1(t_{j+1})
\]

and the collocation equations:

\[
U^{(r)}(x_i) = -b_i U^{(r-1)}(x_i) + \sum_{j=0}^{N+1} b_{ij} U^{(r-1)}(x_j), \quad r \geq 1, \quad i = 0, 1, 2, \ldots, N
\]

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

\[
U^{(r-1)}(x_i) = -b_i U^{(r-1)}(x_i) + \sum_{j=0}^{N+1} b_{ij} U^{(r-1)}(x_j), \quad r \geq 2, \quad i = 0, 1, 2, \ldots, N
\]  

(11)

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

\[
(c_i^{j+1} + 4c_i^{j+1} + c_i^{j+1}) - a_x \frac{6}{h^3} (c_i^{j+1} - 2c_i^{j+1} + c_i^{j+1})
\]

\[
= -b_i (c_{i-1}^{j+1} + 4c_i^{j+1} + c_{i+1}^{j+1}) + \sum_{j=0}^{N+1} b_{ij} (c_{i-1}^{j+1} + 4c_i^{j+1} + c_{i+1}^{j+1})
\]

\[
- \sum_{j=0}^{N+1} b_{ij} (c_{i-1}^{j+1} + 4c_i^{j+1} + c_{i+1}^{j+1}) + (b_{j+1} + 2b_j) (c_i^{0} + 4c_i^{0} + c_i^{0}) + a_i f_i^{j+1}
\]  

(12)

Simplifying the above relation leads to the following system of \((N+1)\) linear equations in \((N+3)\) unknowns \( c_{i-1}^{j+1}, c_i^{j+1}, c_{i+1}^{j+1}, \ldots, c_N^{j+1} \), \( i = 0, 1, 2, \ldots, N \):

\[
\begin{bmatrix}
1 - \frac{6}{h^3} a_x \beta a_x & 1 - \frac{6}{h^3} a_x \beta a_x & \cdots & 1 - \frac{6}{h^3} a_x \beta a_x \\
1 - \frac{6}{h^3} a_x \beta a_x & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
1 - \frac{6}{h^3} a_x \beta a_x & \cdots & \cdots & \cdots \\
\end{bmatrix}
\begin{bmatrix}
F_0^{j+1} \\
F_1^{j+1} \\
\vdots \\
F_{N-1}^{j+1} \\
\end{bmatrix}
\]

(13)

where,

\[
F_i = -b_i (c_{i-1}^{j+1} + 4c_i^{j+1} + c_{i+1}^{j+1}) + \sum_{j=0}^{N+1} b_{ij} (c_{i-1}^{j+1} + 4c_i^{j+1} + c_{i+1}^{j+1}) + (b_{j+1} + 2b_j) (c_i^{0} + 4c_i^{0} + c_i^{0}) + a_i f_i^{j+1}
\]

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:

\[
\begin{align*}
 u(a,t) &= (c_{-1} + 4c_0 + c_1) = f_o(t) \\
u(b,t) &= (c_{N-1} + 4c_N + c_{N-1}) = f_1(t)
\end{align*}
\]

\[
\begin{align*}
c_{-1} &= -4c_0 - c_1 + f_o(t) \\
c_{N+1} &= -4c_N - c_{N+1} + f_1(t)
\end{align*}
\]

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:

\[
A C^{j+1} = F, \quad j = 1, 2, 3, \ldots
\]  

(14)

where,

\[
C^{j+1} = [c_0^{j+1}, c_1^{j+1}, \ldots, c_N^{j+1}]^T, \quad j = 1, 2, 3, \ldots
\]

The coefficient matrix \( A \) is given as under:

\[
A = \begin{bmatrix}
a_x \frac{36}{h^3} & \alpha & \beta & \alpha & \cdots & \cdots & \cdots \\
\beta & a_x \frac{36}{h^3} & \alpha & \beta & \alpha & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots \\
\alpha & \cdots & \cdots & \cdots & \cdots & \cdots & \alpha & \beta & \alpha \\
\end{bmatrix}
\]

\[
\alpha = \left(1 - \frac{6}{h^3} a_x \beta a_x\right), \quad \beta = \left(4 + a_x \frac{12}{h^3}\right)
\]
The Neumann boundary conditions can also be applied in order to eliminate \( c_{N-1} \) and \( c_{N+1} \), as:

\[
\begin{align*}
1 \quad 1 \quad 0 \\
1 \quad 1 \quad 1 \\
1 \quad 1 \quad 0 \\
1 \quad 1 \quad 1 \\
\end{align*}
\]

\[
(3, \quad x \quad N \quad N \\
3, \quad x \quad N \quad N \\
3, \quad x \quad N \quad N \\
3, \quad x \quad N \quad N \\
\end{align*}
\]

\[
\begin{align*}
&= \left( 1 - a_0 \frac{3}{h} \right) c^{i+1}_{i-1} + \left( 4 + a_0 \frac{6}{h^2} \right) c^{i+1}_i + \left( 1 - a_0 \frac{3}{h} \right) c^{i+1}_{i+1} = F_i, \quad i = 0, 1, 2, \ldots, N \\
\end{align*}
\]

where,

\[
F_i = (c^0_{i-1} + 4c^0_i + c^0_{i+1}) + \frac{1}{2} a_{i} f^2_i
\]

The above Eq. (17) is a system of \((N+1)\) linear equations in \((N+3)\) unknowns \([c^1_{i-1}, c^1_0, c^1_1, \ldots, c^1_N, c^1_{N+1}]^T\). 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 \( C^j \). This is found by repeatedly solving the recurrence relationship, once the initial vector \( C^0 = [c^0_0, c^0_1, \ldots, c^0_N]^T \), 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 \( C^0 \) 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:

\[
\begin{align*}
U_x (a, 0) &= u_x (x, 0), \\
U_x (x_i, 0) &= u^0 (x_i), \quad i = 1, 2, 3, \ldots, N - 1 \\
U_x (b, 0) &= u_x (x_N, 0)
\end{align*}
\]

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

\[
\begin{align*}
G C^0 &= E
\end{align*}
\]

where,

\[
G = \begin{bmatrix}
0 & 2 & 0 & 0 & \cdots & 0 & 0 \\
1 & 4 & 1 & 0 & \cdots & 0 & 0 \\
1 & 4 & 1 & 0 & \cdots & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
1 & 4 & 1 & 0 & \cdots & 0 & 0 \\
0 & 2 & 4 \\
\end{bmatrix}
\]

**NUMERICAL RESULTS**

The proposed method is tested on the following three problems. Let:
The maximum norm error is given with the following definitions:

\[ \| e \|_{\infty} = \max_{i, j} | u_i(x_j, t_{M}) - u_i(x_j, t_{N}) | \]

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:

\[ \| e \|_{\infty} = \max_{i} \left| u_i(x_j, t_{M}) - u_i(x_j, t_{N}) \right| \]

\( L_2 \) norm error:

\[ \| e \|_{L_2} = \left( \frac{1}{N} \sum_{i=0}^{N} \left| u_i(x_j, t_{M}) - u_i(x_j, t_{N}) \right|^2 \right)^{\frac{1}{2}} \]

### Example 1:

Following is the second order parabolic integro-differential equation:

\[
\int_{0}^{t} (t-s)^{\alpha-1} u_s(x, s)ds - u_{ss}(x, t) = f(x, t), \quad x \in [0,1], \quad t > 0, \quad \alpha = 0.5
\]

with the initial condition:

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

and boundary conditions:

\[ u(0, t) = 0 = u(1, t), \quad t \geq 0 \]

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 \( k \) and 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:

\[
\int_{0}^{t} (t-s)^{\alpha-1} u_s(x, s)ds - u_{ss}(x, t) = f(x, t), \quad x \in [0,1], \quad t > 0, \quad \alpha = 0.5
\]

with the initial condition:

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

and Dirichlet boundary conditions:

\[ u(0, t) = (t + 1), \quad u(1, t) = (t + 1) \cos(\pi), \quad t \geq 0 \]

The exact solution of the problem is:
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:

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

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = f(x,t), \quad x \in [-1, 1], \quad t > 0, \quad \alpha = 0.5$$

(20)
with the initial condition:

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

and Neumann boundary conditions:

\begin{align*}
    u_x(-1,t) &= \pi(t+1)^2 \cos \pi, \\
    u_x(1,t) &= \pi(t+1)^2 \cos \pi, \quad t \geq 0
\end{align*}

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.

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. 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


