Punjab University Journal of Mathematics (2025), 57(04), 353-373
https://doi.org/10.52280/pujm.2025.57(04)01

Application of Cubic B-Spline Functions in Galerkin Finite Element Method for Solving Second Order Sub-Diffusion Equation

Ammara Yasin
Department of Mathematics,
Government College University Faisalabad, Pakistan

*Muhammad Kashif Iqbal
Department of Mathematics,
Government College University Faisalabad, Pakistan

Muhammad Abbas
Department of Mathematics,
University of Sargodha, Pakistan

*Corresponding Author
kashifiqbal@gcuf.edu.pk

Received
22 April, 2025

Accepted
16 September, 2025

Published Online
13 November, 2025

Abstract. In this study, the Galerkin finite element method (FEM) is applied to find the numerical solutions of the second-order sub-diffusion equation. The proposed approach employs cubic B-spline functions as both the trial and test functions. The weak formulation of the governing equation is developed, and the connection between the global and local coordinate systems is established through a suitable transformation. For time discretization, the standard finite difference formulation is used for the time derivative, while the Crank-Nicolson scheme is applied to approximate the unknown functions. A stability analysis is performed to verify the robustness of the scheme and to ensure that no numerical errors grow during computations. Additionally, error estimates are derived to assess the accuracy of the proposed method. The efficiency and reliability of the developed scheme are demonstrated by solving several test problems. The computed results are presented in both graphical and tabular forms, showing good agreement with the exact or reference solutions.

AMS (MOS) Subject Classification Codes: 35S29; 40S70; 25U09
Key Words: Diffusion equation, Galerkin method, Cubic B-spline basis functions, Finite element method.

1. INTRODUCTION

In recent decades, the development of advanced mathematical techniques has played a vital role in helping researchers gain a deeper understanding of various physical phenomena through modeling and simulation. The foundation of such simulations is built upon mathematical models-most commonly expressed as differential equations–that describe the underlying physical behavior of systems with high precision. To accurately interpret and apply these phenomena, obtaining reliable solutions to these differential equations is essential [23]. In recent years, there has been a notable increase in research aimed at solving real-world problems using mathematical modeling and numerical simulation [8, 9]. A major milestone in this field was achieved during the 1980s through the pioneering work of Hughes at Stanford University [18, 19], which significantly contributed to the theoretical and numerical understanding of the advection-diffusion equation. This breakthrough stimulated extensive research efforts over the following decades, further enhancing the study and application of differential equations in engineering and applied sciences.

The diffusion equation is a fundamental second-order linear partial differential equation that models the distribution of temperature or concentration within a region over time. The diffusion term in the equation represents the physical process of particles spreading due to random motion, capturing their movement from areas of higher concentration to those of lower concentration. This equation plays a vital role in many scientific and engineering applications, such as water quality prediction, air pollution analysis, meteorological modeling, and oceanographic studies, where the accurate representation of fluid transport or the dispersion of trace substances is essential. However, obtaining analytical solutions for such transport processes is often impossible due to the complexity of real-world systems, which typically involve time-dependent and spatially varying velocity fields. Therefore, numerical approximation methods for solving the governing equations-particularly the diffusion equation-are indispensable tools for analyzing and simulating these phenomena.

The thermal behavior of a one-dimensional rod can be represented by the scalar field $u(\mathfrak{s}, \mathfrak{t})$, where $\mathfrak{s}$ denotes the spatial coordinate and $\mathfrak{t}$ represents time. The temperature distribution along the rod evolves according to the following partial differential equation:

$$\frac{\partial}{\partial \mathfrak{t}}\mathfrak{u}(\mathfrak{s}, \mathfrak{t}) = \alpha \frac{\partial^2}{\partial \mathfrak{s}^2}\mathfrak{u}(\mathfrak{s}, \mathfrak{t}), a \leq \mathfrak{s} \leq b, \mathfrak{t} \geq 0, \tag{1. 1}$$

where $\alpha > 0$ is the thermal diffusivity, which measures how efficiently heat spreads through the rod. Thermal diffusivity depends on the material's thermal conductivity, density, and specific heat, and it controls how fast the substance approaches thermal equilibrium with its surroundings. The initial temperature distribution along the rod and the boundary constraints are prescribed by

$$\mathfrak{u}(\mathfrak{s}, 0) = f(\mathfrak{s}), \mathfrak{s} \in [a, b], \tag{1. 2}$$

$$\mathfrak{u}(a, \mathfrak{t}) = \mathfrak{u}(b, \mathfrak{t}) = 0. \tag{1. 3}$$

The one-dimensional diffusion equation has been widely investigated by many researchers, including Caglar *et al.* [4], Mohebbi [27] and Sun [35]. Elias and Rogerio [10] showed that the heat equation can still be solved analytically by using an exponential sinusoidal one-dimensional model to describe the temperature profile. Monte [29] studied transient heat conduction in a one-dimensional composite slab using a natural analytical method, focusing on how multilayered composite slabs respond to sudden temperature changes in the

surrounding fluid. Lu [26] introduced a new analytical approach to solve the transient heat conduction equation in a one-dimensional hollow composite cylinder with time-dependent boundary conditions. Gorguis and Benny Chan [13] compared the traditional separation of variables technique with the Adomian decomposition method for solving the heat equation.

Cooper [5] provided a modern introduction to the theory of partial differential equations, including an overview of numerical approaches. Ames [1], Morton and Mayers [31], and Cooper [5] presented a more rigorous mathematical treatment of finite difference methods for solving the heat equation. Fletcher [11], Golub and Ortega [12], and Hoffman [17] offered a more practical viewpoint, discussing implementation aspects alongside the theoretical background. Marwaha and Chopra [30] examined transient thermal distribution in a slab with internal heat generation caused by chemical, electrical, or nuclear energy conversion, and provided a numerical solution for the problem. Dehghan has made notable contribution to the numerical analysis of the heat equation, developing numerical methods based on two-level finite difference schemes [7]. Mohebbi and Dehghan [27] proposed a hybrid method that combines a 4th order compact finite difference formulation for space discretization and cubic B-spline collocation for temporal approximation, achieving fourth-order accuracy of $\mathcal{O}(\mathrm{h}^4 + k^4)$ in both space and time. Noye [33] introduced a new three-point implicit finite difference method for the heat equation. Although conditionally stable, this method provides a broader stability range compared to conventional schemes. Cecchi and Pirozzi [6] proposed a family of fully discrete finite difference schemes with two time levels and three spatial points.

B-splines, known for their smooth and piecewise-defined structure, have played an important role in developing efficient numerical algorithms. Their ability to represent higher-order derivatives with accuracy makes them particularly useful in various computational methods. In addition, their flexibility-covering a wide range of degrees and forms, both polynomial and non-polynomial makes them well-suited as basis functions for different numerical schemes. A major development in B-spline construction was introduced by Ya-Juan and Guo-Zhao [41], who derived new basis functions involving hyperbolic trigonometric components, thereby improving geometric flexibility. Xu and Wang [40] developed complementary curve modeling techniques that combine trigonometric and hyperbolic function spaces to meet diverse geometric design needs. Wang and Fang [39] further advanced this field by proposing a unified framework that integrates polynomial, trigonometric, and hyperbolic function spaces, allowing for more general geometric representations.

The influence of various B-spline function shapes has been widely studied in the literature, particularly through methods such as the differential quadrature method and the B-spline collocation method [24, 25]. Recent research, including the work of Gorgulu and Dursun [16], has made important progress in enhancing the computational performance of the Galerkin finite element method by using cubic B-splines. Similarly, Sunarsih *et al.* [36] applied a finite difference discretization technique to numerically solve the theoretical model initially proposed by Mittal and Rohila [32]. They further improved the numerical analysis by adopting a fourth-order collocation method to obtain the solution of the model equation.

While both the cubic B-spline-based Galerkin finite element method (FEM) and the cubic B-spline collocation method are effective tools for solving differential equations, the

Galerkin FEM generally provides stronger theoretical grounding and greater flexibility for complex problems. One key advantage of the Galerkin FEM lies in its u se of the weak formulation of the governing equation, which naturally supports rigorous proofs of stability and convergence. In this approach, the residual is made orthogonal to a set of test functions, unlike in collocation methods, where the equation is enforced only at discrete points. As a result, Galerkin-based schemes often exhibit unconditional stability, meaning that the numerical solution remains stable regardless of the time step size. They also tend to achieve optimal convergence rates in different norms, allowing accuracy to improve rapidly as the mesh is refined. Another notable benefit of the Galerkin FEM is its ability to handle complex geometries and various boundary conditions within a unified variational framework, making it especially suitable for real-world engineering and physical problems. Additionally, the weak formulation used in FEM effectively manages discontinuous coefficients or material property variations, as it can naturally handle jumps across element boundaries. The Galerkin framework also facilitates the derivation of a priori error estimates, which are valuable for assessing the accuracy of computed solutions and guiding adaptive mesh refinement strategies. For certain classes of problems, it can also preserve energy conservation properties. In contrast, while collocation methods are often easier to implement and computationally less demanding for simple geometries, they can struggle with irregular domains and may not provide the same level of theoretical rigor, global error control, or robustness as the Galerkin FEM.

This paper is structured to provide a clear and comprehensive development of the proposed numerical scheme. In Section 2, the computational framework is formulated using cubic B-spline basis functions, and the fully discretized form of the governing equations is derived. Section 3 focuses on the stability analysis, which is carried out using the von Neumann method. In Section 4, the theoretical convergence properties are discussed, and corresponding error bounds are established. Section 5 presents the numerical experiments that verify the accuracy and efficiency of the proposed approach through several test problems. Finally, Section 6 summarizes the main results and highlights their significance for future computational studies.

2. Description of the Numerical Scheme

The interval $[a, b]$, is split into N uniform segments, resulting in a uniform mesh with node points $\mathfrak{s}_k$ (where $a = \mathfrak{s}_0 < \mathfrak{s}_1 < ...\mathfrak{s}_N = b$). Cubic B-splines, $\phi_k, (k = -1, ..., N + 1)$, are defined over this discretized domain [21, 28]:

$$\phi_k = \frac{1}{\mathsf{h}^3}\begin{cases} (\mathfrak{s}-\mathfrak{s}_{k-2})^3 & \mathfrak{s}\in[\mathfrak{s}_{k-2},\mathfrak{s}_{k-1}],\\ \mathsf{h}^3+3\mathsf{h}^2(\mathfrak{s}-\mathfrak{s}_{k-1})+3\mathsf{h}(\mathfrak{s}-\mathfrak{s}_{k-1})^2-3(\mathfrak{s}-\mathfrak{s}_{k-1})^3 & \mathfrak{s}\in[\mathfrak{s}_{k-1},\mathfrak{s}_{k}],\\ \mathsf{h}^3+3\mathsf{h}^2(\mathfrak{s}_{k+1}-\mathfrak{s})+3\mathsf{h}(\mathfrak{s}_{k+1}-\mathfrak{s})^2-3(\mathfrak{s}_{k+1}-\mathfrak{s})^3 & \mathfrak{s}\in[\mathfrak{s}_{k},\mathfrak{s}_{k+1}],\\ (\mathfrak{s}_{k+2}-\mathfrak{s})^3 & \mathfrak{s}\in[\mathfrak{s}_{k+1},\mathfrak{s}_{k+2}],\\ 0 & otherwise. \end{cases}$$

The approximate solution

$$\mathsf{u}_N(\mathfrak{s},\mathfrak{t}) = \sum_{j=-1}^{N+1} \delta_j(\mathfrak{t})\phi_j(\mathfrak{s}). \tag{2. 4}$$

These unknown coefficients $\delta_j(\mathfrak{t})$ are determined through satisfaction of the boundary conditions along with the conditions derived from the weighted residual method. Given the local support of cubic B-splines, which span four elements, each element, $[\mathfrak{s}_k, \mathfrak{s}_{k+1}]$ is influenced by four distinct B-splines. Utilizing the local coordinate transformation $\mathfrak{y} = \mathfrak{s} - \mathfrak{s}_k$, where $0 \leq \mathfrak{y} \leq \mathsf{h}$, the cubic B-spline function can be conveniently represented within the element $[\mathfrak{s}_k, \mathfrak{s}_{k+1}]$ as

$$\begin{aligned}
\phi_{k-1} &= \frac{(\mathsf{h}-\mathfrak{y})^3}{\mathsf{h}^3},\\
\phi_k &= \frac{4\mathsf{h}^3 - 3\mathsf{h}^2\mathfrak{y} + 3\mathsf{h}(\mathsf{h}-\mathfrak{y})^2 - 3(\mathsf{h}-\mathfrak{y})^3}{\mathsf{h}^3}, \qquad 0 \leq \mathfrak{y} \leq \mathsf{h}\\
\phi_{k+1} &= \frac{\mathsf{h}^3 + 3\mathsf{h}^2\mathfrak{y} + 3\mathsf{h}\mathfrak{y}^2 - 3\mathfrak{y}^3}{\mathsf{h}^3},\\
\phi_{k+2} &= \frac{\mathfrak{y}^3}{\mathsf{h}^3}.
\end{aligned} \tag{2. 5}$$

Over each finite element, we can write the approximation function defined in Eq. (2. 4) as a combination of the basis functions specified in Eq. (2. 5) as

$$\mathsf{u}_N(\mathfrak{s},\mathfrak{t}) = \sum_{j=k-1}^{k+2} \delta_j(\mathfrak{t})\phi_j(\mathfrak{s}). \tag{2. 6}$$

Using the B-spline basis functions and trial function from Eq. (2. 4), the nodal values, $\mathsf{u}_k, \mathsf{u}'_k$ and u''_k at knots $\mathfrak{s}_k$ can be written in terms of time-dependent parameters δ_j as:

$$\begin{aligned}
\mathsf{u}_k &= \mathsf{u}(\mathfrak{s}_k) = \delta_{k-1} + 4\delta_k + \delta_{k+1},\\
\mathsf{h}\mathsf{u}'_k &= \mathsf{h}\mathsf{u}'(\mathfrak{s}_k) = 3(\delta_{k+1} - \delta_{k-1}),\\
\mathsf{h}^2\mathsf{u}''_k &= \mathsf{h}^2\mathsf{u}''(\mathfrak{s}_k) = 6(\delta_{k-1} - 2\delta_k + \delta_{k+1}),
\end{aligned} \tag{2. 7}$$

where the prime symbol denotes differentiation with respect to the spatial coordinate $\mathfrak{s}$. Through application of the Galerkin approach to Eq. (1. 1) with shape function $\phi_i(\mathfrak{s})$, we derive the weak formulation of Eq. (1. 1) across the finite element domain $[\mathfrak{s}_k, \mathfrak{s}_{k+1}]$:

$$\int_{\mathfrak{s}_k}^{\mathfrak{s}_{k+1}} \phi_i \mathsf{u}_\mathfrak{t} d\mathfrak{s} = \alpha \int_{\mathfrak{s}_k}^{\mathfrak{s}_{k+1}} \phi_i \mathsf{u}_{\mathfrak{s}\mathfrak{s}} d\mathfrak{s}, \tag{2. 8}$$

By substituting Eq. (2. 6) into Eq. (2. 8) and employing the cubic B-spline function, $\phi_i(\mathfrak{s})$, as the shape function, the resulting differential equation is obtained as

$$\sum_{j=k-1}^{k+2} \left[\left(\int_0^{\mathsf{h}} \phi_i\phi_j d\mathfrak{y} \right) \dot{\delta}_j - \alpha \left(\int_0^{\mathsf{h}} \phi_i\phi_j d\mathfrak{y} \right) \delta_j \right] = 0, \tag{2. 9}$$

$$\sum_{j=k-1}^{k+2} \left[\left(\int_0^{\mathsf{h}} \phi_i\phi_j d\mathfrak{y} \right) \dot{\delta}_j + \alpha \left(\int_0^{\mathsf{h}} \phi'_i\phi'_j d\mathfrak{y} \right) \delta_j \right] = \alpha \sum_{j=k-1}^{k+2} \phi_i\phi_j \Big|_0^{\mathsf{h}} \delta_j, \tag{2. 10}$$

Eq. (2. 10) can be written in matrix form as

$$A^e\dot{\delta}^e + B^e\delta^e = C^e\delta^e, \tag{2. 11}$$

where $\delta^e = (\delta_{k-1}, \delta_k, \delta_{k+1}, \delta_{k+2})^T$ are the local element parameters and $\dot{}$ represent differentiation with respect to $\mathfrak{t}$. The local element matrix A^e_{ij},B^e_{ij} and C^e_{ij} are given by

$$A^e_{ij} = \int_0^{\mathsf{h}} \phi_i \phi_j d\mathfrak{y},\ B^e_{ij} = \int_0^{\mathsf{h}} \phi_i^{'} \phi_j^{'} d\mathfrak{y},\ C^e_{ij} = \phi_i \phi_j \Big|_0^{\mathsf{h}}, \tag{2. 12}$$

with $i, j \in [k-1, k+2]$. The element matrix from Eq. (2. 12) is given by:

$$A^e_{ij} = \frac{\mathsf{h}}{140} \begin{pmatrix} 20 & 129 & 60 & 1 \\ 129 & 1188 & 933 & 60 \\ 60 & 933 & 1188 & 129 \\ 1 & 60 & 129 & 20 \end{pmatrix},$$

$$B^e_{ij} = \frac{1}{10\mathsf{h}} \begin{pmatrix} 18 & 21 & -36 & -3 \\ 21 & 102 & -87 & -36 \\ -36 & -87 & 102 & 21 \\ -3 & -36 & 21 & 18 \end{pmatrix},$$

$$C^e_{ij} = 3 \begin{pmatrix} 1 & 0 & -1 & 0 \\ 4 & -1 & -4 & 1 \\ 1 & -4 & -1 & 4 \\ 0 & -1 & 0 & 1 \end{pmatrix}.$$

Assembling all element contributions yields the global version of Eq. (2. 11)

$$A\dot{\delta} + B\delta - C\delta = 0, \tag{2. 13}$$

where $\delta = (\delta_{-1}, \delta_0, ..., \delta_n, \delta_{n+1})^T$ constitutes the vector of global element parameters, while A, B and C represent global matrices whose k^{th} rows are defined as:

$$\begin{aligned} A &= \frac{\mathsf{h}}{140}(1, 120, 1191, 2416, 1191, 120, 1), \\ B &= \frac{1}{10\mathsf{h}}(-3, -72, -45, 240, -45, -72, -3), \\ C &= (0, 0, 0, 0, 0, 0, 0). \end{aligned} \tag{2. 14}$$

By applying the Crank-Nicolson method for temporal discretization, we obtain the approximation $\delta = \frac{\delta^m + \delta^{m+1}}{2}$. Furthermore, employing a finite difference approximation for the time derivative yields $\dot{\delta} = \frac{\delta^{m+1} - \delta^m}{\Delta \mathfrak{t}}$. Substituting these discretized approximations into Eq. (2. 13), we arrive at the following system.

$$A\delta^{m+1} + 0.5\Delta\mathfrak{t}(B - C)\delta^{m+1} = A\delta^m - 0.5\Delta\mathfrak{t}(B - C)\delta^m. \tag{2. 15}$$

The system of equations (2. 15) comprises $(N + 3)$ linear equations in $(N + 3)$ unknown element parameters, represented by the vector $\delta_{-1}^{m+1}, \ldots, \delta_{N+1}^{m+1}$. By appropriately incorporating the boundary conditions, this system can be reduced to a more compact $(N + 1) \times (N + 1)$ linear system characterized by a septadiagonal coefficient matrix. The

initial parameter vector $\delta^0 = (\delta_0^0, \ldots, \delta_N^0)$ is determined by using the prescribed initial conditions as:

$$\mathsf{u}(\mathfrak{s}, 0) = \sum_{j=-1}^{N+1} \delta_j^0 \phi_j(\mathfrak{s}), \tag{2. 16}$$

where all parameters δ^0 are determined. The approximate solution u_N must satisfy the following relation at each nodal point $\mathfrak{s}_k$:

$$\mathsf{u}_N(\mathfrak{s}_k, 0) = \mathsf{u}(\mathfrak{s}_k, 0),\ k = 0, 1, ..., N. \tag{2. 17}$$

Eq. (2. 16) gives the following matrix equation

$$\begin{bmatrix} 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & 1 & 4 & 1 & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} \delta_{-1}^0 \\ \delta_0^0 \\ \delta_1^0 \\ \delta_2^0 \\ \vdots \\ \delta_{N-2}^0 \\ \delta_{N-1}^0 \\ \delta_N^0 \\ \delta_{N+1}^0 \end{bmatrix} = \begin{bmatrix} (\mathsf{u}_\mathsf{x})_0^0 \\ \mathsf{u}_0^0 \\ \mathsf{u}_1^0 \\ \mathsf{u}_2^0 \\ \vdots \\ \mathsf{u}_{N-2}^0 \\ \mathsf{u}_{N-1}^0 \\ \mathsf{u}_N^0 \\ (\mathsf{u}_\mathsf{x})_N^0 \end{bmatrix}.$$

Two additional equations are derived from the tangent conditions imposed at the boundaries, completing the system, which is then solved numerically.

3. Stability Analysis

The stability of the suggested numerical method is investigated using Von Neumann analysis in this section. Following standard procedure, we consider a representative component from Eq. (2. 15) in the form:

$$\kappa_1\delta_{n-3}^{m+1} + \kappa_2\delta_{n-2}^{m+1} + \kappa_3\delta_{n-1}^{m+1} + \kappa_4\delta_n^{m+1} + \kappa_5\delta_{n+1}^{m+1} + \kappa_6\delta_{n+2}^{m+1} + \kappa_7\delta_{n+3}^{m+1} = \kappa_7\delta_{n-3}^m + \kappa_6\delta_{n-2}^m + \kappa_5\delta_{n-1}^m + \kappa_4\delta_n^m + \kappa_3\delta_{n+1}^m + \kappa_2\delta_{n+2}^m + \kappa_1\delta_{n+3}^m, \tag{3. 18}$$

where
$\kappa_1 = \frac{\mathsf{h}}{140} - \frac{\alpha\Delta\mathsf{t}}{2}\frac{3}{10\mathsf{h}}$, $\kappa_2 = \frac{120\mathsf{h}}{140} - \frac{\alpha\Delta\mathsf{t}}{2}\frac{72}{10\mathsf{h}}$, $\kappa_3 = \frac{1191\mathsf{h}}{140} - \frac{\alpha\Delta\mathsf{t}}{2}\frac{45}{10\mathsf{h}}$,
$\kappa_4 = \frac{2416\mathsf{h}}{140} + \frac{\alpha\Delta\mathsf{t}}{2}\frac{240}{10\mathsf{h}}$, $\kappa_5 = \frac{1191\mathsf{h}}{140} - \frac{\alpha\Delta\mathsf{t}}{2}\frac{45}{10\mathsf{h}}$, $\kappa_6 = \frac{120\mathsf{h}}{140} - \frac{\alpha\Delta\mathsf{t}}{2}\frac{72}{10\mathsf{h}}$, $\kappa_7 = \frac{120\mathsf{h}}{140} - \frac{\alpha\Delta\mathsf{t}}{2}\frac{3}{10\mathsf{h}}$.
Now, substituting the Fourier mode $\delta_n^m = s^m e^{in\theta}$ into (3. 18) we obtain,

$$s^{m+1}((\kappa_1+\kappa_7)\cos 3\theta + (\kappa_2+\kappa_6)\cos 2\theta + (\kappa_3+\kappa_5)\cos\theta + \kappa_4) - \iota((\kappa_1-\kappa_7)\sin 3\theta + (\kappa_2-\kappa_6)\sin 2\theta + (\kappa_3-\kappa_5)\sin\theta) = s^m((\kappa_1+\kappa_7)\cos 3\theta + (\kappa_2+\kappa_6)\cos 2\theta + (\kappa_3+\kappa_5)\cos\theta + \kappa_4) - \iota((\kappa_1-\kappa_7)\sin 3\theta + (\kappa_2-\kappa_6)\sin 2\theta + (\kappa_3-\kappa_5)\sin\theta),$$

which gives:

$$s^{m+1} = Ls^m. \tag{3. 19}$$

Thus, adopting the values $\kappa_i, i = 1, 2, ..., 7$, gives rise to the following relation,

$$L = \frac{c - \iota d}{c + \iota d}, \tag{3. 20}$$

where

$$c = (\kappa_1 + \kappa_7)\cos 3\theta + (\kappa_2 + \kappa_6)\cos 2\theta + (\kappa_3 + \kappa_5)\cos\theta + \kappa_4,$$
$$d = (\kappa_1 + \kappa_7)\sin 3\theta + (\kappa_2 + \kappa_6)\sin 2\theta + (\kappa_3 + \kappa_5)\sin\theta.$$

Taking the modulus of Eq. (3. 20) gives $|L| \leq 1$, therefore, the applied numerical method is unconditionally stable.

4. Error Estimation

Definition 1. *Let W be a Banach space and $b : W \times W \to \mathbb{R}$ a bilinear form. Boundedness requires:*

$$|b(\mathsf{u}, w)| \leq Q||\mathsf{u}||_W||w||_W \ \ \forall\, \mathsf{u}, w \in W,\ Q > 0, \tag{4. 21}$$

There exists a constant Q, independent of u and w. The bilinear form is coercive (or W-elliptic) if

$$b(\mathsf{u}, \mathsf{u}) \geq q||\mathsf{u}||_W^2 \ \ \forall \mathsf{u} \in W,\ q > 0, \tag{4. 22}$$

where the constant q is independent of u.

Theorem 1. *Consider a Hilbert space W equipped with a bilinear form $b : W \times W \to \mathbb{R}$ that is both bounded and coercive. Given a bounded linear functional $f \in W$, let u denote the exact solution and u_N the approximate solution to problem (1. 1). Then, the error between these solutions satisfies the following bound:*

$$||\mathsf{u} - \mathsf{u}_N||_W \leq \frac{Q}{q} \inf_{w_k \in W_k} ||\mathsf{u} - \mathsf{u}_N||_W, \tag{4. 23}$$

where Q and q are constants.

5. Numerical Results

This section presents the numerical solution to Eq. (1. 1) using the cubic B-spline Galerkin finite element method. The accuracy of the proposed method is assessed using the L_2 and L_∞ error norms, defined as:

$$L_2 = ||\mathsf{u}(\mathfrak{s}, \mathfrak{t}) - \mathsf{u}_N||_2 \approx \sqrt{\mathsf{h}\sum_{j=0}^{N} |\mathsf{u}(\mathfrak{s}, \mathfrak{t})_j - (\mathsf{u}_N)_j|^2},$$

$$L_\infty = ||\mathsf{u}(\mathfrak{s}, \mathfrak{t}) - \mathsf{u}_N||_\infty \approx \max|\mathsf{u}(\mathfrak{s}, \mathfrak{t})_j - (\mathsf{u}_N)_j|$$

The numerical order of convergence can be calculated from the following formula.

$$\text{EOC} = \frac{log(L_\infty(N_k)) - log(L_\infty(N_{k+1}))}{log(2)},$$

where $L_\infty(N_k)$ and $L_\infty(N_{k+1})$ represents the error norms at nodal values N_k and N_{k+1} respectively.

Example 5.1. [14, 35]

$$\frac{\partial}{\partial \mathfrak{t}}\mathfrak{u}(\mathfrak{s},\mathfrak{t}) = \frac{\partial^2}{\partial \mathfrak{s}^2}\mathfrak{u}(\mathfrak{s},\mathfrak{t}),\ 0 < \mathfrak{s} < 1,\ \mathfrak{t} \geq 0,$$
$$\mathfrak{u}(\mathfrak{s},0) = \sin(\pi\mathfrak{s}),$$
$$\mathfrak{u}(0,\mathfrak{t}) = \mathfrak{u}(1,\mathfrak{t}) = 0.$$

The exact solution is given by $\mathfrak{u}(\mathfrak{s},\mathfrak{t}) = e^{-\pi^2\mathfrak{t}}\sin(\pi\mathfrak{s})$. To evaluate the performance of the proposed method, a series of numerical experiments have been carried out using different values of the spatial step size ($\mathfrak{h}$) and the temporal step size ($\Delta\mathfrak{t}$). These tests aim to assess the accuracy and efficiency of the method in solving the one-dimensional diffusion equation. The final simulation time is set to $T = 1$. The accuracy of the proposed approach is examined by comparing the maximum absolute errors with those obtained using two well-known numerical schemes: the Crank-Nicolson (CN) method and the Cubic B-spline Collocation (CBSC) method, as reported in [14, 35]. The computed errors for all methods are summarized in Table 1. The comparison shows that the proposed scheme provides accuracy comparable to both CN and CBSC methods, confirming its reliability as an alternative approach for solving the one-dimensional diffusion equation. Table 2 presents the maximum absolute norm values at selected knot points for $\mathfrak{h} = 0.01$ and $\Delta\mathfrak{t} = 0.0125$ at $T = 1$. As shown in Table 3, reducing the time step size results in improved agreement between the numerical and analytical solutions. This improvement is reflected by a noticeable decrease in both the discrete error norm and the maximum nodal error norm. Figure 1 illustrates the comparison between the numerical and exact solutions for $\mathfrak{h} = \Delta\mathfrak{t} = \frac{1}{80}$ at different time levels, while Figures 2 and 3 show the corresponding time-space plots. The results clearly demonstrate the stability, accuracy, and effectiveness of the proposed numerical method.

TABLE 1. Maximum absolute error for Example 5.1

$\mathfrak{h} = \Delta\mathfrak{t}$	CN[35]	CBSC[14]	Presented method	EOC
0.1	3.0×10^{-2}	3.7195×10^{-2}	3.4641×10^{-5}	
0.05	6.9×10^{-3}	8.4588×10^{-3}	1.30278×10^{-5}	1.41089
0.025	1.7×10^{-3}	2.0698×10^{-3}	4.53006×10^{-6}	1.52399
0.0125	4.2×10^{-4}	5.1473×10^{-4}	1.68443×10^{-6}	1.42727

TABLE 2. Numerical errors at the grid points for various mesh sizes for Example 5.1 with $\Delta \mathfrak{t} = 0.0125$ at $T = 1$.

$\mathfrak{s}$	Exact Solution	Numerical Solution	Absolute Error
0.1	0.0000159833	0.0000154628	5.20519×10^{-7}
0.2	0.0000304021	0.000029412	9.90086×10^{-7}
0.3	0.0000418449	0.0000404822	1.36274×10^{-6}
0.4	0.0000491917	0.0000475897	1.60199×10^{-6}
0.5	0.0000517232	0.0000500388	1.68443×10^{-6}
0.6	0.0000491917	0.0000475897	1.60199×10^{-6}
0.7	0.0000418449	0.0000404822	1.36274×10^{-6}
0.8	0.0000304021	0.000029412	9.90086×10^{-7}
0.9	0.0000159833	0.0000154628	5.20519×10^{-7}

TABLE 3. The error norms for Example 5.1 at $T = 1$.

$\mathfrak{s}$	$\Delta \mathfrak{t} = 0.1$	$\Delta \mathfrak{t} = 0.05$	$\Delta \mathfrak{t} = 0.025$	$\Delta \mathfrak{t} = 0.0125$	Exact Solution
0.1	6.22231×10^{-6}	0.0000129826	0.0000151957	0.0000157841	0.0000159833
0.2	0.0000118355	0.0000246943	0.0000289039	0.0000300231	0.0000304021
0.3	0.0000162902	0.0000339888	0.0000397829	0.0000413232	0.0000418449
0.4	0.0000191503	0.0000399562	0.0000467676	0.0000485783	0.0000491917
0.5	0.0000201358	0.0000420125	0.0000491743	0.0000510783	0.0000517232
0.6	0.0000191503	0.0000399562	0.0000467676	0.0000485783	0.0000491917
0.7	0.0000162902	0.0000339888	0.0000397829	0.0000413232	0.0000418449
0.8	0.0000118355	0.0000246943	0.0000289039	0.0000300231	0.0000304021
0.9	6.22231×10^{-6}	0.0000129826	0.0000151957	0.0000157841	0.0000159833
L_2	2.23356×10^{-5}	6.86651×10^{-6}	1.80232×10^{-6}	4.56016×10^{-7}	
L_∞	3.15874×10^{-5}	9.71072×10^{-6}	2.54887×10^{-6}	6.44903×10^{-7}	

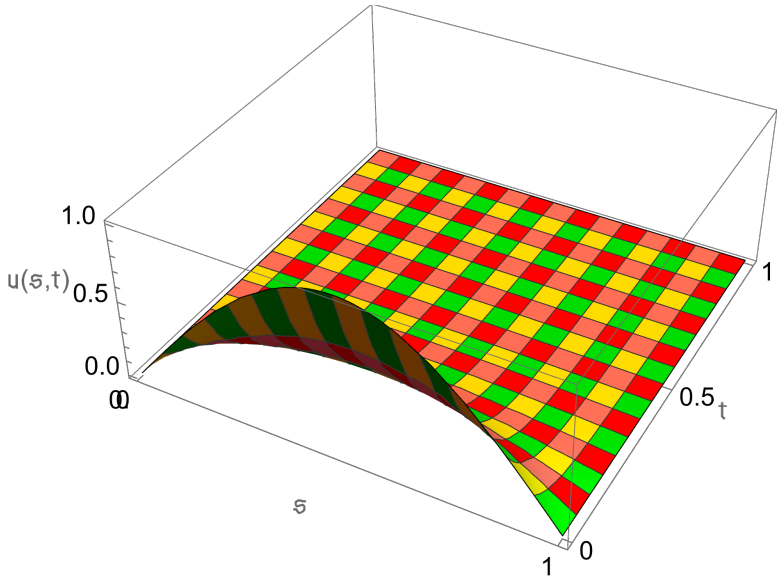

FIGURE 1. Three-dimensional plot of the exact solution for Example 5.1 when $h = \Delta t = \frac{1}{80}$.

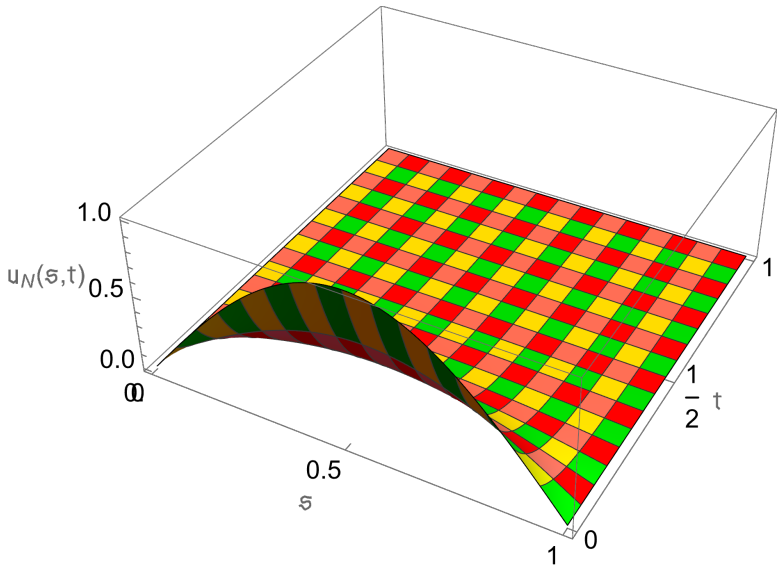

FIGURE 2. Three-dimensional plot of the approximate solution for Example 5.1 with $h = \Delta t = \frac{1}{80}$.

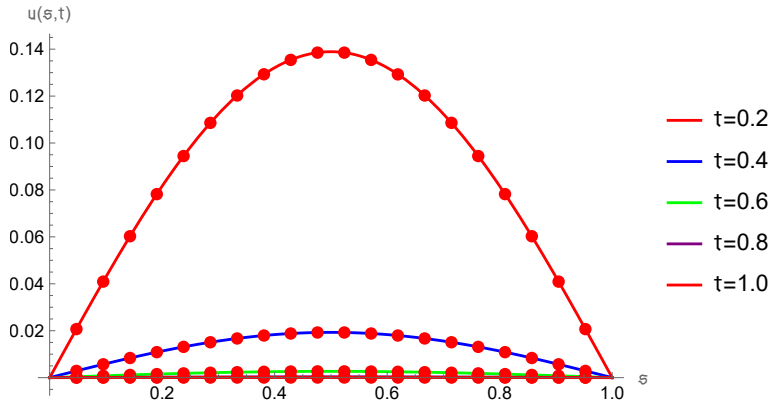

FIGURE 3. Comparative graphical representation of approximated and exact curves for Example 5.1 where $\mathsf{h} = \Delta\mathfrak{t} = \frac{1}{80}$

Example 5.2. [14, 37]

$$\frac{\partial}{\partial \mathfrak{t}}\mathsf{u}(\mathfrak{s},\mathfrak{t}) = \frac{\partial^2}{\partial \mathfrak{s}^2}\mathsf{u}(\mathfrak{s},\mathfrak{t}),\ -1 < \mathfrak{s} < 1,\ \mathfrak{t} \geq 0,$$
$$\mathsf{u}(\mathfrak{s},0) = \sin(\pi\mathfrak{s}),$$
$$\mathsf{u}(-1,\mathfrak{t}) = \mathsf{u}(1,\mathfrak{t}) = 0.$$

To assess the reliability of the numerical results, the exact solution identical to that used in Example 5.1 is employed for comparison. Table 4 presents the maximum errors obtained using the proposed scheme alongside those from the Crank-Nicolson method [37] for different values of $\mathsf{h} = \Delta\mathfrak{t}$. The results clearly indicate that the proposed approach achieves higher accuracy than the Crank-Nicolson scheme. Table 5 lists the maximum absolute norm values at various spatial knot points for the case where the spatial step size is $\mathsf{h} = 0.02$ and the time step is $\Delta\mathfrak{t} = 0.01$. Furthermore, Table 6 summarizes the computed results for $\mathsf{h} = 0.01$, $T = 1$, and several values of $\Delta\mathfrak{t}$. As expected, decreasing the time step size $\Delta\mathfrak{t}$ results in a reduction of both the L_2 and L_∞ error norms, confirming the convergence behavior of the proposed method. The three-dimensional plots shown in Figures 4 and 5, generated for $\mathsf{h} = \Delta\mathfrak{t} = 0.01$, provide clear visual evidence supporting the accuracy and consistency of the numerical scheme. Additionally, Figure 6 illustrates the comparison between the computed and exact solutions for $\mathsf{h} = \Delta\mathfrak{t} = 0.01$ at various time levels, further validating the effectiveness of the proposed method.

TABLE 4. The Comparison of maximum absolute error for Example 5.2.

$h = \Delta t$	CN[37]	CBS[14]	Presented method
0.1	1.809×10^{-2}	7.92279×10^{-4}	3.15683×10^{-5}
0.06	7.792×10^{-3}	4.12053×10^{-4}	1.63029×10^{-5}
0.05	4.513×10^{-3}	2.58991×10^{-4}	9.71073×10^{-6}
0.04	2.820×10^{-3}	1.82947×10^{-4}	6.35029×10^{-6}

TABLE 5. Numerical errors at the grid points for various mesh sizes for Example 5.2 with $\Delta t = 0.01$ at $T = 1$.

s	Exact Solution	Numerical Solution	Absolute Error
-0.9	-0.0000159833	-0.0000158556	1.27726×10^{-7}
-0.7	-0.0000418449	-0.0000415105	3.34391×10^{-7}
-0.5	-0.0000517232	-0.0000513099	4.1333×10^{-7}
-0.3	-0.0000418449	-0.0000415105	3.34391×10^{-7}
-0.1	-0.0000159833	-0.0000158556	1.27726×10^{-7}
0.1	0.0000159833	0.0000158556	1.27726×10^{-7}
0.3	0.0000418449	0.0000415105	3.34391×10^{-7}
0.5	0.0000517232	0.0000513099	4.1333×10^{-7}
0.7	0.0000418449	0.0000415105	3.34391×10^{-7}
0.9	0.0000159833	0.0000158556	1.27726×10^{-7}

TABLE 6. The error norms for Example 5.2 when $T = 1, N = 200$.

$\mathfrak{s}$	$\Delta t = 0.05$	$\Delta t = 0.025$	$\Delta t = 0.0125$	$\Delta t = 0.0625$	Exact Solution
-0.9	-0.0000129826	-0.0000151957	-0.0000157841	-0.0000159334	-0.0000159833
-0.7	-0.0000339888	-0.0000397829	-0.0000413232	-0.0000417141	-0.0000418449
-0.5	-0.0000420125	-0.0000491743	-0.0000510783	-0.0000515615	-0.0000517232
-0.3	-0.0000339888	-0.0000397829	-0.0000413232	-0.0000417141	-0.0000418449
-0.1	-0.0000129826	-0.0000151957	-0.0000157841	-0.0000159334	-0.0000159833
0.1	0.0000129826	0.0000151957	0.0000157841	0.0000159334	0.0000159833
0.3	0.0000339888	0.0000397829	0.0000413232	0.0000417141	0.0000418449
0.5	0.0000420125	0.0000491743	0.0000510783	0.0000515615	0.0000517232
0.7	0.0000339888	0.0000397829	0.0000413232	0.0000417141	0.0000418449
0.9	0.0000129826	0.0000151957	0.0000157841	0.0000159334	0.0000159833
L_2	9.71072×10^{-6}	2.54887×10^{-6}	6.44903×10^{-7}	1.61708×10^{-7}	
L_∞	1.71072×10^{-6}	3.44187×10^{-6}	7.54103×10^{-7}	2.62808×10^{-7}	

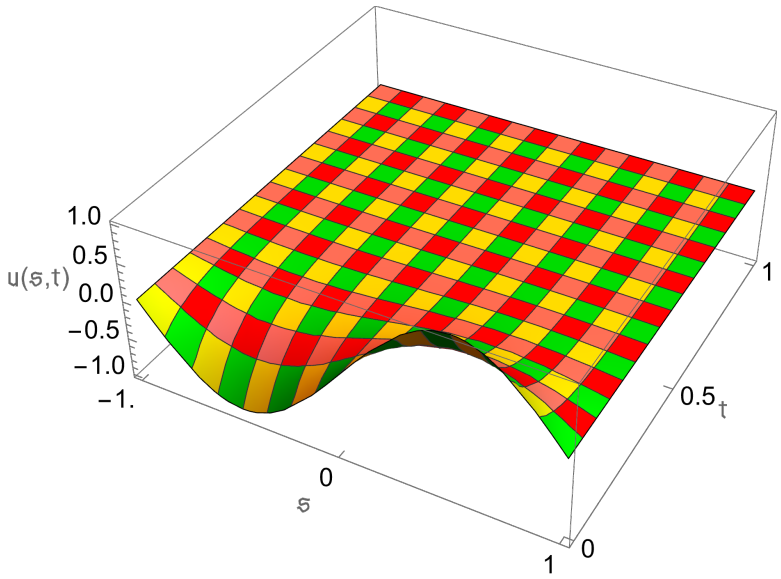

FIGURE 4. 3D representation of exact solution for Example 5.2 when $h = \Delta t = 0.01$.

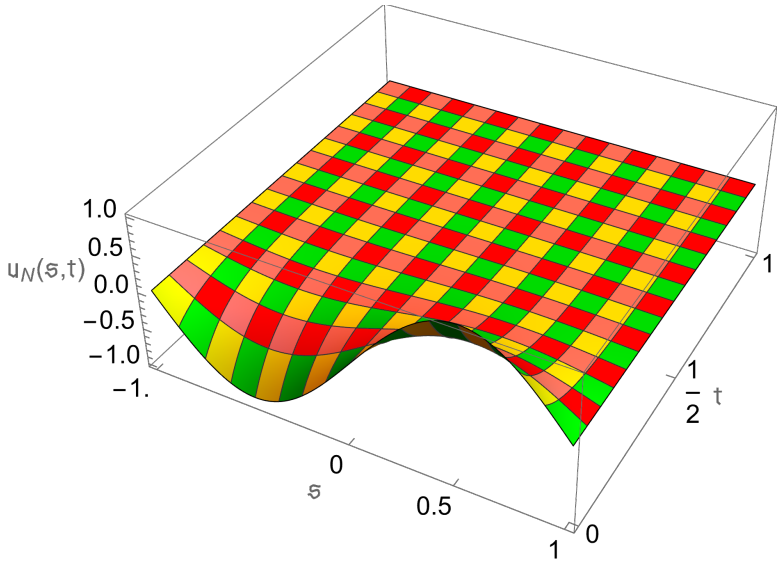

FIGURE 5. Time-space representation of the numerical solution obtained for Example 5.2 with $h = \Delta t = 0.01$.

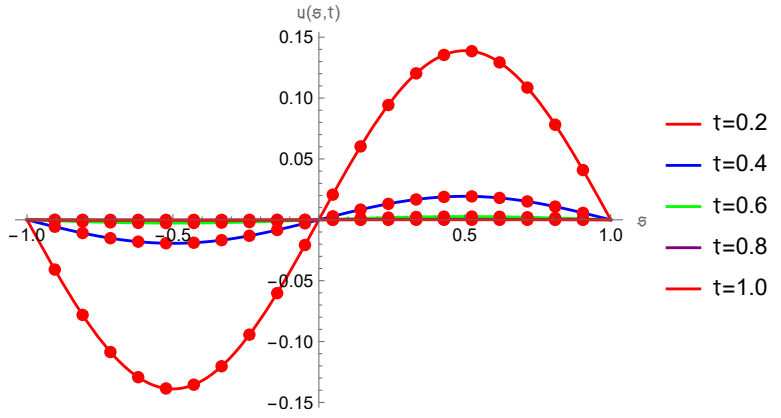

FIGURE 6. Comparison of exact and numerical solutions for Example 5.2 at various time levels, with $h = \Delta t = 0.01$

Example 5.3. [35]

$$\frac{\partial}{\partial \mathfrak{t}}\mathsf{u}(\mathfrak{s},\mathfrak{t}) = \frac{1}{\pi^2}\frac{\partial^2}{\partial \mathfrak{s}^2}\mathsf{u}(\mathfrak{s},\mathfrak{t}),\ 0 < \mathfrak{s} < 1,\ \mathfrak{t} \geq 0,$$
$$\mathsf{u}(\mathfrak{s},0) = \sin(\pi\mathfrak{s}),$$
$$\mathsf{u}(0,\mathfrak{t}) = 0 = \mathsf{u}(1,\mathfrak{t}).$$

The analytical solution is $\mathsf{u}(\mathfrak{s},\mathfrak{t}) = e^{-\mathfrak{t}}\sin(\pi\mathfrak{s})$. Table 7 demonstrates the superior error control capability of the proposed numerical scheme. It provides a comparison of the errors produced by the proposed method and those obtained using the Crank-Nicolson (CN) method [35], with equal spatial (h) and temporal $(\Delta\mathfrak{t})$ step sizes. The results presented in Table 8 for various nodal points reveal an increasingly close agreement between the computed and exact solutions as the mesh is refined. Table 9 summarizes the results for $N = 120$, $T = 1$, and a range of time step sizes $(\Delta\mathfrak{t})$. The observed data confirm a strong convergence trend, with both L_2 and L_∞ error norms decreasing significantly as $\Delta\mathfrak{t}$ is reduced. These findings indicate that the proposed scheme yields highly accurate numerical approximations consistent with the analytical solution. Figure 9 presents solution profiles at different time levels for $\mathsf{h} = 0.02$ and $\Delta\mathfrak{t} = 0.01$ within the spatial domain $-2 \leq \mathfrak{s} \leq 2$. A corresponding three-dimensional space-time surface plot also illustrates the close agreement between the exact and numerical results for $\mathsf{h} = 0.02$, $\Delta\mathfrak{t} = 0.01$, and $T = 1$.

TABLE 7. The Comparison of maximum absolute error for Example 5.3.

$\mathsf{h} = \Delta\mathfrak{t}$	CN[35]	Presented method
0.2	1.1×10^{-2}	1.16824×10^{-3}
0.1	2.7×10^{-3}	3.06898×10^{-4}
0.05	6.8×10^{-4}	7.66617×10^{-5}
0.025	1.7×10^{-4}	1.91617×10^{-5}
0.0125	4.2×10^{-5}	4.79018×10^{-6}
0.00625	1.1×10^{-5}	1.19753×10^{-6}

TABLE 8. Numerical errors at the grid points with various mesh sizes for Example 5.3 when $\Delta \mathfrak{t} = 0.01, \mathsf{h} = 0.005$ at $T = 1$.

$\mathfrak{s}$	Exact Solution	Numerical Solution	Absolute Error
0.1	0.113681	0.11368	9.47352×10^{-7}
0.2	0.216234	0.216232	1.80197×10^{-6}
0.3	0.297621	0.297618	2.4802×10^{-6}
0.4	0.349874	0.349871	2.91565×10^{-6}
0.5	0.367879	0.367876	3.0657×10^{-6}
0.6	0.349874	0.349871	2.91565×10^{-6}
0.7	0.297621	0.297618	2.4802×10^{-6}
0.8	0.216234	0.216232	1.80197×10^{-6}
0.9	0.113681	0.11368	9.47352×10^{-7}

TABLE 9. The error norms for Example 5.3 when $T = 1, N = 100$.

$\mathfrak{s}$	$\Delta \mathfrak{t} = 0.1$	$\Delta \mathfrak{t} = 0.05$	$\Delta \mathfrak{t} = 0.025$	$\Delta \mathfrak{t} = 0.0125$	Exact Solution
0.1	0.113586	0.113657	0.113675	0.11368	0.113681
0.2	0.216054	0.216189	0.216223	0.216231	0.216234
0.3	0.297372	0.297559	0.297605	0.297617	0.297621
0.4	0.349582	0.349801	0.349856	0.34987	0.349874
0.5	0.367573	0.367803	0.36786	0.367875	0.367879
0.6	0.349582	0.349801	0.349856	0.34987	0.349874
0.7	0.297372	0.297559	0.297605	0.297617	0.297621
0.8	0.216054	0.216189	0.216223	0.216231	0.216234
0.9	0.113586	0.113657	0.113675	0.11368	0.113681
L_2	2.1701×10^{-4}	5.42084×10^{-5}	1.35494×10^{-5}	3.38717×10^{-6}	
L_∞	3.06899×10^{-4}	7.66623×10^{-5}	1.91617×10^{-5}	4.79018×10^{-6}	

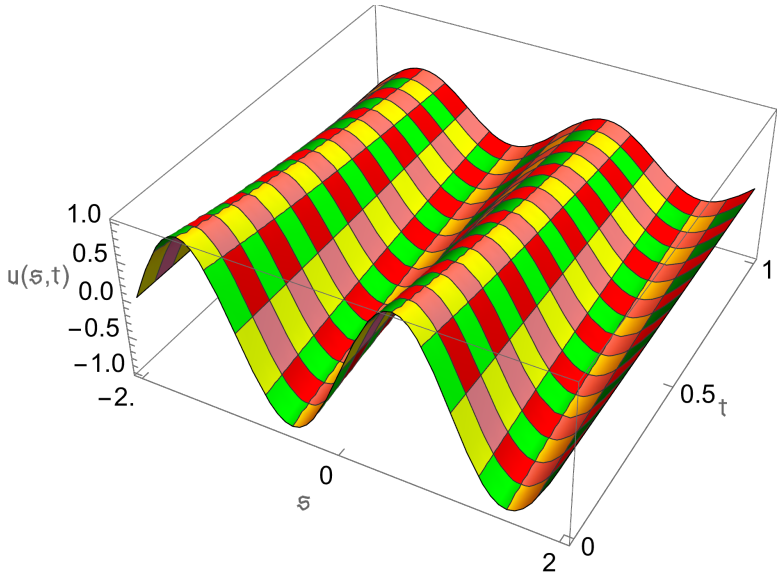

FIGURE 7. Three-dimensional plot of the exact solution for Example 5.3 with $h = 0.02$, $\Delta t = 0.01$, and spatial domain $-2 \leq ś \leq 2$

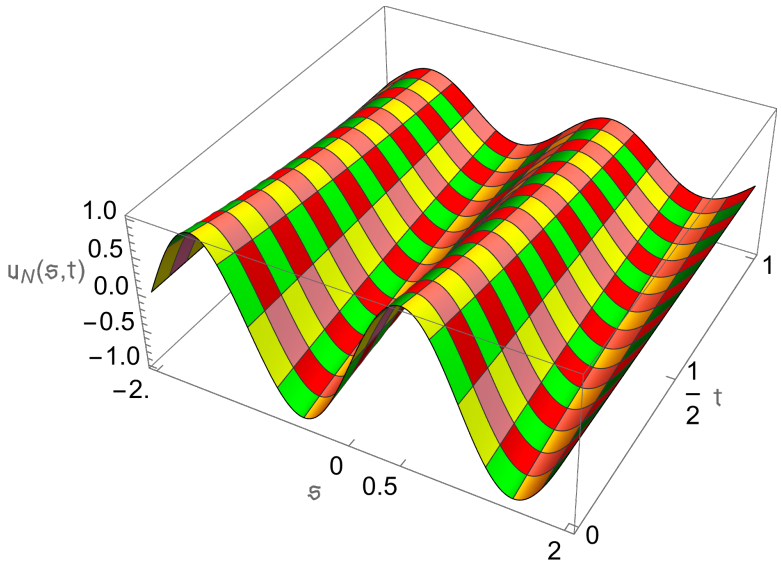

FIGURE 8. Three-dimensional plot of the approximate solution for Example 5.3 with $h = 0.02$, $\Delta t = 0.01$, and spatial domain $-2 \leq ś \leq 2$

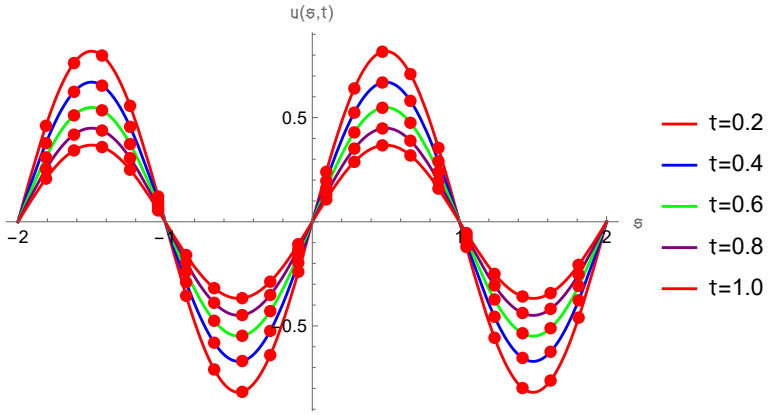

FIGURE 9. Plot illustrating the agreement between approximated and exact solutions for Example 5.3, where $h = 0.02, \Delta t = 0.01$, and $-2 \leq \mathfrak{s} \leq 2$

6. CONCLUSION

- The Galerkin finite element method has been used in this study to obtain numerical solutions for the sub-diffusion equation.
- Cubic B-spline functions have been effectively utilized as both trial and weight functions, providing flexibility and enhanced accuracy in the approximation of the solution.
- The transformation from local to global coordinate systems has been accurately achieved using local-to-global mapping techniques.
- The time derivative has been discretized using a finite difference formula, while the Crank-Nicolson method has been applied to handle the discretization of the unknown terms and spatial derivatives.
- Theoretical analyses of the stability and error estimation have been thoroughly carried out.
- The efficiency and effectiveness of the proposed scheme have been verified through four numerical experiments involving problems with known analytical solutions.
- The computational results have been compared with those reported in [14, 35, 37], showing clear improvements in accuracy.
- Graphical comparisons have been employed to visually exhibit the accuracy, efficiency, and robustness of the presented numerical approach.
- The overall results confirm that the proposed method is highly effective for solving this class of partial differential equations and can be extended to other related mathematical models.

7. ACKNOWLEDGMENTS

The authors are grateful to anonymous referees for their valuable suggestions, which significantly improved this manuscript.

FUNDING

This research received no funding.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] W. F. Ames, *Numerical Methods for Partial Differential Equations.* Acad. Press **(2014)**.
[2] P. B. Bochev and M. D. Gunzburger, *Least-Squares Finite Element Methods.* Springer Sci. & Bus. Media **166** (2009) 1-56.
[3] S. K. Bhowmik, *Piecewise Polynomial Approximation of a Nonlocal Phase Transitions Model.* J. Math. Anal. Appl. **420** (2014) 1069-1094.
[4] H. Caglar, M. Özer, and N. Caglar, *The Numerical Solution of the One-Dimensional Heat Equation by Using Third Degree B-Spline Functions.* Chaos Solitons Fractals **38**, No. 4 (2008) 1197-1201.
[5] J. Cooper, *The Heat and Wave Equations in Higher Dimensions.* In Introduction to Partial Differential Equations with MATLAB. Birkhäuser, Boston, MA (1998) 297-366.
[6] M. M. Cecchi and M. A. Pirozzi, *High Order Finite Difference Numerical Methods for Time-Dependent ConvectionDominated Problems.* Appl. Numer. Math. **55**, No. 3 (2005) 334-356.
[7] M. Dehghan, *Finite Difference Procedures for Solving a Problem Arising in Modeling and Design of Certain Optoelectronic Devices.* Math. Comput. Simul. **71**, No. 1 (2006) 16-30.
[8] S. Dhawan, S. Kapoor, S. Kumar, and S. Rawat, *Contemporary Review of Techniques for the Solution of Nonlinear Burgers Equation.* J. Comput. Sci. **3**, No. 5 (2012) 405-419.
[9] S. Dhawan, S. Kumar, and S. Chander, *A Comparative Study of Numerical Techniques for 2D Transient Heat Conduction Equation Using Finite Element Method.* In CSC (2009) 249-253.
[10] E. A. Elias, R. Cichota, H. H. Torriani, and Q. De Jong Van Lier, *Analytical Soil Temperature Model: Correction for Temporal Variation of Daily Amplitude.* Soil Sci. Soc. Am. J. **68**, No. 3 (2004) 784-788.
[11] C. A. Fletcher, *Computational Techniques for Fluid Dynamics 2: Specific Techniques for Different Flow Categories.* Springer Sci. & Bus. Media **2** (2012).
[12] G. Golub and J. M. Ortega, *Scientific Computing: An Introduction with Parallel Computing.* Acad. Press Inc., Boston (1993).
[13] A. Gorguis and W. K. B. Chan, *Heat Equation and Its Comparative Solutions.* Comput. Math. Appl. **55**, No. 12 (2008) 2973-2980.
[14] J. Goh and A. I. M. Ismail, *Cubic B-Spline Collocation Method for One Dimensional Heat and Advection Diffusion Equations.* J. Appl. Math. **2012**, No. 1 (2012) Article ID 458708.
[15] J. Goh, A. A. Majid, and A. I. M. Ismail, *Numerical Method Using Cubic B-Spline for the Heat and Wave Equation.* Comput. Math. Appl. **62**, No. 12 (2011) 4492-4498.
[16] M. Z. Gorgulu and I. R. K. Dursun, *The Galerkin Finite Element Method for Advection Diffusion Equation.* Sigma J. Eng. Nat. Sci. **37**, No. 1 (2019) 119-128.
[17] J. D. Hoffman and S. Frankel, *Numerical Methods for Engineers and Scientists.* CRC Press (2018).
[18] T. J. Hughes, *A Multidimensional Upwind Scheme with No Crosswind Diffusion.* Finite Elem. Methods Convect. Domin. Flows, AMD **34** (1979).
[19] T. J. Hughes, L. P. Franca, and G. M. Hulbert, *A New Finite Element Formulation for Computational Fluid Dynamics: VIII. The Galerkin/Least-Squares Method for Advective-Diffusive Equations.* Comput. Methods Appl. Mech. Eng. **73**, No. 2 (1989) 173-189.
[20] D. Irk, I. Dag, and M. Tombul, *Extended Cubic B-Spline Solution of the Advection Diffusion Equation.* KSCE J. Civ. Eng. **19**, No. 1 (2015) 929-934.

[21] M. K. Iqbal, M. Abbas, and N. Khalid, *New Cubic B-Spline Approximation for Solving Non-Linear Singular Boundary Value Problems Arising in Physiology.* Commun. Math. Appl. **9**, No. 3 (2018) 377-392.

[22] R. K. Jain and R. C. Mittal, *Numerical Solution of Convection-Diffusion Equation Using Cubic B-Splines Collocation Methods with Neumann Boundary Conditions.* Int. J. Appl. Math. Comput. **4**, No. 2 (2012) 115-127.

[23] S. Kapoor and S. Dhawan, *A Computational Technique for the Solution of Burgers Equation.* Int. J. Appl. Math. Mech. (2010) 84-95.

[24] A. Korkmaz and I. Dag, *Cubic B-Spline Differential Quadrature Methods for the Advection Diffusion Equation.* Int. J. Numer. Methods Heat Fluid Flow **22**, No. 8 (2012) 1021-1036.

[25] A. Korkmaz and I. Dag, *Quartic and Quintic B-Spline Methods for Advection Diffusion Equation.* Appl. Math. Comput. **274**, No. 1 (2016) 208-219.

[26] X. Lu, P. Tervola, and M. Viljanen, *An Efficient Analytical Solution to Transient Heat Conduction in a One Dimensional Hollow Composite Cylinder.* J. Phys. A: Math. Gen. **38**, No. 1 (2005) 101-145.

[27] A. Mohebbi and M. Dehghan, *High-Order Compact Solution of the One Dimensional Heat and Advection Diffusion Equations.* Appl. Math. Model. **34**, No. 10 (2010) 3071-3084.

[28] A. Mubashir Hayat, M. Abbas, H. Emadifar, A. S. Alzaidi, T. Nazir, and F. Aini Abdullah, *An Efficient Computational Scheme for Solving Coupled Time-Fractional Schrödinger Equation via Cubic B-Spline Functions.* PLoS One **19**, No. 5 (2024) e0296909.

[29] F. De Monte, *Transient Heat Conduction in One-Dimensional Composite Slab: A Natural Analytic Approach.* Int. J. Heat Mass Transf. **43**, No. 19 (2000) 3607-3619.

[30] I. J. Marwah and M. G. Chopra, *Transient Heat Transfer in a Slab with Heat Generation.* Def. Sci. J. **32**, No. 2 (1982) 143-149.

[31] J. A. Mackenzie, *KW Morton and DF Mayers. Numerical Solution of Partial Differential Equations.* Cambridge Univ. Press **38**, No. 3 (1995) 544-545.

[32] R. C. Mittal and R. Rohila, *The Numerical Study of Advection Diffusion Equations by the Fourth Order Cubic B-Spline Collocation Method.* Math. Sci. **14**, No. 4 (2020) 409-423.

[33] T. Nazir, M. Abbas, A. I. M. Ismail, A. A. Majid, and A. Rashid, *The Numerical Solution of Advection Diffusion Problems Using New Cubic Trigonometric B-Splines Approach.* Appl. Math. Model. **40**, No. 78 (2016) 4586-4611.

[34] E. Süli and D. F. Mayers, *An Introduction to Numerical Analysis.* Cambridge Univ. Press (2003).

[35] H. Sun and J. Zhang, *A High Order Compact Boundary Value Method for Solving One Dimensional Heat Equations.* Numer. Methods Partial Differ. Equ. **19**, No. 6 (2003) 846-857.

[36] S. Sunarsih, D. P. Sasongko, S. Sutrisno, G. A. Putri, and H. Hadiyanto, *Analysis of Waste Water Facultative Pond Using Advection-Diffusion Model Based on Explicit Finite Difference Method.* Environ. Eng. Res. **26**, No. 3 (2021) Article ID 190496.

[37] T. Tarmizi, E. Safitri, S. Munzir, and M. Ramli, *On the Numerical Solutions of a One-Dimensional Heat Equation: Spectral and Crank Nicolson Method.* AIP Conf. Proc. **2268**, No. 1 (2020) Article ID 050006.

[38] V. Thomée, *Galerkin Finite Element Methods for Parabolic Problems.* Springer Sci. & Bus. Media **25** (2007).

[39] G. Wang, *Unified and Extended Form of Three Types of Splines.* J. Comput. Appl. Math. **216**, No. 2 (2008) 498-508.

[40] G. Xu and G. Z. Wang, *AHT Bézier Curves and NUAHT B-Spline Curves.* J. Comput. Sci. Technol. **22**, No. 4 (2007) 597-607.

[41] L. Ya-Juan and W. Guo-Zhao, *Two Kinds of B-Basis of the Algebraic Hyperbolic Space.* J. Zhejiang Univ. Sci. A **6**, No. 7 (2005) 750-759.