

DIRECT SOLUTION OF THIRD-ORDER LINEAR INTEGRO-DIFFERENTIAL EQUATION USING CUBIC B-SPLINE METHOD

(Penyelesaian Terus Persamaan Pembezaan-kamiran Linear Peringkat Tiga Menggunakan Kaedah B-Splin Kubik)

HAMIDA ALI & NORAZAK SENU*

ABSTRACT

Solving Volterra integro-differential equations with cubic B-splines offers a powerful, flexible numerical approach to model complex real-world systems, such as those in population dynamics or material science. The collocation method, based on the cubic B-spline approach, has been developed to solve third-order linear Fredholm and Volterra integro-differential equations. Cubic B-splines are often preferred over other interpolation and approximation techniques due to their smoothness, local control, numerical stability, and computational efficiency. This cubic B-spline method collocates both the solution and its derivatives. Meanwhile, the integral part is approximated using the GaussLegendre quadrature formula. A theoretical convergence analysis of the cubic B-spline method has been conducted and found that the order convergence is second order. The method was applied to six test examples, and the numerical solutions were compared with the corresponding analytical solutions. The maximum absolute error and mean square root error are decreased by increasing the number N . It has been found that the proposed method is efficient and capable of solving third-order linear integro-differential equations for different values of N .

Keywords: Cubic B-spline; Fredholm integro-differential equations; GaussLegendre quadrature; Volterra integro-differential equations.

ABSTRAK

Kaedah kolokasi, berdasarkan pendekatan B-splin kubik, telah dibangunkan untuk menyelesaikan persamaan pembezaan-kamiran Fredholm dan Volterra peringkat ketiga. Kaedah B-splin kubik ini mengkolokasikan kedua-dua penyelesaian dan terbitannya. Sementara itu, bahagian kamiran dianggarkan menggunakan formula kuadratur Gauss-Legendre. Analisis penumpuan secara teori terhadap kaedah B-splin kubik telah dilaksanakan. Kaedah ini telah digunakan kepada enam contoh ujian, dan penyelesaian berangka dibandingkan dengan penyelesaian analitik yang sepadan. Didapati bahawa kaedah yang dicadangkan adalah cekap dan mampu menyelesaikan persamaan pembezaan-kamiran linear peringkat ketiga untuk nilai N yang berbeza.

Kata kunci: B-splin Kubik; persamaan pembezaan-kamiran Fredholm; Kuadratur Gauss Legendre; persamaan pembezaan-kamiran Volterra.

1. Introduction

Third-order integro-differential equations, which include the third-order derivative of the unknown function, are prevalent in various scientific fields. They are used to formulate many problems in mechanics, mathematical biology, physics, and economics, encompassing applications from spectroscopic data analysis to simultaneous dual relationships and heat transfer issues. Generally, obtaining exact solutions for third-order integro-differential equations can be challenging or even impossible. Consequently, numerical methods are often employed to find approximate solutions to these complex equations.

Numerous approaches have been proposed to solve integro-differential equations. These include the Laplace Adomian decomposition method studied by Wazwaz (2010) and Heris (2012) to solve integro differential equation; and the modified Taylor expansion method studied by Rashidinia and Tahmasebi (2013). The fast multiscale Galerkin method has been studied by Chen *et al.* (2020) for solved second-order linear Fredholm integro-differential equations (FIDE), while sinc-collocation methods have addressed second-order linear and non-linear integro-differential equations in Mohsen and El-Gamel (2007) and Yeganeh *et al.* (2012). The Tau method in Hosseini and Shahmorad (2005) and Pour-Mahmoud *et al.* (2005) has been applied to Fredholm integrodifferential equations, the variational iteration method studied by Lanlege *et al.* (2023) for solved linear fredholm integro differential equation. The Variational iteration method in Shang and Han (2010) has been used for n th-order integro-differential equations. The hybrid block method in Janodi *et al.* (2020) was utilized for first-order Volterra integro-differential equations (VIDE), and the Runge-Kutta Fehlberg method studied by Filiz (2014) has solved linear integro-differential equations. The Adomian decomposition method and variational iteration methods in Abdirashidov *et al.* (2020) was applied to higher order integro-differential equation Fredholm. The B-spline method, known for its ability and versatility in handling local phenomena, provides a significant advantage in solving integro-differential equations, especially when combined with collocation.

The cubic B-spline method in Mahmoodi *et al.* (2013) has solved second-order Fredholm and Volterra integro-differential equations. Moreover, quartic B-spline wavelet and operational matrices in Maleknejad and Rostami (2014) have been employed for solving integro-differential equations as detailed. The extended trapezoidal method in Ishak and Ahmad (2016) has been applied to Volterra integro-differential equations, while the Galerkin method with a new quadratic spline wavelet in Cerna and Finek (2020) was explored for approximate solutions of integro-differential equations. The cubic B-spline method in Lanlege *et al.* (2023) and Iqbal *et al.* (2024) was used to solved heat equation. Linear Volterra integro-differential equations in Aggarwal *et al.* (2018) studied have been using the Mahgoub transform method, and the Haar wavelet method in Alqarni *et al.* (2021) has been applied to third-order integro-differential equations. The cubic B-spline finite element method in Sabawi and Hussien (2024) was utilized for second order Volterra integro differential equation. Finite difference method studied by Saber *et al.* (2023) to solve linear Fredholm integro-differential equation. The quadrature formulas by Shodmonova *et al.* (2020) has solved nonlinear integro-differential equation.

Lastly, the stability of integro-differential equations in Jin and Luo (2009) was examined. Generally, integro-differential equations up to the n th order can be solved using the B-spline method of order $n + 1$. However, in this paper, cubic B-spline collocation will be employed to approximate unknown functions and their derivatives for the first, second, and third orders of the analytical solution simultaneously. Additionally, the Gauss-Legendre quadrature will be utilized to approximate the integral part. A third-order linear Fredholm integro-differential equation will be considered with boundary conditions:

$$y'''(x) + \int_a^b k(x, t)y(t)dt = g(x), \quad x \in [a, b] \quad (1)$$

$$y(a) = \alpha, \quad y'(a) = \beta, \quad y(b) = \gamma$$

and third-order linear Volterra integro-differential equation with boundary conditions:

$$y'''(x) + \int_a^x k(x, t)y(t)dt = g(x), \quad a \leq t \leq x \quad (2)$$

$$y(a) = \alpha, y'(a) = \beta, \quad y(b) = \gamma.$$

In the given context, α , γ and β are constants, $y(x)$ is an unknown function, $g(x)$ is a known

function, and k satisfies the Lipschitz condition and is a continuous function on the interval $[a, b]$.

In this study, the unknown function in Eq. (1) and (2) will be approximated using the cubic B-spline collocation method, while the integral part will be approximated using the Gauss-Legendre formula. Section 2 provides an explanation of the cubic B-spline method. A linear Fredholm integro-differential equation is considered in Section 3, and a linear Volterra integro-differential equation is discussed in Section 4. Section 5 describes the error analysis of the proposed method. Numerical results are presented in Section 6. The algorithm for solving third-order linear integro-differential equations using the cubic B-spline method is presented in Section 7. The paper concludes with a discussion and conclusion in Section 8.

2. Cubic B-spline Formulation

The fundamental role of the cubic B-spline function is to generate an interpolating spline S for use in the formulation of the cubic B-spline collocation method. Due to its smoothness and flexibility, the cubic B-spline method is of significant importance in both mathematics and engineering.

Let consider partition $\rho = \{x_0 < x_1 < \dots < x_N\}$ on interval on $[a, b]$, then the cubic B-spline is referred by:

$$S_3(\rho) = s \in C^2[a, b] : s|_{[x_i, x_{i+1}]} \in P_3, \quad i = 0, 1, \dots, N$$

The B-spline defined by recursion De Boor (1978):

$$B_{i,k+1}^n(x) = \frac{x - x_i}{x_{i+k} - x_i} B_{i,k}^{n-1}(x) + \frac{x_{i+k+1} - x}{x_{i+k+1} - x_{i+1}} B_{i+1,k}^{n-1}(x). \quad (3)$$

The B-spline of degree zero is defined by:

$$B_i^0(x) = \begin{cases} 1 & x \in [x_{i-1}, x_i] \\ 0 & \text{otherwise.} \end{cases}$$

Cubic B-spline is defined using the following equation:

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

Let $s(x)$ be the cubic B-spline function interpolating at the knot, then $s(x)$ is possible to write as linear combination:

$$s(x) = \sum_{i=-1}^{n+1} c_i B_i(x) \quad (4)$$

Table 1: Values of $B_i(x)$, $B'_i(x)$, $B''_i(x)$ at knots.

	x_{i-1}	x_i	x_{i+1}	other knot
$B_i(x)$	1	4	1	0
$B'_i(x)$	$3/h$	0	$-3/h$	0
$B''_i(x)$	$1/h^2$	$-12/h^2$	$6/h^2$	0

For a given function $y(x_i)$, there exists a unique cubic spline $s(x_i) = \sum_{i=-1}^{n+1} c_i B_i(x)$ satisfying the interpolating conditions:

$$s(x_i) = y(x_i), \quad i = 0, 1, \dots, N$$

and the end conditions:

$$C_1 : s(x_0) = y(x_0), \quad s(x_N) = y(x_N)$$

$$C_2 : D^j s(x_0) = D^j s(x_N) \quad j = 1, 2$$

$$C_3 : s''(x_0) = 0, \quad s''(x_N) = 0.$$

By applying Eq. (3), the values of the first and second derivatives of $B_i(x)$ are obtained at the knots. These derivatives are tabulated in Table 1. Let denote $m_i = s'(x_i)$ and $M_i = s''(x_i)$. From Table 1, it can be deduced:

$$y(x_i) = s(x_i) = \frac{c_{i-1} + 4c_i + c_{i+1}}{6} \quad (5)$$

$$m_i = s'(x_i) = \frac{c_{i-1} - c_{i+1}}{2h} \quad (6)$$

$$M_i = s''(x_i) = \frac{c_{i-1} - 2c_i + c_{i+1}}{h^2} \quad (7)$$

Moreover in Fyfe (1969), it found that:

$$m_i = s'(x_i) = y'(x_i) - \frac{1}{180} h^4 y^{(5)}(x_i) + \mathcal{O}(h^6) \quad (8)$$

By substituting m_i into Eq. (8) and expanding m_{i-1} at x_i using the Taylor series expansion of $y'(x_i)$, the following is obtained:

$$m_{i-1} = y'(x_{i-1}) - \frac{1}{180} h^4 y^{(5)}(x_{i-1}) + \mathcal{O}(h^6) \quad (9)$$

$$= y'(x_i) - h y''(x_i) + \frac{h^2}{2} y'''(x_i) - \frac{1}{6} h^3 y^{(4)}(x_i) + \mathcal{O}(h^4). \quad (10)$$

Similarly, expanding m_{i+1} at x_i

$$m_{i+1} = y'(x_i) + h y''(x_i) + \frac{h^2}{2} y'''(x_i) + \frac{1}{6} h^3 y^{(4)}(x_i) + \mathcal{O}(h^4). \quad (11)$$

Using Eq. (8), (10) and (11) are obtained

$$\frac{m_{i-1} - 2m_i + m_{i+1}}{h^2} = y'''(x_i) + \mathcal{O}(h^2). \quad (12)$$

Substituting m_{i-1} , m_i , and m_{i+1} into Eq. (12) with their respective values from Eq. (6), An

approximation of the third derivatives is obtained. $y(x_i)$:

$$y'''(x_i) = \frac{-c_{i-2} + 2c_{i-1} - 2c_{i+1} + c_{i+2}}{2h^3}. \quad (13)$$

So, using value of S_i , S'_i and S''_i are obtained on estimate the values of $y(x_i)$, $y'(x_i)$, $y''(x_i)$, and $y'''(x_i)$ as

$$\begin{aligned} y(x_i) &= \frac{c_{i-1} + 4c_i + c_{i+1}}{6} \\ y'(x_i) &= \frac{c_{i-1} - c_{i+1}}{2h} \\ y''(x_i) &= \frac{c_{i-1} - 2c_i + c_{i+1}}{h^2} \\ y'''(x_i) &= \frac{-c_{i-2} + 2c_{i-1} - 2c_{i+1} + c_{i+2}}{2h^3}. \end{aligned} \quad (14)$$

3. Linear Fredholm Integro-Differential Equation

In this section, the third-order linear Fredholm Integro-Differential Equation (FIDE) represented by Eq. (1) is considered. For the solution of this equation, the terms $y(x)$ and its first, second, and third derivatives are substituted by cubic B-spline functions. Eq. (1) is collocated at the collocation knots defined by $x_i = a + ih$, $h = (b - a)/N$, $i = 0, 1, 2, 3, \dots, N$:

$$s'''(x_i) + \int_a^b k(x_i, t)s(t)dt = g(x_i), \quad i = 1, 2, \dots, N.$$

The interval $[a, b]$ is split into N equivalent subintervals, resulting in the following:

$$\begin{aligned} s'''(x_i) + \sum_{j=0}^{N-1} \int_{x_j}^{x_{j+1}} k(x_i, t)s(t)dt &= g(x_i), \quad i = 1, 2, \dots, N \\ \sum_{k=-1}^N c_k B_k''' + \sum_{j=0}^{N-1} \int_{x_j}^{x_{j+1}} k(x_i, t) \sum_{k=-1}^N (c_k B_k)dt &= g(x_i), \quad i = 1, 2, \dots, N. \end{aligned} \quad (15)$$

The values of B_k , B'_k and B''_k are obtained from Table 1, and B_k''' is derived from Eq. (13). Consequently, Eq. (15) is transformed as follows:

$$\begin{aligned} \frac{c_{i+2} - 2c_{i+1} + 2c_{i-1} - c_{i-2}}{2h^3} + \sum_{j=0}^{N-1} \int_{x_j}^{x_{j+1}} k(x_i, t) \frac{c_{i-1} + 4c_i + c_{i+1}}{6} dt &= g(x_i) \\ c_{i+2} - 2c_{i+1} + 2c_{i-1} - c_{i-2} + \frac{h^3}{3} (c_{i-1} + 4c_i + c_{i+1}) \sum_{j=0}^{N-1} \int_{x_j}^{x_{j+1}} k(x_i, t) dt &= 2h^3 g(x_i), \quad i = 1, 2, \dots, N. \end{aligned} \quad (16)$$

Using Table 1 and the stated boundary conditions, the following is obtained:

$$\begin{aligned} y(a) &= \frac{c_1 + 4c_0 + c_{-1}}{6} = \alpha \\ y'(a) &= \frac{c_1 - c_{-1}}{2h} = \beta \\ y(b) &= \frac{c_{N-1} + 4c_N + c_{N+1}}{6} = \gamma. \end{aligned}$$

The Gauss-Legendre formula in Babolian *et al.* (2005); Epperson (2013) is used for the integral part, as described

$$\int_{-1}^1 f(\xi) d\xi \approx \sum_{k=1}^m w_k f(\xi_k), w_k = \frac{1}{(1 - \xi_k)^2 [P'(\xi_k)]^2}$$

and ξ_k are the roots of Legendre polynomial $P_n(\xi)$, m is the number of knots used and w_k are weights. Using the following relation,

$$t_k = \frac{a + b + (b - a)\xi_k}{2},$$

the Gauss-Legendre formula is used on interval $[a, b]$, yield

$$\int_a^b f(t) dt \approx \sum_{k=1}^m w_k f(x_k).$$

So, converting each subinterval $[x_j, x_{j+1}]$ in to the interval $[-1, 1]$ using relation

$$t = \frac{1}{2}[(x_{j+1} + x_j) + (x_{j+1} - x_j)\xi], dt = \frac{h}{2}d\xi.$$

Then, by changing the variable, yields

$$\int_{x_j}^{x_{j+1}} k(x_i, t) dt = \frac{h}{2} \int_{-1}^1 k(x_i, \frac{1}{2}[(x_{j+1} + x_j) + (x_{j+1} - x_j)\xi]) d\xi \quad (17)$$

$$= \frac{h}{2} \sum_{k=1}^m w_k k(x_i, \frac{1}{2}[(x_{j+1} + x_j) + (x_{j+1} - x_j)\xi_k]). \quad (18)$$

Next, by substituting Eq. (18) into Eq. (16), the linear algebra system equations are obtained,

$$\begin{aligned} c_{i+2} - 2c_{i+1} + 2c_{i-1} - c_{i-2} + \frac{h^4}{6}(c_{i-1} + 4c_i + c_{i+1}) \sum_{j=1}^{N-1} \sum_{k=1}^m w_k k(x_i, t_{j,k}) \\ = 2h^3 g(x_i), \quad i = 1, 2, \dots, N, \end{aligned} \quad (19)$$

$$t_{i,k} = \frac{1}{2}[(x_{j+1} + x_j) + (x_{j+1} - x_j)\xi].$$

This system can be represented in matrix form as:

$$(A + D)C = 2h^3 G \quad (20)$$

C , G , A and D are:

$$C = [c_{-1} \quad c_0 \quad c_1 \quad \cdots \quad c_{N-1} \quad c_N \quad c_{N+1} \quad c_{N+2}]^T \quad (21)$$

$$G = [g_{x_1} \quad g_{x_2} \quad \cdots \quad g_{x_{N-1}} \quad g_{x_N}]^T \quad (22)$$

$$A = \begin{bmatrix} -1 & 2 & 0 & -2 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & 0 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & -1 & 2 & 0 & -2 & 1 & 0 \\ 0 & \cdots & 0 & -1 & 2 & 0 & -2 & 1 \end{bmatrix} \quad (23)$$

$$D = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & \cdots & d_{1,N-1} & d_{1,N} \\ d_{2,1} & d_{2,2} & d_{2,3} & \cdots & d_{2,N-1} & d_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{N,1} & d_{N,2} & d_{N,3} & \cdots & d_{N,N-1} & d_{N,N} \end{bmatrix} \quad (24)$$

where

$$\begin{aligned} d_{i,1} &= -I_{i,0} \quad i = 1, 2, \dots, N, \\ d_{i,2} &= -4I_{i,0} - I_{i,1}, \\ d_{i,j} &= -I_{i,j-3} - 4I_{i,j-2} - I_{i,j-1}, \quad i = 1, 2, \dots, N-1, \quad j = 3, 4, \dots, N-2, \\ d_{i,N-1} &= -I_{i,N-2} - 4I_{i,N-1}, \\ d_{i,N} &= I_{i,N-1}, \end{aligned}$$

and

$$I_{i,j} = \int_{x_j}^{x_{j+i}} k(x_i, t) dt, \quad i = 1, 2, \dots, N, \quad j = 0, 1, \dots, N-1.$$

To obtain on the numerical solution Eq. (1), solving linear system in Eq. (20) by using Gauss elimination method, the coefficients c_i can be determined by substituting c_i in Eq. (16).

4. Linear Volterra Integro-Differential Equation

In this section, the third-order linear Volterra integro-differential equation (VIDE) Eq. (2) is considered. To solve this equation, $y(x)$ and its derivatives can be replaced by a cubic B-spline equation. By collocating Eq. (2) at the collocation knots $x_i = a + ih$, where $h = (b - a)/N$ and $i = 0, 1, 2, \dots, N$, The following is obtained:

$$s'''(x_i) + \int_a^x k(x_i, t) s(t) dt = g(x_i), \quad i = 1, 2, \dots, N.$$

By splitting the interval $[a, b]$ into N equivalent subintervals, then obtained

$$s'''(x_i) + \sum_{j=0}^{i-1} \int_{x_j}^{x_{j+1}} k(x_i, t) s(t) dt = g(x_i), \quad i = 1, 2, \dots, N. \quad (25)$$

The value of $s'''(x_i)$, and $s(x)$ are obtained from Eq. (14) and (13). Then Eq. (25) becomes:

$$\begin{aligned} \frac{c_{i+2} - 2c_{i+1} + 2c_{i-1} - c_{i-2}}{2h^3} + \sum_{j=0}^{i-1} \int_{x_j}^{x_{j+1}} k(x_i, t) \frac{c_{i-1} + 4c_i + c_{i+1}}{6} dt &= g(x_i) \\ c_{i+2} - 2c_{i+1} + 2c_{i-1} - c_{i-2} + \frac{h^3}{3} (c_{i-1} + 4c_i + c_{i+1}) \sum_{j=0}^{i-1} \int_{x_j}^{x_{j+1}} k(x_i, t) dt \\ &= 2h^3 g(x_i) \quad i = 1, 2, \dots, N. \end{aligned} \quad (26)$$

Using Table 1 and the stated boundary conditions are obtained:

$$\begin{aligned} y(a) &= \frac{c_1 + 4c_0 + c_{-1}}{6} = \alpha, \\ y'(a) &= \frac{c_1 - c_{-1}}{2h} = \beta, \\ y(b) &= \frac{c_{N+1} + 4c_N + c_{N-1}}{6} = \alpha. \end{aligned}$$

For the integral part, the Gauss-Legendre formula as shown in Eq. (18) are used. By substituting this into Eq. (26), the system of linear algebraic equations are obtained.

$$c_{i+2} - 2c_{i+1} + 2c_{i-1} - c_{i-2} + \frac{h^4}{6} (c_{i-1} + 4c_i + c_{i+1}) \sum_{k=1}^{N-1} \sum_{m=1}^m w_k k(x_i, t_{i,k}) = 2h^3 g(x_i),$$

where $i = 1, 2, \dots, N$. This system can be represented in matrix form as:

$$(A + B)C = 2h^3 G, \quad (27)$$

where C , G and A are defined by matrices in Eq. (21), (22) and (23) respectively, while B given as

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & 0 & 0 & 0 & \cdots & 0 \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & 0 & 0 & \cdots & 0 \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} & 0 & \cdots & 0 \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ b_{N,1} & b_{N,2} & b_{N,3} & b_{N,4} & b_{N,5} & b_{N,6} & \cdots & b_{N,N} \end{bmatrix}, \quad (28)$$

where

$$\begin{aligned} b_{i,1} &= -I_{i,0}, \quad b_{1,2} = -4I_{1,0}, \quad i = 1, 2, \dots, N, \\ b_{1,3} &= -4I_{1,0}, \quad b_{i,2} = -4I_{i,0} - I_{i,1}, \\ b_{i,j} &= -I_{i,j-3} - 4I_{i,j-2} - I_{i,j-1} \quad i = 1, 2, \dots, N-1, \quad j = 3, 4, \dots, N-2, \\ b_{N,N-1} &= -I_{N,N-2} - 4I_{N,N-1}, \quad b_{N,N} = I_{N,N-1}, \end{aligned}$$

and

$$I_{i,j} = \int_{x_j}^{x_{j+i}} k(x_i, t) dt, \quad i = 1, 2, \dots, N, \quad j = 0, 1, \dots, i-1.$$

The linear system of equations referred to by Eq. (27) is solved using the Gauss elimination method, which allows for the determination of the coefficients c_i , where i ranges from -1 to $N + 2$. By inserting these c_i values into Eq. (16), the approximate solution of the third-order Volterra integro-differential Eq. (2) is obtained.

5. Convergence Analysis

In this section, the convergence of the proposed method will be analysed. The following definition as given in Schumaker (2007) will be used.

Definition 5.1. Let $K_j < \infty$ independent of h , if $s(x)$ is the cubic B-spline interpolate $f \in C^4[a, b]$ such that:

$$D^j(f - s) \leq K_j \|D^{4-j} f\|, \quad j = 0, 1, 2, 3$$

where D^j is the derivative, $K_j = 2/j$ and $q = 4 - j$ is the greatest integer for which such an inequality holds, then q is the order of convergence of the cubic B-spline method.

Theorem 5.2. *The approximate method*

$$s'''(x_i) + \frac{h}{2} \sum_{j=1}^{N-1} \sum_{k=1}^m w_k k(x_i, t_{j,k}) s(t) = g(x_i), \quad i = 1, 2, \dots, N$$

$$s(a) = \alpha, \quad s'(a) = \beta, \quad s''(a) = \gamma$$

for approximation solution of a third-order linear Fredholm integro-differential Eq. (1) is error bound is given by:

$$|e_i^3| \leq \frac{h}{2} W L \sum_{j=0}^{N-1} |e_j| + E(h, x_i)$$

where $e_i^3 = y_i''' - s_i'''$, $i = 1, 2, \dots, N$

Proof. Let $x_i = a + ih$ where $h = (b - a)/N$ and $i = 1, 2, \dots, N$, the corresponding approximation method for third-order linear Fredholm integro-differential Eq. (1) is

$$s'''(x_i) + \frac{h}{2} \sum_{j=1}^{N-1} \sum_{k=1}^m w_k k(x_i, t_{j,k}) s(t) = g(x_i), \quad i = 1, 2, \dots, N \quad (29)$$

$$s(a) = \alpha, \quad s'(a) = \beta, \quad s''(a) = \gamma \quad (30)$$

using Gauss-Legendre formula in integral part in Eq. (1) are obtained on

$$y'''(x_i) + \frac{h}{2} \sum_{j=1}^{N-1} \sum_{k=1}^m w_k k(x_i, t_{i,k}) y(t) + E(h, x_i) = g(x_i), \quad i = 1, 2, \dots, N \quad (31)$$

$$y(a) = \alpha, \quad y'(a) = \beta, \quad y''(a) = \gamma$$

where

$$E(h, x_i) = \left| \int_a^b k(x_i, t) y(t) dt - \frac{h}{2} \sum_{j=0}^{N-1} \sum_{k=1}^m w_k k(x_i, t_{j,k}) y(t) \right|.$$

By subtracting Eq. (31) from Eq. (29) and using cubic B-spline interpolation conditions, obtained on

$$|s'''(x_i) - y'''(x_i)| = \frac{h}{2} \sum_{j=0}^{N-1} \sum_{k=1}^m |w_k k(x_i, t_{i,k}) s(t) - w_k k(x_i, t_{i,k}) y(t)| + E(h, x_i)$$

$$\leq \frac{h}{2} W L \sum_{j=0}^{N-1} |s_i - y_i| + E(h, x_i)$$

Suppose $W = \max |w_i|$, $s(t_i) = s_i$, $y(t_i) = y_i$ and kernel integral satisfies Lipschitz condition

$$|k(x_i, t_{i,k}) s(t) - w_k k(x_i, t_{i,k}) y(t)| < L |s_i - y_i|$$

where L is independent of x , t , s , y , and $e_i = s_i - y_i$ then obtained

$$|e_j^3| \leq \frac{h}{2} W L \sum_{i=0}^{N-1} |e_i| + E(h, x_i), \quad i = 0, 1, \dots, N \quad \square$$

From above relation, its clear when $h \rightarrow 0$ then $\lim_{h \rightarrow 0} \max E(h, x_i) = 0$, then both the functions approximate error and the quadrature error are zero and for a fixed i ; $|e_i| \rightarrow 0$ as $h \rightarrow 0$.

Generally,

$$|y(x) - \tilde{s}(x)| \leq |y(x) - s(x)| + |s(x) - \tilde{s}(x)| \leq |\mathcal{O}(h^4) - \mathcal{O}(h^2)| \approx \mathcal{O}(h^2).$$

It is noted that the truncation error arising from approximating the solution $y(x)$ with cubic splines $s(x)$ is $\mathcal{O}(h^2)$. Additionally, the truncation error from the numerical integration of the integral using Gauss-Legendre quadrature with $m = 6$ is $\mathcal{O}(h^2)$. Therefore, it can be concluded that second-order convergence is exhibited by our method overall, as this is determined by the larger of the two errors.

6. Algorithm

The following is the detailed algorithm for developed method (see Algorithm 6.1).

Algorithm 6.1. Cubic B-spline algorithm for solving linear integro-differential equation

Input: a, b, N .

Output: absolute error, maximum error E and mean square root error M

- 1: Compute $h = (b - a)/N$.
- 2: Compute the roots of Legendre polynomial ξ_k and w_k .
- 3: Compute the differential operator matrix A in Eq. (23).
- 4: **if FIDE then**
- 5: Construct the integral operator matrix D in Eq. (24).
- 6: Linear map from $[-1, 1]$ to $[a, b]$.
- 7: Calculate integral part by Gauss-Legendre quadrature formula :

$$I_{i,j} = \frac{h}{2} \sum w_j k(x_j, t_j), \quad i = 1, 2, \dots, N-1, \quad j = 0, 1, \dots, N-1$$
- 8: Compute $A + D$.
- 9: Add boundary condition to matrix $A + D$.
- 10: **end if**
- 11: **if VIDE then**
- 12: Construct the integral operator matrix B in Eq. (28).
- 13: Linear map from $[-1, 1]$ to $[a, b]$.
- 14: Calculate integral part using Gauss-Legendre quadrature formula:

$$I_{i,j} = \frac{h}{2} \sum w_k k(x_i, t_j), \quad i = 1, 2, \dots, N-1, \quad j = 0, 1, \dots, i-1.$$
- 15: Compute $A + B$.
- 16: Add boundary condition to matrix $A + B$.
- 17: **end if**
- 18: Input matrix G in Eq. (22).
- 19: Compute coefficients of cubic B-spline by solving linear system equation using Gauss elimination method.
- 20: Compute approximation solution:

$$y_i = c_{i-1} + 4c_i + c_{i+1}/6; \quad i = 1, 2, 3, \dots, N$$
- 21: Calculate absolute error, maximum error E and mean square root error M

7. Numerical Results

In this section, six examples of linear integro-differential equations (FIDE) and Volterra integro-differential equations (VIDE) are considered with initial and boundary conditions to demonstrate the effectiveness of the cubic B-spline method. These problems have been solved for a variety of values of N , and the absolute errors E as well as the root mean square errors M have been calculated at specific points. If y_{ns} represents the numerical solution and y_{as} represents the analytical solution, then the maximum absolute error E is defined as follows:"

$$E = \max |y_{as} - y_{ns}|.$$

and mean square root error, M is

$$M = \left(\frac{1}{N} \sum_{i=1}^N |y_{ns} - y_{as}|^2 \right)^{\frac{1}{2}}.$$

Example 6.1

Consider the problem of linear FIDE with analytic solution $y(x) = e^x$, Wazwaz (2011)

$$y'''(x) - \int_0^1 xy(t)dt = e^x - x, \quad y(0) = 1, \quad y'(0) = 1, \quad y''(0) = 1, \quad y(1) = e^1$$

This example was solved using the proposed method with $N=10, 20$, and 40 , and absolute error at the various knots are shown in Table 2. Increasing the number of knots then the accuracy

get better. The maximum absolute error and mean square root error at different values of N are presented in Table 3. The maximum absolute error and mean square root error are decreased by increasing the number N . The comparison between numerical solution using cubic B-spline method is analytic solution at $N = 40$ are show in Figure 1.

Table 2: The absolute error for solutions of Example 6.1 at various knots.

x	$N=10$	$N=20$	$N=40$
0	0	0	0
0.1	3.740750(-6)	8.565621(-7)	2.119522(-7)
0.2	1.327605(-5)	3.194115(-6)	7.961420(-7)
0.3	2.711170(-5)	6.614159(-6)	1.653190(-6)
0.4	4.292464(-5)	1.059277(-5)	2.651552(-6)
0.5	5.838003(-5)	1.447547(-5)	3.626804(-6)
0.6	7.011619(-5)	1.747419(-5)	4.380905(-6)
0.7	7.473470(-5)	1.866403(-5)	4.681375(-6)
0.8	6.778280(-5)	1.697964(-5)	4.260397(-6)
0.9	4.474054(-5)	1.121119(-5)	2.813807(-6)
1	0	0	0

Table 3: Different errors analysis for Example 6.1.

N	E	M
5	2.978139(-4)	2.062079(-4)
10	7.473470(-5)	4.824295(-5)
20	1.864603(-5)	1.202492(-5)
40	4.681375(-6)	3.014032(-6)
60	2.083350(-6)	1.341173(-6)
80	1.172677(-6)	7.548897(-7)
150	3.338722(-7)	2.149134(-7)

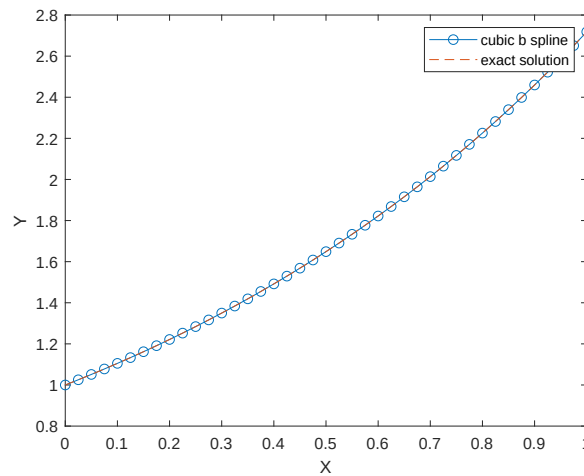


Figure 1: Comparison of numerical solutions with analytic solutions for Example 6.1 with $N = 40$.

Table 4: The absolute errors for solutions of Example 6.2 at various knots.

x	$N = 10$	$N = 20$	$N = 40$
0	0	0	0
0.069314	9.177677(-7)	6.787137(-7)	2.605915(-7)
0.138629	3.807343(-6)	8.177744(-7)	1.271376(-7)
0.207944	1.964583(-7)	1.536240(-6)	1.324004(-6)
0.277258	7.692286(-6)	7.137975(-6)	4.416848(-6)
0.346573	2.353554(-5)	1.5542659(-5)	8.875621(-6)
0.415888	4.031174(-5)	2.510544(-5)	1.382459(-5)
0.485203	5.689322(-5)	3.298075(-5)	5.188295(-5)
0.554517	6.144865(-5)	3.512153(-5)	1.868944(-5)
0.623832	4.803505(-5)	2.627863(-5)	1.385256(-5)

Table 5: Different errors analysis for Example 6.2.

N	E	M
10	6.114865(-5)	3.402006(-5)
40	1.889842(-5)	1.073214(-5)
60	1.285907(-5)	7.333036(-6)
80	9.744334(-6)	5.567190(-6)
150	5.274501(-6)	3.019874(-7)
200	3.972634(-6)	2.276081(-7)

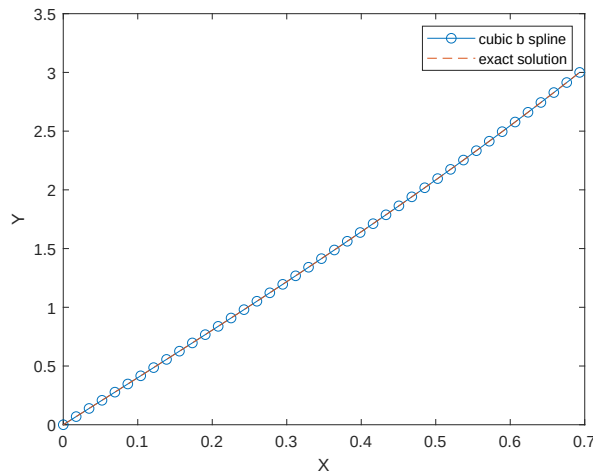


Figure 2: Comparison of numerical solutions with analytic solutions for Example 6.2 with $N = 40$.

Example 6.2

Consider third-order of FIDE with the initial condition and analytic solution $y(x) = 4\sinh(x)$ Wazwaz (2011).

$$y'''(x) - \int_0^{\ln 2} (x-t)y(t)dt = 5\ln(2) - 3 - x + 4\cosh(x),$$

$$y(0) = y''(0) = 0, \quad y'(0) = 4, \quad y(1) = 4\sinh(1)$$

This example was solved by proposed method for deferent value of N the absolute error at various knots displayed in Table 4. As the number of knots increases, the accuracy improves. Table 5 shows the maximum absolute error and mean square root error for different values of N . Both the maximum absolute error and the mean square root error decrease as N increases. Figure 1 presents a comparison between the numerical solution, obtained using the cubic B-spline method, and the analytic solution at $N = 40$.

Example 6.3

Consider third-order linear FIDE with analytic solution $y(x) = \cos(x)$, Wazwaz (2011)

$$y'''(x) - \int_0^{\pi/2} y(t)dt = \sin(x) - 1, \quad y(0) = 1, \quad y'(0) = 0, \quad y(1) = \cos(1)$$

This example was solved at $N = 10, 20$, and 40 . absolute error at the various knots are shown in Table 6. The maximum absolute error and mean square root error at different values of N are presented in Table 7. The comparison between the numerical solution using cubic B-spline method and the analytic solution at $N=40$ is show in Figure 3.

Table 6: The absolute errors for solutions of Example 6.3 at various knots.

x	$N=10$	$N=20$	$N=40$
0	0	0	0
0.157079	2.531225(-4)	2.089308(-4)	1.048427(-4)
0.314159	1.477880(-3)	7.432379(-4)	3.728673(-4)
0.471238	2.750323(-3)	1.463962(-3)	7.342607(-4)
0.628318	4.442203(-3)	2.231850(-3)	1.119135(-3)
0.785398	5.627267(-3)	2.907366(-3)	1.457534(-3)
0.942477	6.675141(-3)	3.350727(-3)	1.679441(-3)
1.099557	6.657671(-3)	3.421927(-3)	1.714777(-3)
1.256637	5.942883(-3)	2.980775(-3)	1.493424(-3)
1.413716	3.601306(-3)	1.886942(-3)	9.452250(-4)
1.570796	0	0	0

Table 7: Different errors analysis for Example 6.3.

N	E	M
5	1.267912(-2)	8.915113(-3)
10	6.675141(-3)	4.4540019(-3)
40	1.724812(-3)	1.137162(-3)
80	8.647144(-4)	5.697210(-4)
150	4.616777(-4)	3.040943(-4)
200	2.770722(-4)	1.825180(-4)

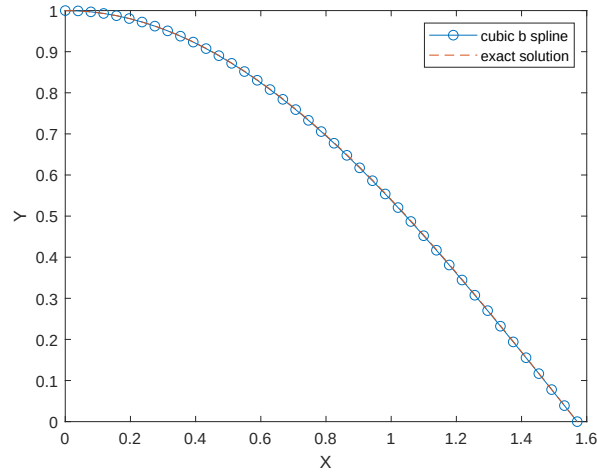


Figure 3: Comparison of numerical solutions with analytic solutions for Example 6.3 with $N = 40$.

Example 6.4

Consider third order of linear VIDE with the initial conditions and analytic solution $y(x) = e^x - x$, Wazwaz (2011).

$$y'''(x) - \int_0^x (x-t)y(t)dt = 1 + x + \frac{x^3}{6},$$

$$y(0) = 1, \quad y'(0) = 0, \quad y''(0) = 1, \quad y(1) = e - 1$$

with values of $N=10, 20, 40$, the approximate solutions are determined, and absolute error at the various knots are given in Table 8. Figure 4 shows a comparison of numerical solutions with analytic solutions an $N = 40$. Different error analyses at different values of N are presented in Table 9. All errors are decreased by increasing the number of N .

Table 8: The absolute errors for solutions of Example 6.4 at various knots.

x	$N=10$	$N=20$	$N=40$
0	0	0	0
0.1	1.263378(-4)	9.865055(-5)	4.892617(-5)
0.2	7.283133(-4)	3.580018(-4)	1.175854(-4)
0.3	1.388211(-3)	7.192028(-4)	3.568248(-4)
0.4	2.272838(-3)	1.118000(-3)	5.547847(-4)
0.5	2.942644(-3)	1.484591(-3)	7.368266(-4)
0.6	3.542140(-3)	1.743544(-3)	8.654926(-4)
0.7	3.609294(-3)	1.813796(-3)	9.005056(-4)
0.8	3.266295(-3)	1.608742(-3)	7.988178(-4)
0.9	2.029361(-3)	1.036449(-3)	5.147167(-4)
1	0	0	0

Table 9: Different errors analysis for Example 6.4.

N	E	M
5	7.176249(-3)	4.972577(-3)
10	3.609294(-3)	2.865055(-3)
40	9.017252(-4)	5.891431(-4)
150	2.396144(-4)	1.564350(-4)
200	1.436674(-4)	9.379753(-5)
250	1.590448(-5)	9.541040(-6)

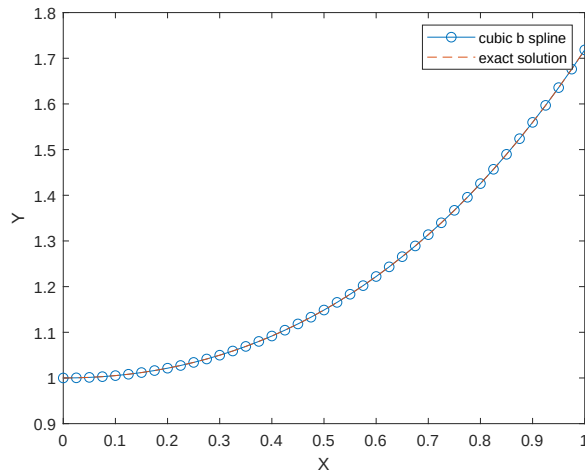
Figure 4: Comparison of numerical solutions with analytic solutions for Example 6.4 with $N = 40$.

Table 10: The errors absolute for solutions of Example 6.5 at various knots.

x	$N=10$	$N=20$	$N=40$
0	0	0	0
0.1	1.135782(-4)	2.166082(-5)	1.321083(-5)
0.2	5.856997(-4)	6.263832(-5)	3.977824(-5)
0.3	4.489879(-4)	9.465740(-5)	6.397583(-5)
0.4	1.356364(-7)	1.000059(-4)	7.536000(-5)
0.5	1.295804(-4)	7.186000(-5)	6.893325(-5)
0.6	2.702300(-4)	1.463997(-5)	4.532289(-5)
0.7	4.299610(-4)	5.559809(-5)	1.097735(-5)
0.8	4.874115(-4)	1.107335(-4)	2.161893(-5)
0.9	4.015889(-4)	1.101078(-4)	3.370874(-5)
1	0	0	0

Example 6.5

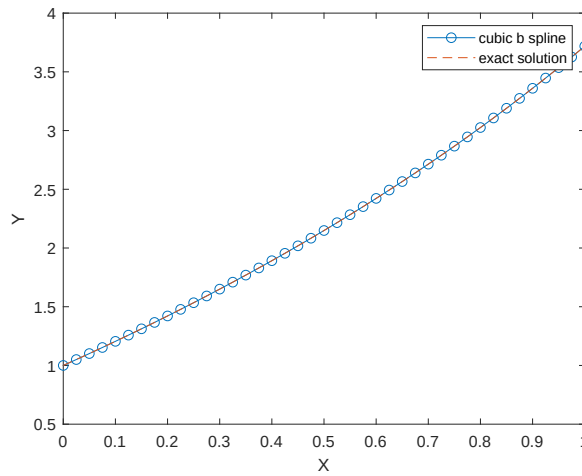
Consider third order of linear VIDE with initial conditions and analytic solution $y(x) = e^x + x$, Alqarni *et al.* (2021).

$$y'''(x) - \int_0^x y(t)dt = 1 - \frac{x^2}{2}, \quad y(0) = 1, \quad y'(0) = 2, \quad y'(1) = e + 1, \quad y(1) = e + 1$$

Table 10 are presented numerical solution of cubic B-spline method at different values of $N=10$,

Table 11: Different errors analysis for Example 6.5.

N	E	M
5	2.007017(-3)	1.229910(-3)
10	4.874117(-4)	2.605956(-4)
40	7.536003(-5)	4.473086(-5)
150	2.565739(-5)	1.533049(-5)
250	1.590448(-5)	9.541040(-6)


 Figure 5: Comparison of numerical results with an analytic solution for Example 6.5 with $N=40$.

20, and 40. Figure 5 are shown between numerical solution and analytic solution at $N = 40$. Mean square root and maximum absolute error at the various values of N are decreased by increasing the value of N are given in Table 11.

Example 6.6

Consider third order of linear VIDE with the initial condition and analytic solution $y(x) = e^x + x^2$, Alqarni *et al.* (2021).

$$y'''(x) - \int_0^x (x-t)y(t)dt = 1+x - \frac{x^4}{12}, \quad y(0) = 1, \quad y'(0) = 1, \quad y''(0) = 3, \quad y(1) = 1 + e^1$$

The proposed method was applied to solve the example, with the absolute error at different knots provided in Table 12. Increasing the number of knots leads to improved accuracy. Table 13 presents the maximum absolute error and the root mean square error for various values of N , both of which decrease as N grows. Figure 1 illustrates a comparison between the numerical solution using the cubic B-spline method and the analytical solution at $N = 40$.

8. Conclusion

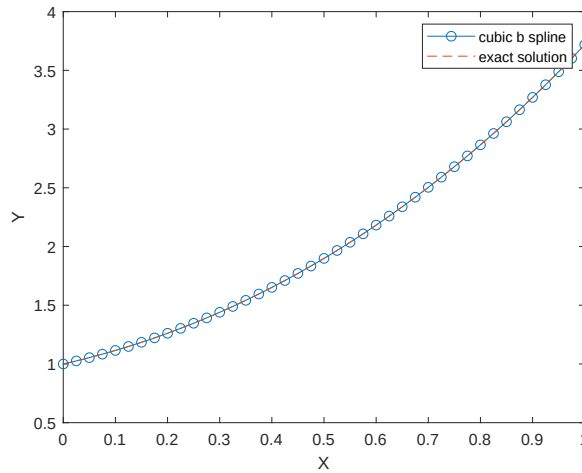
The error analysis for different grid points is presented in tables for all examples. According to these tables, the proposed cubic B-spline method demonstrates high efficiency. Meanwhile, the figures provide a comparison between numerical and analytical solutions at $N = 40$. The approximate solutions of third-order linear Volterra and Fredholm integro-differential equations were obtained using the cubic B-spline method with the Gauss-Legendre quadrature formula.

Table 12: The absolute errors for solutions of Example 6.6 at various knots.

x	$N=10$	$N=20$	$N=40$
0	0	0	0
0.1	9.569188(-6)	6.363924(-6)	3.209899(-6)
0.2	4.849592(-5)	2.522511(-5)	1.274767(-5)
0.3	1.041433(-4)	5.575230(-5)	2.821466(-5)
0.4	1.835304(-4)	9.586040(-5)	4.857010(-5)
0.5	2.676821(-4)	1.413481(-4)	7.169096(-5)
0.6	3.535064(-4)	1.848370(-4)	9.382996(-5)
0.7	4.074062(-4)	2.144789(-4)	1.089566(-4)
0.8	4.061964(-4)	2.124005(-4)	1.079651(-4)
0.9	2.896158(-4)	1.528487(-4)	7.773135(-5)
1	0	0	0

Table 13: Different errors analysis for Example 6.6.

N	E	M
5	7.154708(-4)	4.458028(-4)
10	4.074062(-4)	2.566020(-4)
40	1.111896(-4)	6.846185(-5)
80	5.596143(-5)	3.445216(-5)
100	4.481850(-5)	2.759307(-5)
150	2.992763(-5)	1.842182(-5)

Figure 6: Comparison of numerical solutions with exact solutions for Example 6.6 with $N = 40$.

The integro-differential equations are converted into a system of linear algebraic equations in terms of the coefficients of the linear combination found in the basic spline function representation of the solution. It has been proven that the method can accommodate large numbers of N . The presented method is stable, as evidenced by the decrease in solution error with increasing N . The calculations in this paper were performed using MATLAB.

References

- Abdirashidov A., Usanov F., Kasimova F. & Ismoilov X. 2020. Application of approximate methods for solving higher order Fredholm integro-differential equations. *ISJ Theoretical & Applied Science* **7**(87): 243–249.
- Aggarwal S., Sharma N. & Chauhan R. 2018. Solution of linear volterra integro-differential equations of second kind using mahgoub transform. *International Journal of Latest Technology in Engineering, Management & Applied Science* **7**(5): 173–176.
- Alqarni M.M., Amin R., Shah K., Nazir S., Awais M., Alshehri N.A. & Mahmoud E.E. 2021. Solution of third order linear and nonlinear boundary value problems of integro-differential equations using Haar wavelet method. *Results in Physics* **25**: 104176.
- Babolian E., MasjedJamei M. & Eslahchi M.R. 2005. On numerical improvement of gauss-legendre quadrature rules. *Applied Mathematics and Computation* **160**(3): 779–789.
- Cerna D. & Finek V. 2020. Galerkin method with new quadratic spline wavelets for integral and integro-differential equations. *Journal of Computational and Applied Mathematics* **363**: 426–443.
- Chen J., He M. & Huang Y. 2020. A fast multiscale Galerkin method for solving second order linear Fredholm integro-differential equation with Dirichlet boundary conditions. *Journal of Computational and Applied Mathematics* **364**: 112352.
- De Boor C. 1978. *A Practical Guide to Splines*. New York: Springer-Verlag.
- Epperson J.F. 2013. *An Introduction to Numerical Methods and Analysis*. New Jersey: John Wiley & Sons.
- Filiz A. 2014. Numerical solution of linear Volterra integro-differential equation using Runge-Kutta-Fehlberg method. *Applied and Computational Mathematics* **3**(1): 9–14.
- Fyfe D.J. 1969. The use of cubic splines in the solution of two-point boundary value problems. *The Computer Journal* **12**(2): 188–192.
- Heris J.M. 2012. Solving the integro-differential equations using the modified Laplace Adomian decomposition method. *Journal of Mathematical Extension* **6**(1): 41–55.
- Hosseini S.M. & Shahmorad S. 2005. Numerical piecewise approximate solution of Fredholm integro-differential equations by the Tau method. *Applied Mathematical Modelling* **29**(11): 1005–1021.
- Iqbal M., Zainuddin N., Daud H., Kanan R., Jusoh R., Ullah A. & Khan I.K. 2024. Numerical solution of heat equation using modified cubic b-spline collocation method. *Journal of Advanced Research in Numerical Heat Transfer* **20**(1): 23–35.
- Ishak F. & Ahmad S.N. 2016. Development of extended trapezoidal method for numerical solution of volterra integro-differential equations. *International Journal of Mathematical and Computational Sciences* **10**(11): 579–582.
- Janodi M.R., Majid Z.A., Ismail F. & Senu N. 2020. Numerical solution of volterra integro-differential equations by hybrid block with quadrature rules method. *Malaysian Journal of Mathematical Sciences* **14**(2): 191–208.
- Jin C. & Luo J. 2009. Stability of an integro-differential equation. *Computers & Mathematics with Applications* **57**(7): 1080–1088.
- Lanlege D.I., Edibo F. & Omeiza M.S. 2023. Solution of Fredholm integro-differential equation by variational iteration method. *FUDMA Journal of Sciences* **7**(2): 1–8.
- Mahmoodi Z., Rashidinia J. & Babolian E. 2013. B-spline collocation method for linear and nonlinear Fredholm and volterra integro-differential equations. *Applicable Analysis* **92**(9): 1787–1802.
- Maleknejad K. & Rostami Y. 2014. Numerical solution for integro-differential equations by using quartic b-spline wavelet and operational matrices. *International Journal of Mathematical and Computational Sciences* **8**(7): 1031–1039.
- Mohsen A. & El-Gamel M. 2007. A sinc-collocation method for the linear Fredholm integro-differential equations. *Zeitschrift für angewandte Mathematik und Physik* **58**(3): 380–390.
- Pour-Mahmoud J., Rahimi-Ardabili M.Y. & Shahmorad S. 2005. Numerical solution of the system of Fredholm integro-differential equations by the tau method. *Applied Mathematics and Computation* **168**(1): 465–478.
- Rashidinia J. & Tahmasebi A. 2013. Approximate solution of linear integro-differential equations by using modified Taylor expansion method. *World Journal of Modelling and Simulation* **9**(4): 289–301.
- Sabawi Y.A. & Hussen B.O. 2024. A cubic b-spline finite element method for volterra integro-differential equation. *Palestine Journal of Mathematics* **13**(3): 571–583.
- Saber S.R., Sabawi Y. & Hasso M.S. 2023. Numerical solution of the Fredholm integro-differential equations using

- high-order compact finite difference method. *Journal of Education and Science* **32**(3): 9–22.
- Schumaker L. 2007. *Spline Functions: Basic Theory*. New York: Cambridge University Press.
- Shang X. & Han D. 2010. Application of the variational iteration method for solving nth-order integro-differential equations. *Journal of Computational and Applied Mathematics* **234**(5): 1442–1447.
- Shodmonova G., Islomov U., Abdisamatov O., Khikmatullaev S., Kholiyorov U. & Khamraeva S. 2020. Numerical solution of nonlinear integro-differential equations. *IOP Conference Series: Materials Science and Engineering* **896**(1): 012117.
- Wazwaz A.M. 2010. The combined laplace transform–adomian decomposition method for handling nonlinear volterra integro–differential equations. *Applied Mathematics and Computation* **216**(4): 1304–1309.
- Wazwaz A.M. 2011. *Linear and Non-Linear Integral Equations*. Berlin: Springer.
- Yeganeh S., Ordokhani Y. & Saadatmandi A. 2012. A sinc-collocation method for second-order boundary value problems of nonlinear integro-differential equation. *Journal of Information and Computing Science* **7**(2): 151–160.

*Department of Mathematics and Statistics,
Faculty of Science,
Universiti Putra Malaysia
43400 Serdang,
Selangor, MALAYSIA
E-mail: norazak@upm.edu.my**

*Institute for Mathematical Research,
Universiti Putra Malaysia,
43400 Serdang,
Selangor, MALAYSIA
E-mail: norazak@upm.edu.my**

*Department of Mathematics,
Faculty of Education,
Misurata University,
Misurata, LIBYA
E-mail: h.shafter@edu.misuratau.edu.ly*

Received: 3 October 2024

Accepted: 18 November 2024

*Corresponding author