Research Article

Mamta Kapoor*

A comparative study for the numerical approximation of 1D and 2D hyperbolic telegraph equations with UAT and UAH tension B-spline DQM

https://doi.org/10.1515/nleng-2022-0280
received July 2, 2022; accepted January 31, 2023

Abstract: Two numerical regimes for the one- and two-dimensional hyperbolic telegraph equations are contrasted in this article. The first implemented regime is uniform algebraic trigonometric tension B-spline DQM, while the second implemented regime is uniform algebraic hyperbolic tension B-spline DQM. The resulting system of ODEs is solved by the SSP RK43 method after the aforementioned equations are spatially discretized. To assess the success of chosen tactics, a comparison of errors is shown. The graphs can be seen, and it is asserted that the precise and numerical results are in agreement with one another. Analyses of convergence and stability are also given. It should be highlighted that there is a dearth of study on 1D and 2D hyperbolic telegraph equations. This aim of this study is to efficiently create results with fewer mistakes. These techniques would surely be useful for other higher-order nonlinear complex natured partial differential equations, including fractional equations, integro equations, and partial-integro equations.

Keywords: uniform algebraic trigonometric tension B-spline, uniform algebraic hyperbolic tension B-spline, differential quadrature method, hyperbolic telegraph equation

Nomenclature

UAT tension B-spline	uniform algebraic trigonometric tension B-spline
UAH tension B-spline	uniform algebraic hyperbolic tension B-spline
MUAT tension B-spline	modified uniform algebraic trigonometric tension B-spline
MUAH tension B-spline	modified uniform algebraic hyperbolic tension B-spline
x_l, x_r	lower and upper limits for spatial discretization
Δt	increment in time
t	specified time level
τ	tension parameter

1 Introduction

The model of the telegraph equation is mainly and mostly used in signal processing for the propagation of transmission of the electric impulses and wave theory process. A series of implementations is noticed of such models in the biomedical sciences and aerospace.

The foundational equations of atomic physics are based on the hyperbolic partial differential equations, which model the vibrations of structures. The telegraph equation appears in the investigation of wave phenomena and electrical signal propagation in a transmission line wire [1,2]. Examples of processes governed by equation include the propagation of acoustic waves in porous media of the Darcy type and parallel flows of viscous Maxwell fluids. For more details, see [3,4]. These equations are also used by biologists to examine the one-dimensional random movement of bugs along a hedge and the pulsed blood flow in arteries [5].

The hyperbolic partial differential equations have an essential role in modeling different basic equations in atomic physics [6]. It is one of the most useful in the insight of various physical processes in engineering and applied sciences. The hyperbolic telegraph equation is handy to model the structure's vibration, like machines, beams, and buildings. Various numerical schemes have

* **Corresponding author: Mamta Kapoor,** Department of Mathematics, Lovely Professional University, Phagwara, Punjab, 144411, India, e-mail: mamtakapoor.78@yahoo.com

been introduced in the literature regarding the development of solutions of partial differential equations [5,7–12]. 1D Telegraph equation is being solved by using different techniques, namely Taylor matrix method [13], unconditionally stable finite difference scheme [14], modified B-spline collocation scheme [10], implicit difference scheme [5], variational iteration method [15], cubic B-spline collocation scheme [16], and many more. 2D Telegraph equation has been solved by a meshless local Petrov–Galerkin method [17] and PDQM [18]. Ding and Zhang [19] provided a compact FD scheme. Some more noticeable and related work is specified in refs. [20–22].

Asif *et al.* [23] used a Haar Wavelet collocation approach to deal with 1D and 2D hyperbolic telegraph equation. Wang *et al.* [24] used a meshless approach for the solution of hyperbolic telegraph equation in (1+1) dimension. Zhou *et al.* [25] applied hybrid meshless regime for the second-order hyperbolic telegraph equation. Khan *et al.* [26] employed natural ADM for the fractional order hyperbolic telegraph equation. Wang and Hou [27] implemented a direct meshless approach to tackle 2D, second-order hyperbolic telegraph equation. Lin *et al.* [28] provided a meshless collocation regime for 2D hyperbolic telegraph equation. Ahmad *et al.* [29] provided the numerical study of multi dimensional hyperbolic telegraph equation *via* a local meshless method. Ahmad *et al.* [30] studied the integer order hyperbolic telegraph equation regarding physical and related sciences.

1.1 Hyperbolic telegraph equation

1.1.1 Governing equation I. [Hyperbolic telegraph equation in one dimension] [31]

$$u_{tt}(x,t) + 2\alpha u_t(x,t) + \beta^2 u(x,t) = u_{xx}(x,t) + g(x,t),\ x \in [a,b],\ t \geq 0. \tag{1}$$

I.C.s:

$$u(x,0) = g_1(x), \quad x \in [a,b], \tag{2}$$

$$u_t(x,0) = g_2(x), \quad x \in [a,b]. \tag{3}$$

B.C.s:

$$u(a,t) = \phi_1, \quad t > 0, \tag{4}$$

$$u(b,t) = \phi_2, \quad t > 0, \tag{5}$$

where g, g_1, g_2, ϕ_1, and ϕ_2 are considered as known functions. and $u_t = \frac{\partial u}{\partial t}$, $u_{xx} = \frac{\partial^2 u}{\partial x^2}$, and $u_{tt} = \frac{\partial^2 u}{\partial t^2}$.

1.1.2 Governing equation II. [Hyperbolic telegraph equation in two dimensions] [32]

$$u_{tt}(x,y,t) + 2\alpha u_t(x,y,t) + \beta^2 u(x,y,t) = u_{xx}(x,y,t) + u_{yy}(x,y,t) + h(x,y,t). \tag{6}$$

I.C.s:

$$u(x,y,0) = \phi_1(x,y,0),\ (x,y) \in [a,b] \times [c,d], \tag{7}$$

$$u_t(x,y,0) = \phi_2(x,y,0),\ (x,y) \in [a,b] \times [c,d]. \tag{8}$$

B.C.s: (Dirichlet boundary conditions)

$$u(a,y,t) = \psi_1(y,t),\ (x,y) \in \partial\Omega, \quad t > 0, \tag{9}$$

$$u(b,y,t) = \psi_2(y,t),\ (x,y) \in \partial\Omega, \quad t > 0, \tag{10}$$

$$u(x,c,t) = \psi_3(x,t),\ (x,y) \in \partial\Omega, \quad t > 0, \tag{11}$$

$$u(x,d,t) = \psi_4(x,t),\ (x,y) \in \partial\Omega, \quad t > 0. \tag{12}$$

1.2 Differential quadrature method

To solve partial differential equations, DQM is a numerical. It was initially proposed by Bellman *et al.* [33]. This section of DQM has seen a significant growth during the past couple decades. The scientific community has focused a lot of work on employing various basis functions to obtain numerical approximations for DQM. DQM uses the weighting summation of functional values at the supplied grid points of the computational domain to estimate the partial derivative of a smooth function.

Bellman *et al.* [33] provided two approaches for computation the weighting coefficients. Numerous numerical methods were created by various scholars to improve on Bellman's methods for determining the weighting coefficients. Quan and Chang [34] provided one of the most efficient and easily applicable techniques for finding weighting coefficients in DQM. After that Shu [35] proposed one of the easiest approaches to compute the weighting coefficients. Some latest work in this regard is provided in literature [36–44].

1.2.1 Main advantages of the study/novelty of the study

The current situation calls for the creation of innovative numerical regimes. It is noted that there is still a need for extensive numerical exploration of the 1D and 2D

hyperbolic telegraph equations. Therefore, the goal of this article is to efficiently produce results with fewer errors. Even if there are a lot of numerical techniques in the literature, new research is always possible. Some positive results are attained with the aid of recently created regimes.

Since the suggested method has produced some positive results, a large class of PDEs, including partial-integro differential equations and fractional order PDEs, can be effectively solved using these methods.

The framework of the article. To help readers understand our research, this document is organized into several sections and subsections.

- The implementation of numerical schemes is specified in Subsections 2.1 and 2.2.
- The DQM methodology is described in Section 2.3.
- Uniform algebraic trigonometric (UAT) tension B-spline DQM is developed in Subsection 2.4.
- Uniform algebraic hyperbolic (UAH) tension B-spline DQM is developed in Subsection 2.5.
- Discretization formulas are applied to the 1D and 2D hyperbolic telegraph equations in Section 3.
- Numerical tests on the 1D and 2D hyperbolic telegraph equations are presented in Section 4.
- Stability and convergence analysis is given in depth in Section 5.
- The conclusion is elaborated in Section 6.

2 Numerical schemes

2.1 Implementation of the numerical scheme upon 1D HT equation

Eq. (1) can be written as follows:

$$\begin{cases} w_t + 2\alpha w + \beta^2 u = A u_{xx} + g(x,t), \\ \text{Considered } u_t = w, \\ \dfrac{\partial u}{\partial t} = w. \end{cases} \tag{13}$$

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t} w_i = -2\alpha w_i - \beta^2 u_i + \left[\displaystyle\sum_{k=1}^{n} a_{i,k}^{(2)} u(x_i, t) \right], \\ \dfrac{\mathrm{d}u_i}{\mathrm{d}t} = w_i. \end{cases} \tag{14}$$

I.C.:

$$u(x_i, 0) = u_0(x_i) \quad \text{and} \quad w(x_i, 0) = w_0(x_i).$$

At the boundary points, Dirichlet boundary conditions can be applied immediately and without modification. MUAT DQM or MUAH DQM can be used when the Neumann or mixed boundary conditions are present.

2.2 Implementation of the numerical scheme upon 2D HT equation

Eq. (6) can be written as follows:

$$\begin{cases} w_t = -2\alpha w - \beta^2 u = u_{xx} + u_{yy} + h(x,y,t), \\ \text{Considered } u_t = w, \\ \dfrac{\partial u}{\partial t} = w. \end{cases} \tag{15}$$

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t} w_{ij} = -2\alpha w_{ij} - \beta^2 u_{ij} + \left[\displaystyle\sum_{k=1}^{n} a_{i,k}^{(2)} u(x_k, y_j, t) \right. \\ \qquad\qquad \left. + \displaystyle\sum_{k=1}^{m} a_{j,k}^{(2)} u(x_i, y_k, t) \right], \\ \dfrac{\mathrm{d}u_{ij}}{\mathrm{d}t} = w_{ij}. \end{cases} \tag{16}$$

I.C.:

$$u(x_i, y_j, 0) = u_0(x_i, y_j) \quad \text{and} \quad w(x_i, y_j, 0) = w_0(x_i, y_j).$$

Dirichlet boundary conditions can be directly implemented at the boundary points without any change. In the case of the Neumann or mixed boundary conditions, MUAT DQM or MUAH DQM can be implemented.

2.3 Methodology

A key tool for numerical discretization is DQM. Obtaining the weighting coefficient values is the primary viewpoint of DQM. Numerous test functions are suggested in the literature as a means of achieving the weighting coefficients in DQM. Finding the weighting coefficients is the central idea in DQM; for the aforementioned task, a number of test functions are provided in the literature, and several numerical regimes have been produced utilizing these test functions.

As basis functions for the creation of weighting coefficients, we have used modified cubic UAT tension B-splines and modified cubic UAH tension B-splines in the current study. To solve hyperbolic telegraph equations in one and two dimensions, DQM has never been

Table 1: The values of $UATB_{i,4}$ and $UATB'_{i,4}$ at different node points

	γ_{m-2}	γ_{m-1}	γ_m	γ_{m+1}	γ_{m+2}
$UATB_{i,4}(\gamma)$	0	a_1	a_2	a_3	0
$UATB_{i,4}'(\gamma)$	0	a_4	0	a_5	0

used in literature using UAT tension B-splines or UAH tension B-splines. This approach will surely give academics new perspectives to work with when tackling complex natured PDEs in the future.

First, 1D hyperbolic telegraph equation is taken into the account. Considered $(n + 1)$ number of grid points in the domain $[a, b]$ such as; $a = x_0 < x_1 < x_2 < x_3 < \cdots < x_n = b$. These grid points have uniform distribution and taken step size is $h = x_{i+1} - x_i$, in the x-direction. The rth approximation of $u(x, t)$ is given as follows:

$$u_x^{(r)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(r)} u(x_j), \tag{17}$$

where $u_x^{(1)}(x_i)$ is the first-order partial derivative of u at x_i, respectively, and $u_x^{(2)}(x_i)$ is the second-order partial derivative of u at grid point x_i. $u(x_j)$ is the functional value of u at specified grid points.

Similarly, 1D discretization can be transformed into 2D discretization. Considered the computational domain as $[a, b] \times [c, d]$, where $[a, b] \times [c, d]$ is given with $N + 1$ and $M + 1$ grid points s.t. $a = x_0 < x_1 < x_2 < x_3 < \cdots < x_N = b$ and $c = y_0 < y_1 < y_2 < y_3 < \cdots < y_M = d$ with uniform step sizes, given by $\Delta x = x_{i+1} - x_i$ and $\Delta y = y_{j+1} - y_j$, respectively, in x and y directions. rth spatial partial derivatives of u with respect to x (keeping y_j fixed) and y (keeping x_i fixed), respectively, can be obtained as follows:

$$\frac{\partial^r u(x_i, y_j, t)}{(\partial x^r)} = \sum_{k=1}^{N} a_{ik}^{(r)} u(x_k, y_j, t), \tag{18}$$

$$\frac{\partial^r u(x_i, y_j, t)}{(\partial y^r)} = \sum_{k=1}^{N} b_{jk}^{(r)} u(x_i, y_k, t), \tag{19}$$

where $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ are known as the weighting coefficients of rth order spatial partial derivatives with respect to x and y, respectively (Table 1).

2.4 Applied numerical method I and solution of the model (UAT tension B-spline DQM) [21–23]

$$\mathrm{UAT}_{m,4}(\gamma) = \begin{cases} (1)\ \dfrac{\theta_{m,3}\theta_{m,2}}{\tau \sin(\tau h)}\left[(\gamma - \gamma_{m-2}) - \dfrac{[\sin(\tau(\gamma - \gamma_{m-2}))]}{\tau}\right], [\gamma_{m-2}, \gamma_{m-1}] \\ (2)\ \theta_{m,3}\left[\dfrac{\theta m, 2}{\tau \sin(\tau h)}\left\{(\gamma_{m-1} - \gamma_{m-2}) - \dfrac{\sin[\tau(\gamma_{m-1} - \gamma_{m-2})]}{\tau}\right\} + (\gamma - \gamma_{m-1})\right. \\ \quad - \dfrac{\theta_{m,2}}{\tau \sin(\tau h)}\left\{(\gamma - \gamma_{m-1}) - \dfrac{1}{\tau}(\sin(\tau(\gamma - \gamma_m)) + \sin(\tau(\gamma_m - \gamma)))\right\} \\ \quad \left. - \dfrac{\theta_{m+1,2}}{\tau \sin(\tau h)}\left\{(\gamma - \gamma_{m-1}) - \dfrac{(\sin(\tau(\gamma - \gamma_{m-1})))}{\tau}\right\}\right], [\gamma_{m-1}, \gamma_m] \\ (3)\ 1 - \dfrac{\theta_{m,3}\delta m + 1, 2}{\tau \sin(\tau h)}\left\{(\gamma_{m+1} - \gamma) + \dfrac{\sin(\tau(\gamma - \gamma_{m+1}))}{\tau}\right\} \\ \quad - \theta_{m+1,3}\left[\dfrac{\theta_{m+1,2}}{\tau \sin(\tau h)}\left\{(\gamma_m - \gamma_{m-1}) - \dfrac{\sin(\tau(\gamma_m - \gamma_{m-1}))}{\tau}\right\} + (\gamma - \gamma_m)\right. \\ \quad - \dfrac{\theta_{m+1,2}}{\tau \sin(\tau h)}\left\{(\gamma - \gamma_m) - \dfrac{(\sin(\tau(\gamma - \gamma_{m+1})) - \sin(\tau(\gamma_m - \gamma_{m+1})))}{\tau}\right\} \\ \quad \left. - \dfrac{\theta_{m+2,2}}{\tau \sin(\tau h)}\{(\gamma - \gamma_m) - \dfrac{\sin(\tau(\gamma - \gamma_m))}{\tau}\}\right], [\gamma_m, \gamma_{m+1}] \\ (4)\ \dfrac{\theta_{m+1,3}\theta_{m+2,2}}{\tau \sin(\tau h)}\left[(\gamma_{m+2} - \gamma) + \dfrac{\sin(\tau(\gamma - \gamma_{m+2}))}{\tau}\right], [\gamma_{m+1}, \gamma_{m+2}] \\ (5)\ 0, \quad \text{elsewhere} \end{cases}$$

Evaluated values of $\mathrm{UATB}_{i,4}(y)$ and $\mathrm{UATB}'_{i,4}(y)$ at the node points:

$$a_1 = \left[\frac{1}{4h\sin^2\left(\frac{\tau h}{2}\right)}\right]\left[h - \sin\left(\frac{\tau h}{\tau}\right)\right],$$
$$a_2 = 1 - \left[\frac{1}{2h\sin^2\left(\frac{\tau h}{2}\right)}\right]\left[h - \frac{\sin(\tau h)}{\tau}\right],$$
$$a_3 = \frac{1}{4h\sin^2\left(\frac{\tau h}{2}\right)}\left[h - \sin\left(\frac{\tau h}{\tau}\right)\right],$$
$$a_4 = \frac{1}{2h},$$
$$a_5 = \frac{-1}{2h}.$$

Evaluation of $\theta_{i,2}$ and $\theta_{i,3}$:

$$\theta_{i,2} = \frac{(\tau\sin(\tau h))}{(2[\sin^2(\tau(\gamma_{i-1} - \gamma_{i-2})/2) + \sin^2(\tau(\gamma_i - \gamma_{i-1})/2)])},$$
$$\theta_{i,3} = \frac{1}{M_1 + M_2 - M_3 - M_4 + M_5},$$

where

$$M_1 = \left[\frac{2\theta_{i,2}}{\tau\sin(\tau h)}\right]\left[\left(\frac{\gamma_{i-1} - \gamma_{i-2}}{2}\right) - \sin\left(\frac{\tau(\gamma_{i-1} - \gamma_{i-2})}{2\tau}\right)\right],$$
$$M_2 = \gamma_i - \gamma_{i-1},$$
$$M_3 = \left[\frac{2\theta_{i,2}}{\tau\sin(\tau h)}\right]\left[\left(\frac{\gamma_i - \gamma_{i-1}}{2}\right) - \sin\left(\frac{\tau(\gamma_i - \gamma_{i-1})}{2\tau}\right)\right],$$
$$M_4 = \left[\frac{2\theta_{i+1,2}}{\tau\sin(\tau h)}\right]\left[\left(\frac{\gamma_i - \gamma_{i-1}}{2}\right) - \sin\left(\frac{\tau(\gamma_i - \gamma_{i-1})}{2\tau}\right)\right],$$
$$M_4 = \left[\frac{2\theta_{i+1,2}}{\tau\sin(\tau h)}\right]\left[\left(\frac{\gamma_{i+1} - \gamma_i}{2}\right) - \sin\left(\frac{\tau(\gamma_i - \gamma_{i+1})}{2\tau}\right)\right],$$

where $\{\mathrm{MUATB}_0(x),\ \mathrm{MUATB}_1(x),\ \cdots,\ \mathrm{MUATB}_N(x),\ \mathrm{MUATB}_{N+1}(x)\}$ forms a basis over a given domain. To improve the results, modified cubic UAT tension B-spline can be implemented, so that the obtained matrix system will become diagonally dominant [7]. By using the following set of equations, improvised values can be obtained.

$$\begin{cases} \phi_1(x) = \mathrm{UAT}_1(x) - 2\mathrm{UAT}_0(x) \\ \phi_2(x) = \mathrm{UAT}_2(x) - \mathrm{UAT}_0(x) \\ \phi_j(x) = \mathrm{UAT}_j(x), [j = 3, 4, 5, \ldots, N-2] \\ \phi_{N-1}(x) = \mathrm{UAT}_{N-1}(x) - \mathrm{UAT}_{N+1}(x) \\ \phi_N(x) = \mathrm{UAT}_N(x) - 2\mathrm{UAT}_{N+1}(x). \end{cases} \tag{20}$$

Determination of weighting coefficients using method I.

We have the formula to approximate first-order derivative:

$$\mathrm{MUATB}_k^{(1)}(x_i) = \sum_{j=1}^{n} w_{ij}^{(1)}\mathrm{MUATB}_k(x_j), \tag{21}$$

where $i = 1, 2, 3, \ldots, n$ and $k = 1, 2, 3, \ldots, n$.

At grid point x_1:

For $k = 1$:

$$\mathrm{MUATB}'_1(x_1) = \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUATB}_1(x_j) = w_{11}^{(1)}[a_2 + 2a_3] + w_{12}^{(1)}[a_3],$$

For $k = 2$:

$$\mathrm{MUATB}'_2(x_1) = \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUATB}_2(x_j) = w_{11}^{(1)}[a_1 - a_3] + w_{12}^{(1)}[a_2] + w_{13}^{(1)}[a_3],$$

For $k = 3$:

$$\mathrm{MUATB}'_3(x_1) = \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUATB}_3(x_j) = w_{12}^{(1)}[a_1] + w_{13}^{(1)}[a_2] + w_{14}^{(1)}[a_3],$$

For $k = 4$:

$$\mathrm{MUATB}'_4(x_1) = \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUATB}_4(x_j) = w_{13}^{(1)}[a_1] + w_{14}^{(1)}[a_2] + w_{15}^{(1)}[a_3],$$
$$\vdots\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\vdots\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\vdots\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

For $k = n - 2$:

$$\mathrm{MUATB}'_{n-2}(x_1) = \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUATB}_{n-2}(x_j) = w_{1,n-3}^{(1)}[a_1] + w_{1,n-2}^{(1)}[a_2] + w_{1,n-1}^{(1)}[a_3],$$

For $k = n - 1$:

$$\mathrm{MUATB}'_{n-1}(x_1) = \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUATB}_{n-1}(x_j) = w_{1,n-2}^{(1)}[a_1] + w_{1,n-1}^{(1)}[a_2] + w_{1,n}^{(1)}[a_3 - a_1],$$

For $k = n$:

$$\mathrm{MUATB}'_n(x_1) = \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUATB}_n(x_j) = w_{1,n-1}^{(1)}[a_1] + w_{1,n}^{(1)}[a_2 + 2a_1].$$

From the aforementioned set of equation at grid point x_1 and for the values of $k = 1, 2, 3, \ldots, n$, the following tridiagonal system of algebraic equations will be obtained:

$$M\vec{w}^{(1)}[i] = \vec{Y}[i], \quad \text{where } i = 1, 2, 3, \ldots, n,$$

$$M = \begin{bmatrix} a_2 + 2a_3 & a_3 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ a_1 - a_3 & a_2 & a_3 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & a_1 & a_2 & a_3 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & a_1 & a_2 & a_3 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & a_1 & a_2 & a_3 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & a_1 & a_2 & a_3 - a_1 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 & a_1 & a_2 + 2a_1 \end{bmatrix},$$

$$\vec{w}^{(1)}[1] = \begin{bmatrix} w_{1,1}^{(1)} \\ w_{1,2}^{(1)} \\ w_{1,3}^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ w_{1,n-1}^{(1)} \\ w_{1,n}^{(1)} \end{bmatrix}, \quad \text{and}$$

$$\vec{Y}[1] = \begin{bmatrix} \text{MUATB}_1'(x_1) \\ \text{MUATB}_2'(x_1) \\ \text{MUATB}_3'(x_1) \\ \vdots \\ \vdots \\ \vdots \\ \text{MUATB}_{n-1}'(x_1) \\ \text{MUATB}_n'(x_1) \end{bmatrix} = \begin{bmatrix} 2a_5 \\ a_4 - a_5 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

At grid point x_2:

$$\vec{w}^{(1)}[2] = \begin{bmatrix} w_{2,1}^{(1)} \\ w_{2,2}^{(1)} \\ w_{2,3}^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ w_{2,n-1}^{(1)} \\ w_{2,n}^{(1)} \end{bmatrix} \quad \text{and}$$

$$\vec{Y}[2] = \begin{bmatrix} \text{MUATB}_1'(x_2) \\ \text{MUATB}_2'(x_2) \\ \text{MUATB}_3'(x_2) \\ \vdots \\ \vdots \\ \vdots \\ \text{MUATB}_{n-1}'(x_2) \\ \text{MUATB}_n'(x_2) \end{bmatrix} = \begin{bmatrix} a_5 \\ 0 \\ a_4 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}.$$

.. ..
.. ..
.. ..

At grid point x_n:

$$\vec{w}^{(1)}[n] = \begin{bmatrix} w_{n,1}^{(1)} \\ w_{n,2}^{(1)} \\ w_{n,3}^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ w_{n,n-1}^{(1)} \\ w_{n,n}^{(1)} \end{bmatrix} \quad \text{and}$$

$$\vec{Y}[n] = \begin{bmatrix} \text{MUATB}_1'(x_n) \\ \text{MUATB}_2'(x_n) \\ \text{MUATB}_3'(x_n) \\ \vdots \\ \vdots \\ \vdots \\ \text{MUATB}_{n-1}'(x_n) \\ \text{MUATB}_n'(x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ a_5 - a_4 \\ 2a_4 \end{bmatrix}.$$

2.5 Applied numerical method II and solution of the model (UAH tension B-spline DQM) [45–47]

$$
\psi_{i,4}(y)
= \begin{cases}
(1)\ \dfrac{\theta_{i,3}\theta_{i,2}}{\tau\sinh(\tau h)}\left[(\gamma_{i-2}-y)+\dfrac{[\sinh(\tau(y-\gamma_{i-2}))]}{\tau}\right], [\gamma_{i-2},\gamma_{i-1}] \\
(2)\ \theta_{i,3}\left[\dfrac{\theta i,2}{\tau\sinh(\tau h)}\left\{(\gamma_{i-2}-\gamma_{i-1})+\dfrac{\sinh[\tau(\gamma_{i-1}-\gamma_{i-2})]}{\tau}\right\}+(y-\gamma_{i-1})\right. \\
\quad -\dfrac{\theta_{i,2}}{\tau\sinh(\tau h)}\left\{(\gamma_{i-1}-y)+\dfrac{1}{\tau}(\sinh(\tau(y-\gamma_i))+\sinh(\tau(\gamma_i-\gamma_{i-1})))\right\} \\
\quad \left.-\dfrac{\theta_{i+1,2}}{\tau\sinh(\tau h)}\left\{(\gamma_{i-1}-y)+\dfrac{(\sinh(\tau(y-\gamma_{i-1})))}{\tau}\right\}\right] \\
\quad -\dfrac{\theta_{i+1,3}\theta_{i+1,2}}{\tau\sinh(\tau h)}\left\{(\gamma_{i-1}-y)+\dfrac{\sinh(\tau(x-x_{i-1}))}{\tau}\right\}, [\gamma_{i-1},\gamma_i] \\
(3)\ 1-\dfrac{\theta_{i,3}\theta_{i+1,2}}{\tau\sinh(\tau h)}\left\{(y-\gamma_{i+1})-\dfrac{\sinh(\tau(y-\gamma_{i+1}))}{\tau}\right\}-\theta_{i+1,3}\left[\dfrac{\theta_{i+1,2}}{\tau\sinh(\tau h)}\left\{(\gamma_{i-1}-\gamma_i)+\dfrac{\sinh(\tau(\gamma_i-\gamma_{i-1}))}{\tau}\right\}\right. \\
\quad +(y-\gamma_i)-\dfrac{\theta_{i+1,2}}{\tau\sinh(\tau h)}\left\{(\gamma_i-y)+\dfrac{(\sinh(\tau(y-\gamma_{i+1}))-\sinh(\tau(\gamma_i-\gamma_{i+1})))}{\tau}\right\} \\
\quad \left.-\dfrac{\theta_{i+2,2}}{\tau\sin(\tau h)}\left\{(\gamma_i-y)+\dfrac{\sinh(\tau(y-\gamma_i))}{\tau}\right\}\right], [\gamma_i,\gamma_{i+1}] \\
(4)\ \dfrac{\theta_{i+1,3}\theta_{i+2,2}}{\tau\sinh(\tau h)}\left[(y-\gamma_{i+2})-\dfrac{\sinh(\tau(y-\gamma_{i+2}))}{\tau}\right], [\gamma_{i+1},\gamma_{i+2}] \\
(5)\ 0, \quad \text{elsewhere}
\end{cases} \tag{22}
$$

Evaluated values of $\mathrm{UAHB}_{i,4}(y)$ and $\mathrm{UAHB}'_{i,4}(y)$ at the node points

$$
b_1 = \left[\frac{1}{4h\sinh^2\left(\frac{\tau h}{2}\right)}\right]\left[\frac{\sinh(\tau h)}{\tau}-h\right],
$$

$$
b_2 = 1-\left[\frac{1}{2h\sinh^2\left(\frac{\tau h}{2}\right)}\right]\left[\frac{\sinh(\tau h)}{\tau}-h\right],
$$

$$
b_3 = \left[\frac{1}{4h\sinh^2\left(\frac{\tau h}{2}\right)}\right]\left[\frac{\sinh(\tau h)}{\tau}-h\right],
$$

$$
b_4 = \frac{1}{2h},
$$

$$
b_4 = \frac{-1}{2h}.
$$

Evaluation of $\theta_{i,2}$ and $\theta_{i,3}$:

$$
\theta_{i,2} = \frac{\tau\sinh(\tau h)}{2\left[\sinh^2\left(\frac{\tau(\gamma_{i-1}-\gamma_{i-2})}{2}\right)+\sinh^2\left(\frac{\tau(\gamma_i-\gamma_{i-1})}{2}\right)\right]},
$$

$$
\theta_{i+1,2} = \frac{\tau\sinh(\tau h)}{2\left[\sinh^2\left(\frac{\tau(\gamma_i-\gamma_{i-1})}{2}\right)+\sinh^2\left(\frac{\tau(\gamma_{i+1}-\gamma_i)}{2}\right)\right]},
$$

$$
\theta_{i+2,2} = \frac{\tau\sinh(\tau h)}{2\left[\sinh^2\left(\frac{\tau(\gamma_{i+1}-\gamma_i)}{2}\right)+\sinh^2\left(\frac{\tau(\gamma_{i+2}-\gamma_{i+1})}{2}\right)\right]},
$$

$$
\theta_{i,3} = \frac{1}{\frac{\theta_{i,2}}{\tau\sinh(\tau h)}[A_1]+h-\frac{\theta_{i,2}}{\tau\sinh(\tau h)}[A_1]-\frac{\theta_{i+1,2}}{\tau\sinh(\tau h)}[A_1]+\frac{\theta_{i+1,2}}{\tau\sinh(\tau h)}[A_1]},
$$

where $A_1 = -h+\frac{\sinh(\tau h)}{h}$.

$$\theta_{i+1,3} = \frac{1}{\frac{\theta_{i+1,2}}{\tau\sinh(\tau h)}[A_1] + h - \frac{\theta_{i+1,2}}{\tau\sinh(\tau h)}[A_1] - \frac{\theta_{i+2,2}}{\tau\sinh(\tau h)}[A_1] + \frac{\theta_{i+2,2}}{\tau\sinh(\tau h)}[A_1]},$$

where $A_1 = -h + \frac{\sinh(\tau h)}{h}$, where, h is the length of the provided interval.

Here $\{\mathrm{MUAHB}_0(\gamma),\ \mathrm{MUAHB}_1(\gamma),\ \cdots,\ \mathrm{MUAHB}_N(\gamma),\ \mathrm{MUAHB}_{N+1}(\gamma)\}$ forms a basis over given domain. To improve the results, modified cubic UAH tension B-spline can be implemented by in a way such that the obtained matrix system will become diagonally dominant [7]. By using the following set of equations, improvised values can be obtained.

$$\begin{cases} \psi_1(\gamma) = \mathrm{UAH}_1(\gamma) - 2\mathrm{UAH}_0(\gamma) \\ \psi_2(\gamma) = \mathrm{UAH}_2(\gamma) - \mathrm{UAH}_0(\gamma) \\ \psi_k(\gamma) = \mathrm{UAH}_j(\gamma),\ [k = 3, 4, 5, \ldots, n-2] \\ \psi_{n-1}(\gamma) = \mathrm{UAH}_{n-1}(\gamma) - \mathrm{UAH}_{n+1}(\gamma) \\ \psi_n(\gamma) = \mathrm{UAH}_n(\gamma) - 2\mathrm{UAH}_{n+1}(\gamma). \end{cases} \tag{23}$$

Determination of weighting coefficients using method II.

We have the formula to approximate first order derivative:

$$\mathrm{MUAH}_k^{(1)}(\gamma_i) = \sum_{j=1}^{n} w_{ij}^{(1)}\mathrm{MUAH}_k(\gamma_j), \tag{24}$$

where $i = 1, 2, 3, \ldots, n$ and $k = 1, 2, 3, \ldots, n$.

At grid point x_1: By applying the formulas from Eqs (22) and (23) and the values from Table 2 in Eq. (24), we will obtain the following set of equations, for different values of k,

For $k = 1$:

$$\begin{aligned} \mathrm{MUAH}_1'(\gamma_1) &= \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUAH}_1(\gamma_j) \\ &= w_{11}^{(1)}[b_2 + 2b_3] + w_{12}^{(1)}[b_3]. \end{aligned}$$

For $k = 2$:

$$\begin{aligned} \mathrm{MUAH}_2'(\gamma_1) &= \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUAH}_2(\gamma_j) \\ &= w_{11}^{(1)}[b_1 - b_3] + w_{12}^{(1)}[b_2] + w_{13}^{(1)}[b_3]. \end{aligned}$$

Table 2: The values of UAH tension B-spline of order 4, *i.e.*, $\mathrm{UAHB}_{i,4}(\gamma)$ and $\mathrm{UAHB}'_{i,4}(\gamma)$ at different node points

	γ_{i-2}	γ_{i-1}	γ_i	γ_{i+1}	γ_{i+2}
$\mathrm{UAHB}_{i,4}(\gamma)$	0	b_1	b_2	b_3	0
$\mathrm{UAHB}'_{i,4}(\gamma)$	0	b_4	0	b_5	0

For $k = 3$:

$$\begin{aligned} \mathrm{MUAH}_3'(\gamma_1) &= \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUAH}_3(\gamma_j) \\ &= w_{12}^{(1)}[b_1] + w_{13}^{(1)}[b_2] + w_{14}^{(1)}[b_3]. \end{aligned}$$

$$\begin{array}{l} \vdots\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \vdots\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \vdots\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{array}$$

For $k = n$:

$$\begin{aligned} \mathrm{MUAH}_n'(\gamma_1) &= \sum_{j=1}^{n} w_{1j}^{(1)}\mathrm{MUAH}_n(\gamma_j) \\ &= w_{1n-1}^{(1)}[b_1] + w_{1n}^{(1)}[b_2 + 2b_1]. \end{aligned}$$

From the aforementioned set of equation at grid point γ_1 and for the values of $k = 1, 2, 3, \ldots, n$, we will obtain the following tri-diagonal system of algebraic equations:

$$M\vec{w}^{(1)}[i] = \vec{Y}[i], \quad \text{where } i = 1, 2, 3, \ldots, n.$$

$$M = \begin{bmatrix} b_2 + 2b_3 & b_3 & 0 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots \\ b_1 - b_3 & b_2 & b_3 & 0 & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & b_1 & b_2 & b_3 & \ldots & \ldots & \ldots & \ldots & \ldots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \ldots & \ldots & \ldots & \ldots & \ldots & b_1 & b_2 & b_3 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & b_1 & b_2 & b_3 - b_1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & b_1 & b_2 + 2b_1 \end{bmatrix},$$

$$\vec{w}^{(1)}[1] = \begin{bmatrix} w_{11}^{(1)} \\ w_{12}^{(1)} \\ w_{13}^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ w_{1n-1}^{(1)} \\ w_{1n}^{(1)} \end{bmatrix},$$

and

$$\vec{Y}^{(1)}[1] = \begin{bmatrix} \mathrm{MUAH}_1'(\gamma_1) \\ \mathrm{MUAH}_2'(\gamma_1) \\ \mathrm{MUAH}_3'(\gamma_1) \\ \vdots \\ \vdots \\ \vdots \\ \mathrm{MUAH}_n - 1'(\gamma_1) \\ \mathrm{MUAH}_n'(\gamma_1) \end{bmatrix} = \begin{bmatrix} 2b_5 \\ b_4 - b_5 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}.$$

At grid point γ_2: By applying the formulas from Eqs. (22) and (23) and the values from Table 2 in the Eq. (24), we will obtain the following set of equations, for different values of k,

$$M = \begin{bmatrix} b_2+2b_3 & b_3 & 0 & 0 & \dots & \dots & \dots & \dots & \dots \\ b_1-b_3 & b_2 & b_3 & 0 & \dots & \dots & \dots & \dots & \dots \\ 0 & b_1 & b_2 & b_3 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots & \dots & b_1 & b_2 & b_3 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & b_1 & b_2 & b_3-b_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & b_1 & b_2+2b_1 \end{bmatrix},$$

$$\overrightarrow{w}^{(1)}[2] = \begin{bmatrix} w_{21}^{(1)} \\ w_{22}^{(1)} \\ w_{23}^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ w_{2n-1}^{(1)} \\ w_{2n}^{(1)} \end{bmatrix},$$

and

$$\overrightarrow{Y}^{(1)}[2] = \begin{bmatrix} \mathrm{MUAH}_1'(\gamma_2) \\ \mathrm{MUAH}_2'(\gamma_2) \\ \mathrm{MUAH}_3'(\gamma_2) \\ \vdots \\ \vdots \\ \vdots \\ \mathrm{MUAH}_n - 1'(\gamma_2) \\ \mathrm{MUAH}_n'(\gamma_2) \end{bmatrix} = \begin{bmatrix} b_5 \\ 0 \\ b_4 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}.$$

$\therefore$ .....
$\therefore$ .....
$\therefore$ .....

At grid point γ_n: By applying the formulas from Eqs. (22) and (23) and the values from Table 2 in the Eqs. (24), we will obtain the following set of equations, for different values of k,

$$M = \begin{bmatrix} b_2+2b_3 & b_3 & 0 & 0 & \dots & \dots & \dots & \dots & \dots \\ b_1-b_3 & b_2 & b_3 & 0 & \dots & \dots & \dots & \dots & \dots \\ 0 & b_1 & b_2 & b_3 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots & \dots & b_1 & b_2 & b_3 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & b_1 & b_2 & b_3-b_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & b_1 & b_2+2b_1 \end{bmatrix},$$

$$\overrightarrow{w}^{(1)}[n] = \begin{bmatrix} w_{n1}^{(1)} \\ w_{n2}^{(1)} \\ w_{n3}^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ w_{nn-1}^{(1)} \\ w_{nn}^{(1)} \end{bmatrix},$$

and

$$\overrightarrow{Y}^{(1)}[2] = \begin{bmatrix} \mathrm{MUAH}_1'(\gamma_n) \\ \mathrm{MUAH}_2'(\gamma_n) \\ \mathrm{MUAH}_3'(\gamma_n) \\ \vdots \\ \vdots \\ \vdots \\ \mathrm{MUAH}_n - 1'(\gamma_n) \\ \mathrm{MUAH}_n'(\gamma_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ b_5 - b_4 \\ 2b_4 \end{bmatrix}.$$

Similarly, to find weighting coefficients of order r where $r \geq 2$, we use the following equation:

$$w_{ij}^{(r)} = r[w_{ij}^{(1)} w_{ii}^{(r-1)} - (w_{ij}^{(r-1)})/(\gamma_i - \gamma_j)], \quad \text{for } i \neq j, \tag{25}$$

$$w_{ii}^{(r)} = -\sum_{j=1, j\neq i}^{N} w_{ij}^{(r)}, \quad \text{for } i = j. \tag{26}$$

So, with the help of the aforementioned equations, second-order and higher-order weighting coefficients can be easily obtained.

3 Implementation of the scheme

3.1 Implementation of the scheme on 1D HT equation

In the present article, $u_t = v$ is considered as the required transformation. On using this transformation in 1D hyperbolic telegraph equation, the following coupled system of PDEs will be obtained.

$$u_t(x,t) = v(x,t), \tag{27}$$

$$v_t(x,t) + 2\alpha v(x,t) + \beta^2 u(x,t) = u_{xx}(x,t) + g(x,t). \tag{28}$$

On using the discretized values of partial derivatives in the aforementioned coupled system of equations, the following ODE system of equations will be obtained.

$$\frac{(\mathrm{d}u_i)}{\mathrm{d}t} = v_i,$$
$$\frac{(\mathrm{d}v_i)}{\mathrm{d}t} = -2\alpha v(x_i, t) - \beta^2 u(x_i, t) + u_{xx}(x_i, t) + g(x_i, t),$$

or

$$\frac{(\mathrm{d}u_i)}{\mathrm{d}t} = v_i,$$
$$\frac{(\mathrm{d}v_i)}{\mathrm{d}t} = L(u_i),$$

where $L(u_i) = -2\alpha v_i(t) - \beta^2 u_i(t) + (u_{xx})_i(t) + g_i(t)$. The aforementioned system of ODEs is solved using SSP-RK43 scheme using the following formulae [48]:

$$u^{(1)} = u^m + (\Delta t/2)L(u^m),$$
$$u^{(2)} = u^{(1)} + (\Delta t/2)L(u^{(1)}),$$
$$u^{(3)} = (2/3)u^m + (u^2/3) + (\Delta t/6)L(u^{(2)}),$$
$$u^{m+1} = u^{(3)} + (\Delta t/2)L(u^{(3)}).$$

3.2 Implementation of scheme on HT 2D equation

To solve 2D hyperbolic telegraph equation transformation is considered as follows:

$$u_t(x, y, t) = v(x, y, t).$$

By using the aforementioned transformation in 2D hyperbolic telegraph equation, we can obtain the following set of coupled PDEs:

$$u_t(x, y, t) = v(x, y, t), \tag{29}$$

$$v_t(x, y, t) + 2\alpha v(x, y, t) + \beta^2 u(x, y, t) = u_{xx}(x, y, t) + u_{yy}(x, y, t) + h(x, y, t). \tag{30}$$

By using discretization formulae in the aforementioned set of coupled PDEs, the following system of ODEs will be fetched as follows:

$$\frac{\mathrm{d}u(x_i, y_j, t)}{\mathrm{d}t} = v(x_i, y_j, t),$$
$$\frac{\mathrm{d}v(x_i, y_j, t)}{\mathrm{d}t} = -2\alpha v(x_i, y_j, t) - \beta^2 u(x_i, y_j, t) + u_{xx}(x_i, y_j, t) + u_{yy}(x_i, y_j, t) + h(x_i, y_j, t),$$

or

$$\frac{\mathrm{d}u(x_i, y_j, t)}{\mathrm{d}t} = v(x_i, y_j, t),$$
$$\frac{\mathrm{d}v(x_i, y_j, t)}{\mathrm{d}t} = -2\alpha v(x_i, y_j, t) - \beta^2 u(x_i, y_j, t) + u_{xx}(x_i, y_j, t) + u_{yy}(x_i, y_j, t) + h(x_i, y_j, t).$$

or

$$\frac{(\mathrm{d}u_{(i,j)}(t))}{\mathrm{d}t} = v_{i,j}(t),$$
$$\frac{(\mathrm{d}v_{(i,j)}(t))}{\mathrm{d}t} = -2\alpha v_{i,j}(t) - \beta^2 u_{i,j}(t) + (u_{xx})_{i,j}(t) + (u_{yy})_{i,j}(t) + h_{i,j}(t).$$

The aforementioned system of equations is tackled using SSP-RK43 scheme.

4 Numerical experiments

Error formulae:

$$L_2 = \left(h\sum_{i=1}^{n}(u_{\text{exact}} - u_{\text{numerical}})^2\right)^{1/2},$$
$$L_\infty = \max[abs(u_{\text{exact}} - u_{\text{numerical}})],$$

Rel. error

$$= \left(\left(\sum_{i=1}^{n}(u_{\text{exact}} - u_{\text{numerical}})^2\right) \Big/ \left(\sum_{i=1}^{n}(u_{\text{exact}})^2\right)\right)^{1/2}.$$

4.1 Numerical examples regarding 1D HT equation

Example 1. Considered 1D hyperbolic telegraph equation (1) with $g(x, t) = (2 - 2\alpha + \beta^2)\exp(-t)\sin x$ in $[0, \pi]$ [31].

I.C.s: $u(x, 0) = 0$, $v(x, 0) = 0$.
B.C.s: $u(0, t) = 0$, $u(\pi, t) = 0$.
Exact solution: $u(x, t) = \exp(-t)\ \sin x$.

In Figure 1, exact and numerical $u(x, t)$ are matched at $t = 1, 2, 3, 4$ and 5 using Method I. In Figure 2, exact and numerical $u(x, t)$ are matched at $t = 1, 2, 3, 4$, and 5 using Method II. In Table 3, exact and numerical u are compared at different x values using Method I. In Table 4, L_2u and $L_\infty u$ are compared with [10] and [49] using Method I. In Table 5, exact and numerical u are matched at different x values using Method II. In Table 6, L_2u and $L_\infty u$ are matched with [10] and [49] using Method II.

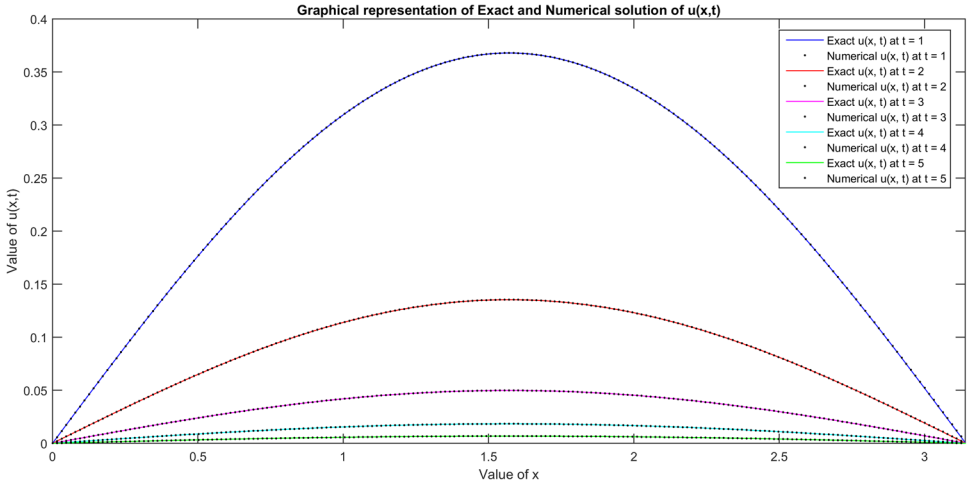

Figure 1: Graphical plot for $u(x, t)$ at $t = 1, 2, 3, 4$, and 5 for $N = 201$, $\alpha = 10$, $\beta = 5$, $\Delta t = 0.0001$, $\tau = 0.5$ using Method I regarding Example 1.

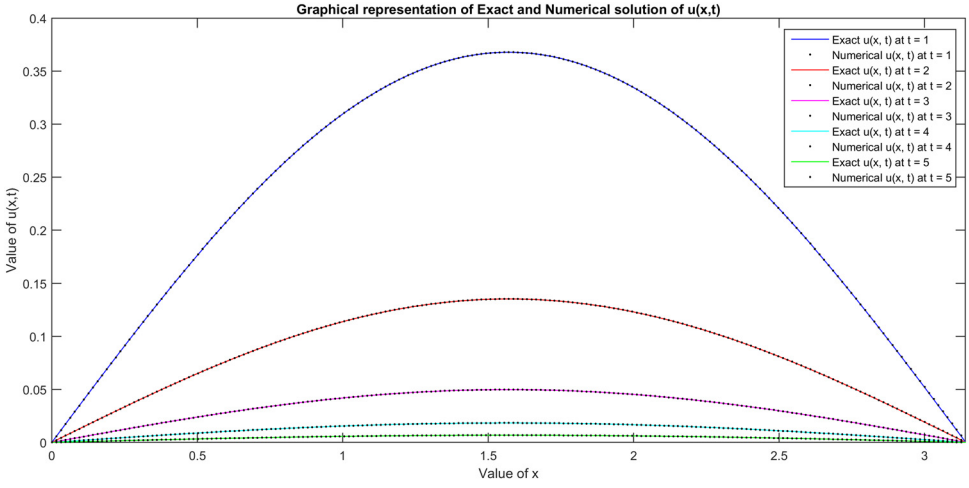

Figure 2: Graphical plot for $u(x, t)$ at $t = 1, 2, 3, 4$, and 5 for $N = 201$, $\alpha = 10$, $\beta = 5$, $\Delta t = 0.0001$, $\tau = 0.5$ using Method II regarding Example 1.

By Table 4, it can be claimed that Method I has produced the better results than [49] at some points, and the present results are better on the changed time levels. From

Table 3: Comparison of exact and numerical results of u at $t = 1$, 2, and 3 for $N = 101$, $\alpha = 10$, $\beta = 5$, $\Delta t = 0.0001$, $\tau = 0.5$ using Method I regarding Example 1

x	$t = 1$		$t = 2$		$t = 3$	
	Exact u	**Num. u**	**Exact u**	**Num. u**	**Exact u**	**Num. u**
0.2827	0.1026	0.1026	0.0378	0.0378	0.0139	0.0139
0.3142	0.1137	0.1137	0.0418	0.0418	0.0154	0.0154
0.3456	0.1246	0.1246	0.0458	0.0458	0.0169	0.0169
0.377	0.1354	0.1354	0.0498	0.0498	0.0183	0.0183
0.4084	0.1461	0.1461	0.0537	0.0537	0.0198	0.0198

Table 4: Comparison of error at different time levels with $N = 701$, $\alpha = 10$, $\beta = 15$, $\Delta t = 0.0001$ using Method I regarding Example 1

t	L_2 [10]	L_2 [49]	L_2u **Method I**
0.5	2.33×10^{-6}	7.95×10^{-5}	7.18×10^{-5}
1	4.37×10^{-6}	1.46×10^{-4}	4.36×10^{-5}
1.5	4.78×10^{-6}	1.59×10^{-4}	2.65×10^{-5}
2	4.27×10^{-6}	1.42×10^{-4}	1.60×10^{-5}
t	L_∞ [10]	L_∞ [49]	$L_\infty u$ **Method II**
0.5	1.86×10^{-6}	8.37×10^{-6}	3.42×10^{-4}
1	3.48×10^{-6}	1.57×10^{-5}	2.07×10^{-4}
1.5	3.83×10^{-6}	1.74×10^{-5}	1.26×10^{-4}
2	3.41×10^{-6}	1.58×10^{-5}	7.63×10^{-5}

Table 5: Comparison of exact and numerical results of u at $t = 1$, 2, and 3 for $N = 101$, $\alpha = 10$, $\beta = 5$, $\Delta t = 0.0001$, $\tau = 0.5$ using Method II regarding Example 1

x	$t = 1$		$t = 2$		$t = 3$	
	Exact u	**Num. u**	**Exact u**	**Num. u**	**Exact u**	**Num. u**
2.20×10^{-1}	8.03×10^{-2}	8.03×10^{-2}	2.95×10^{-2}	2.95×10^{-2}	1.09×10^{-2}	1.09×10^{-2}
2.51×10^{-1}	9.15×10^{-2}	9.15×10^{-2}	3.37×10^{-2}	3.37×10^{-2}	1.24×10^{-2}	1.24×10^{-2}
2.83×10^{-1}	1.03×10^{-1}	1.03×10^{-1}	3.78×10^{-2}	3.78×10^{-2}	1.39×10^{-2}	1.39×10^{-2}
3.14×10^{-1}	1.14×10^{-1}	1.14×10^{-1}	4.18×10^{-2}	4.18×10^{-2}	1.54×10^{-2}	1.54×10^{-2}
3.46×10^{-1}	1.25×10^{-1}	1.25×10^{-1}	4.58×10^{-2}	4.58×10^{-2}	1.69×10^{-2}	1.69×10^{-2}

Table 6, it is noticed that Method II is improvised method on the changed time level.

Example 2. Considered 1D hyperbolic telegraph equation (1) with $g(x, t) = (2 - 2t + t^2)(x - x^2)\exp(-t) + 2t^2\exp(-t)$ in [0,1] [31].

I.C.s: $u(x, 0) = 0$ and $v(x, 0) = 0$.
B.C.s: $u(0, t) = 0$ and $u(1, t) = 0$.
Exact solution: $u(x, t) = (x - x^2)\ t^2\ \exp(-t)$.

In Figure 3, $u(x, t)$ are matched for exact and numerical approximations at $t = 0.001$, 0.002, and 0.003 using Method I. In Figure 4, exact and numerical profiles of $u(x, t)$ are compared at $t = 0.001$, 0.002, and 0.003 using Method II. In Table 7, exact and numerical u are matched at different x and t values using Method I. In Table 8, exact and numerical u are compared at different x and t values using Method II.

Example 3. Considered 1D hyperbolic telegraph equation (1) with $g(x, t) = (3 - 4\alpha + \beta^2)\exp(-2t)\sinh(x)$ in [0,1] [18].

Table 6: Comparison of error at different time levels with $N = 701$, $\alpha = 10$, $\beta = 15$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 1

t	L_2 [10]	L_2 [49]	L_2u Method II
0.5	2.33×10^{-6}	7.95×10^{-5}	7.18×10^{-5}
1	4.37×10^{-6}	1.46×10^{-4}	4.36×10^{-5}
1.5	4.78×10^{-6}	1.59×10^{-4}	2.65×10^{-5}
2	4.27×10^{-6}	1.42×10^{-4}	1.60×10^{-5}
t	L_∞ [10]	L_∞ [49]	$L_\infty u$ Method II
0.5	1.86×10^{-6}	8.37×10^{-6}	3.42×10^{-4}
1	3.48×10^{-6}	1.57×10^{-5}	2.07×10^{-4}
1.5	3.83×10^{-6}	1.74×10^{-5}	1.26×10^{-4}
2	3.41×10^{-6}	1.58×10^{-5}	7.63×10^{-5}

I.C.s: $u(x, 0) = \sinh(x)$ and $u(x, 0) = -2 \quad \sinh(x)$.
B.C.: $u(0, t) = 0$ and $u(1, t) = \exp(-2t) \quad \sinh(1)$.
Exact solution: $u(x, t) = \exp(-2t) \quad \sinh(x)$.

In Figure 5, $u(x, t)$ are compared at $t = 1.0$, 1.5, and 2.0 using the Method I. In Figure 6, $u(x, t)$ are compared at $t = 1.0$, 1.5, and 2.0 using the Method II. In Table 9, $L_\infty u$ are compared with [18] using Methods I and II. At some points, the present results are better than [18]. In Table 10, $L_\infty u$ are compared with [18] at $t = 0.5$, 1.0, 1.5, and 2.0 using Method I and Method II. It is noticed that the present results are better at $t = 1.5$.

Example 4. Considered 1D hyperbolic telegraph equation with $g(x, t) = -2\alpha\sin t\sin x + \beta^2\cos t\sin x$ in [0,1] [18].

I.C.s: $u(x, 0) = \sin x$ and $v(x, 0) = 0$.
B.C.s: $u(0, t) = 0$ and $u(1, t) = \cos t \quad \sin(1)$.
Exact solution: $u(x, t) = \cos t \quad \sin x$.

In Figure 7, $u(x, t)$ are compared for the exact and numerical profiles at $t = 1$, 2, 3, and 4 using Method I. In Figure 8, $u(x, t)$ are matched for exact and numerical profiles at $t = 1$, 2, 3, 4, and 5 using Method II. In Table 11, $L_\infty u$ are compared with [18] using Methods I and II. The present results are better at some points than [18]. In Table 12, $L_\infty u$ are matched with [18] using the Methods I and II. It is noticed that at $t = 1.0$ obtained results are better.

4.2 Numerical examples regarding HT 2D

Example 5. Considered 2D hyperbolic telegraph equation (6) with $h(x, y, t) = 2(\cos t - \sin t)\sin x\sin y$ in $[0, 1] \times [0, 1]$ [32].

I.C.s: $u(x, y, 0) = \sin x\ \sin y$ and $v(x, y, 0) = 0$.
B.C.s: $u(0, y, t) = 0, \quad u(1, y, t) = \cos t\sin(1)\sin y$, $u(x, 0, t) = 0$, $u(x, 1, t) = \cos t\ \sin x\ \sin(1)$.

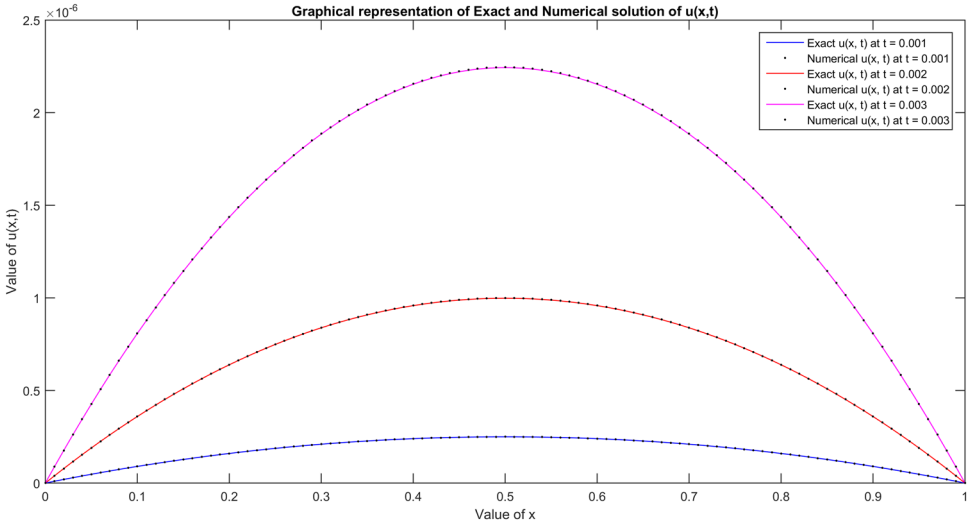

Figure 3: Comparison of results of $u(x, t)$ at $t = 0.001$, 0.002, and 0.003 with $N = 101$, $\alpha = 0.008$, $\beta = 0.008$, $\Delta t = 0.0001$, $\tau = 0.5$ using Method I regarding Example 2.

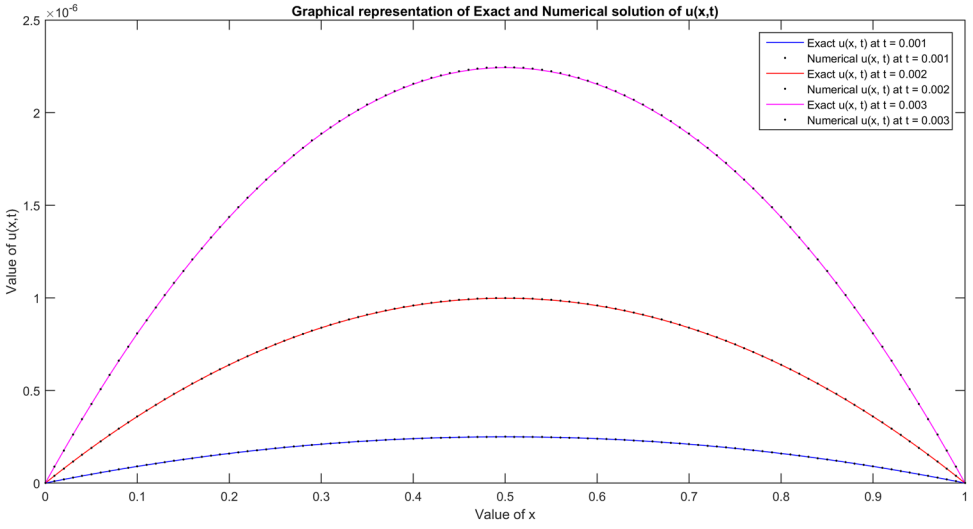

Figure 4: Comparison of results of $u(x, t)$ at $t = 0.001$, 0.002, and 0.003 with $N = 101$, $\alpha = 0.008$, $\beta = 0.008$, $\Delta t = 0.0001$, $\tau = 0.5$ using Method II regarding Example 2.

Table 7: Comparison of results at different node points and time levels regarding $u(x, t)$ for $N = 101$, $\alpha = 0.005$, $\beta = 0.005$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method I regarding Example 2

	$t = 0.1$		$t = 0.2$		$t = 0.3$	
x	Exact u	Num. u	Exact u	Num. u	Exact u	Num. u
1.00×10^{-2}	8.96×10^{-5}	9.33×10^{-5}	3.24×10^{-4}	3.60×10^{-4}	6.60×10^{-4}	8.01×10^{-4}
2.00×10^{-2}	1.77×10^{-4}	1.85×10^{-4}	6.42×10^{-4}	7.11×10^{-4}	1.31×10^{-3}	1.58×10^{-3}
3.00×10^{-2}	2.63×10^{-4}	2.74×10^{-4}	9.53×10^{-4}	1.05×10^{-3}	1.94×10^{-3}	2.33×10^{-3}

Table 8: Comparison of results at different node points and time levels regarding $u(x, t)$ for $N = 101$, $\alpha = 0.005$, $\beta = 0.005$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 2

	$t = 0.1$		$t = 0.2$		$t = 0.3$	
x	Exact u	Num. u	Exact u	Num. u	Exact u	Num. u
1.00×10^{-2}	8.96×10^{-5}	9.33×10^{-5}	3.24×10^{-4}	3.60×10^{-4}	6.60×10^{-4}	8.01×10^{-4}
2.00×10^{-2}	1.77×10^{-4}	1.85×10^{-4}	6.42×10^{-4}	7.11×10^{-4}	1.31×10^{-3}	1.58×10^{-3}
3.00×10^{-2}	2.63×10^{-4}	2.74×10^{-4}	9.53×10^{-4}	1.05×10^{-3}	1.94×10^{-3}	2.33×10^{-3}

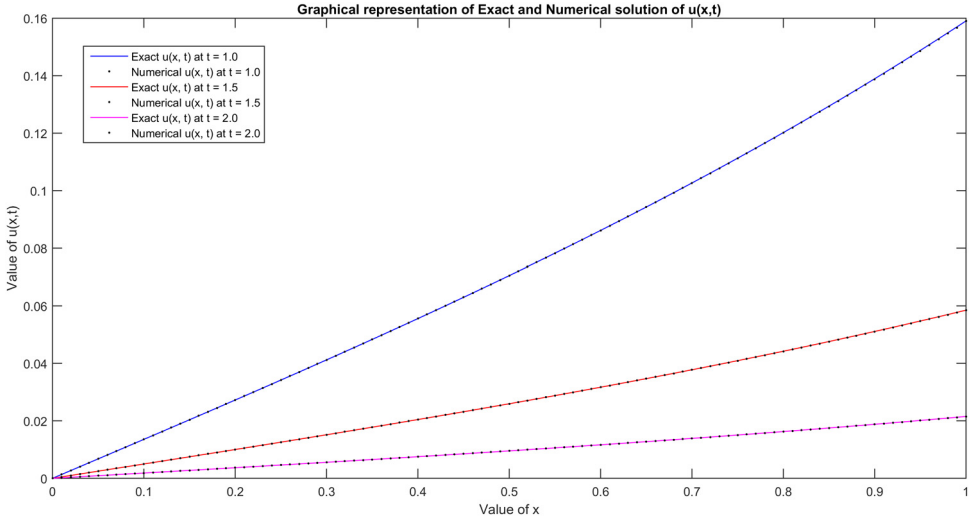

Figure 5: Comparison of $u(x, t)$ at $t = 1.0$, 1.5, and 2.0 for $N = 101$, $\alpha = 30$, $\beta = 10$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method I regarding Example 3.

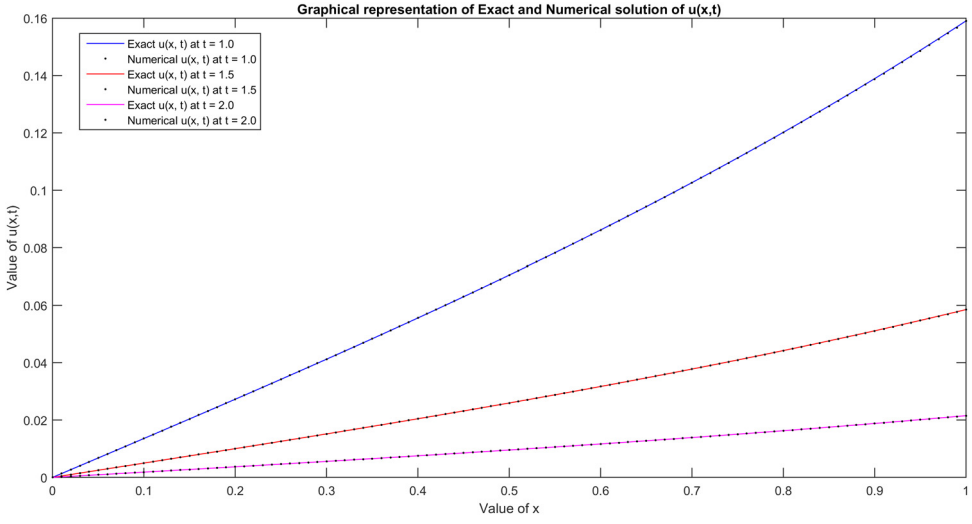

Figure 6: Comparison of $u(x, t)$ at $t = 1.0$, 1.5, and 2.0 for $N = 101$, $\alpha = 30$, $\beta = 10$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 3.

Table 9: Comparison of maximum absolute error at different time levels for $\alpha = 20$, $\beta = 10$, $N = 101$, and $\tau = 0.5$ regarding Example 3

t	L_∞ [18]	$L_\infty u$ [present] Method I	$L_\infty u$ [present] Method II
0.5	2.24×10^{-5}	8.66×10^{-4}	8.66×10^{-4}
1	3.57×10^{-4}	3.18×10^{-4}	3.18×10^{-4}
1.5	1.95×10^{-2}	1.17×10^{-4}	1.17×10^{-4}
2	—	4.31×10^{-5}	4.31×10^{-5}

Table 10: Comparison of maximum absolute error at different time levels for $\alpha = 20$, $\beta = 10$, $N = 101$, $\tau = 0.5$, and $\Delta t = 0.0001$ regarding Example 3

t	L_∞ [18]	$L_\infty u$ [present] Method I	$L_\infty u$ [present] Method II
0.5	2.24×10^{-6}	6.53×10^{-4}	6.53×10^{-4}
1	1.73×10^{-5}	2.42×10^{-4}	2.42×10^{-4}
1.5	4.15×10^{-3}	8.94×10^{-5}	8.94×10^{-5}
2	—	3.29×10^{-5}	3.29×10^{-5}

Exact solution: $u(x, y, t) = \cos t \sin x \sin y$, and $v(x, y, t) = -\sin t \sin x \sin y$.

In Figure 9, surface and contour plots of u are provided at $t = 1$ using Method I. In Figure 10, surface and contour plots of u are provided at $t = 1$ using Method II. In Table 13, L_2u and $L_\infty u$ are compared with [8] using Methods I and II. The present results are much better than [8]. In Table 14, L_2u and $L_\infty u$ are matched with [32] using Methods I and II. In Table 15, relative errors of u are matched with [8] and [32] using Methods I and II. Present results are much better than compared ones.

Example 6. Considered 2D hyperbolic telegraph equation (6) with $h(x, y, t) = (-2\alpha + \beta^2 - 1)\exp(-t)\sinh(x)\sinh(y)$ in $[0, 1] \times [0, 1]$ [32].

I.C.s: $u(x, y, 0) = \sinh(x)\ \sinh(y)$ and $v(x, y, 0) = -\sinh(x)\ \sinh(y)$.

B.C.s: $u(0, y, t) = 0$, $u(1, y, t) = \exp(-t)\ \sinh(1)\ \sinh(y)$,

$u(x, 0, t) = 0$, $u(x, 1, t) = \exp(-t)\ \sinh(x)\ \sinh(1)$.

Exact solution: $u(x, y, t) = \exp(-t)\ \sinh(x)\ \sinh(y)$ and $v(x, y, t) = -\exp(t)\ \sinh(x)\ \sinh(y)$.

In Figure 11, comparisons of surface and contour plots for u are given using Method I. In Figure 12, comparison of surface and contour plots is provided for u using Method II. In Table 16, L_2u and $L_\infty u$ are matched with [8] using Methods I and II. At some points, the present results are better than compared. In Table 17, L_2u and $L_\infty u$ are compared with [8] using Methods I and II. Present results are better than matched results. In Table 18, relative errors of u are better obtained than [8] and [18] using Methods I and II.

Example 7. Considered 2D hyperbolic telegraph equation (6) with $h(x, y, t) = (-3\cos t - 2\alpha \sin t + \beta^2 \cos t)\sinh(x)\ \sinh(y)$ in $[0, 1] \times [0, 1]$ [32].

I.C.s: $u(x, y, 0) = 0$ and $v(x, y, 0) = 0$.

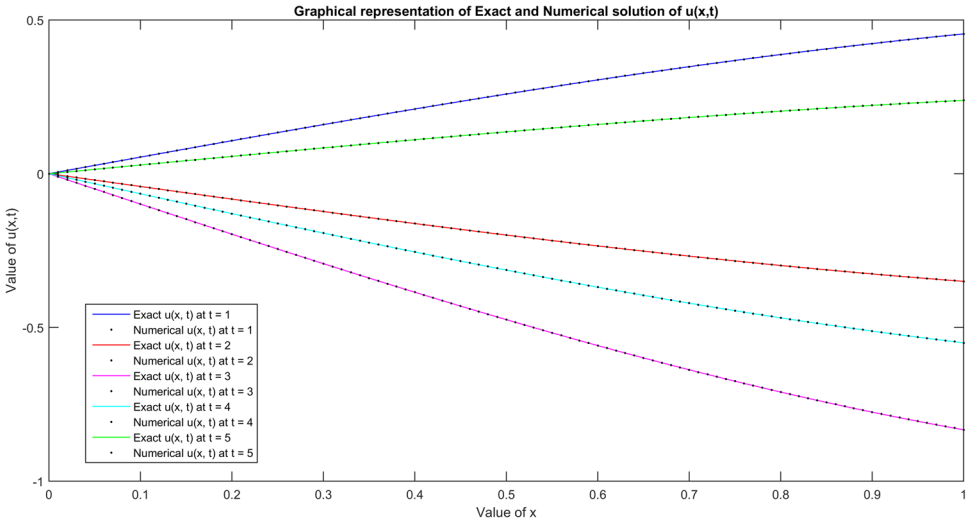

Figure 7: Comparison of $u(x, t)$ at $t = 1, 2, 3, 4$, and 5 for $N = 101$, $\alpha = 30$, $\beta = 5$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method I regarding Example 4.

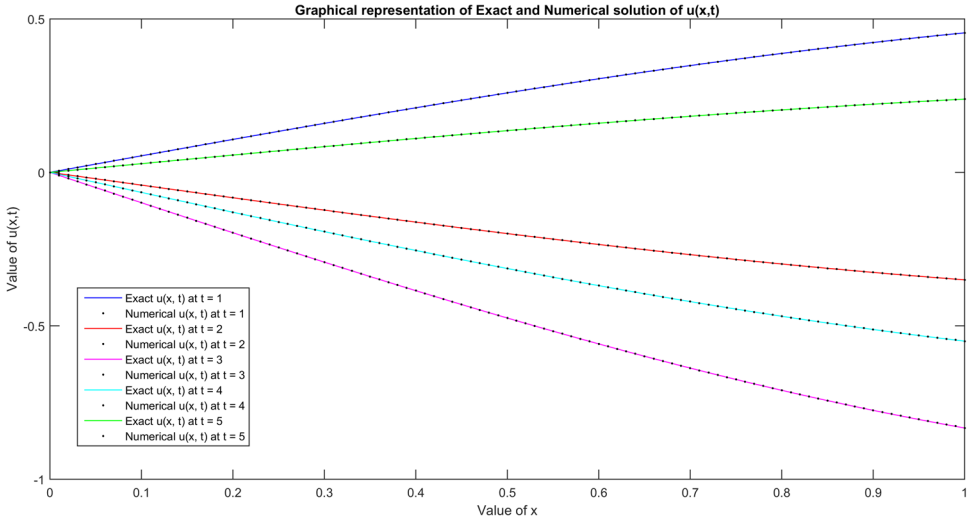

Figure 8: Comparison of $u(x, t)$ at $t = 1, 2, 3, 4$, and 5 for $N = 101$, $\alpha = 30$, $\beta = 5$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 4.

Table 11: Comparison of maximum absolute error at different time levels for $\alpha = 10$, $\beta = 5$, $N = 101$, $\tau = 0.5$, and $\Delta t = 0.001$ regarding Example 4

t	L_∞ [18]	$L_\infty u$ [present] Method I	$L_\infty u$ [present] Method II
0.2	1.67×10^{-5}	5.70×10^{-4}	5.70×10^{-4}
0.5	4.71×10^{-4}	4.03×10^{-4}	4.03×10^{-4}
1	8.63×10^{-4}	7.08×10^{-4}	7.08×10^{-4}
1.5	—	8.39×10^{-4}	8.39×10^{-4}

Table 12: Comparison of maximum absolute error at different time levels for $\alpha = 10$, $\beta = 5$, $N = 101$, $\tau = 0.5$, and $\Delta t = 0.0001$ regarding Example 4

t	L_∞ [18]	$L_\infty u$ [present] Method I	$L_\infty u$ [present] Method II
0.2	1.67×10^{-6}	6.65×10^{-4}	6.65×10^{-4}
0.5	4.73×10^{-5}	5.89×10^{-4}	5.89×10^{-4}
1	7.09×10^{-4}	3.41×10^{-4}	3.41×10^{-4}
1.5	—	8.39×10^{-5}	8.39×10^{-5}

B.C.s: $u(0, y, t) = 0$, $u(1, y, t) = \cos t \sinh(1) \sinh y$, $u(x, 0, t) = 0$, $u(x, 1, t) = \cos t \sinh x \sinh(1)$.

Exact solution: $u(x, y, t) = \cos t \sinh x \sinh y$ and $v(x, y, t) = -\sin t \sinh x \sinh y$.

In Figure 13, surface and contour plots for u are matched using Method I at $t = 1$. In Figure 14, surface and contour plots for u are matched using Method II at $t = 1$. In Table 19, L_2u and $L_\infty u$ are compared with [8] using Methods I and II. Methods I and II have generated better results at some points. In Table 20, L_2u and $L_\infty u$ are matched with [8] using Methods I and II. Methods I and II have produced the acceptable results in most of the cases. In Table 21, relative errors of u are better matched with [8] and [18] using Methods I and II. In Table 22, relative errors of u are compared with [8] and [18] using Methods I and II.

Example 8. Considered 2D hyperbolic telegraph equation (6) with $h(x, y, t) = -2\exp(x + y - t)$ in $[0, 1] \times [0, 1]$ [32].

I.C.s: $u(x, y, 0) = \exp(x + y)$ and $v(x, y, 0) = -\exp(x + y)$.

B.C.s: $u(0, y, t) = \exp(y - t)$, $u(1, y, t) = \exp(1 + y - t)$, $u(x, 0, t) = \exp(x - t)$, $u(x, 1, t) = \exp(1 + x - t)$.

Exact solution: $u(x, y, t) = \exp(x + y - t)$ and $v(x, y, t) = \exp(x + y - t)$.

In Figure 15, surface and contour plots for u are provided at $t = 1$ using Method I. In Figure 16, surface and contour plots are given at $t = 1$ using Method II. In Table 23, L_2u and $L_\infty u$ are compared with [8] using Methods I and II. The present results are highly acceptable.

Example 9. Considered 2D hyperbolic telegraph equation (6) with $h(x, y, t) = 2\pi^2 \exp(-t)\sin(\pi x)\sin(\pi y)$ in $[0, 1] \times [0, 1]$ [32].

I.C.s: $u(x, y, 0) = \sin(\pi x) \sin(\pi y)$ and $v(x, y, 0) = -\sin(\pi x) \sin(\pi y)$.

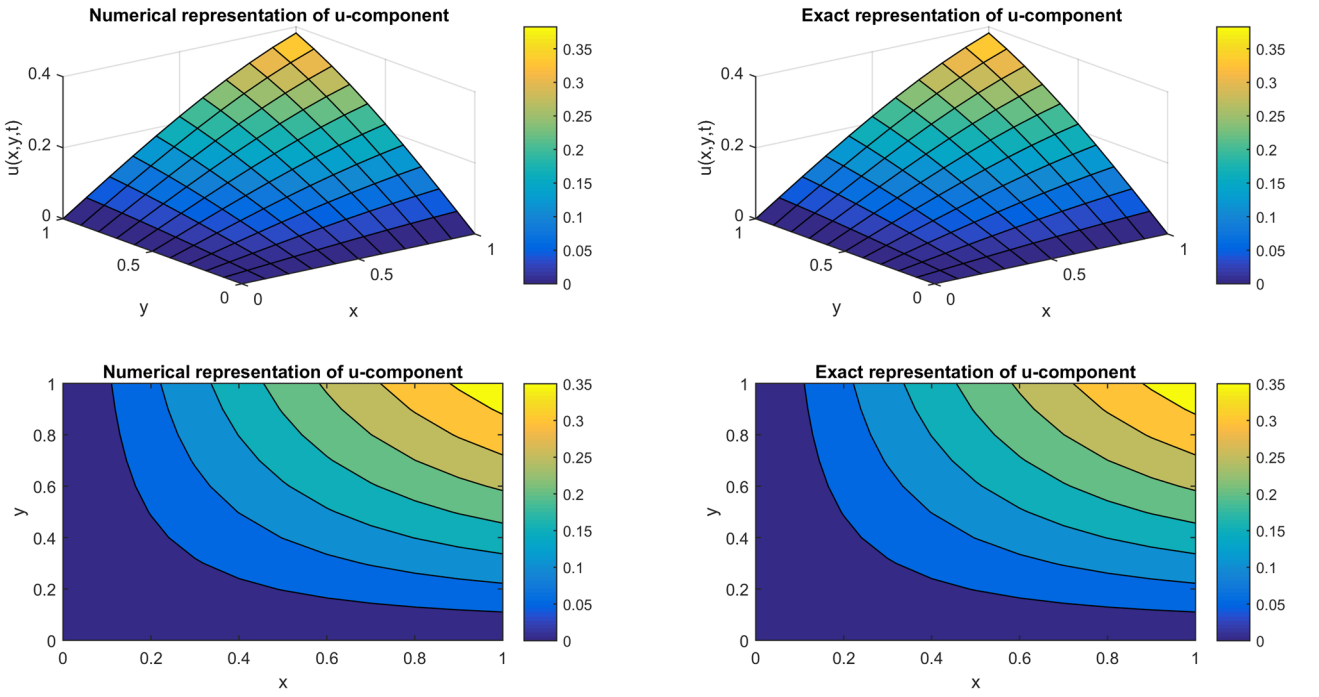

Figure 9: Surface and contour representation of $u(x, t)$ at $t = 1$ with $N = 11$, $\alpha = 1$, $\beta = 1$, $\tau = 0.5$, and $\Delta t = 0.0001$ using Method I regarding Example 5.

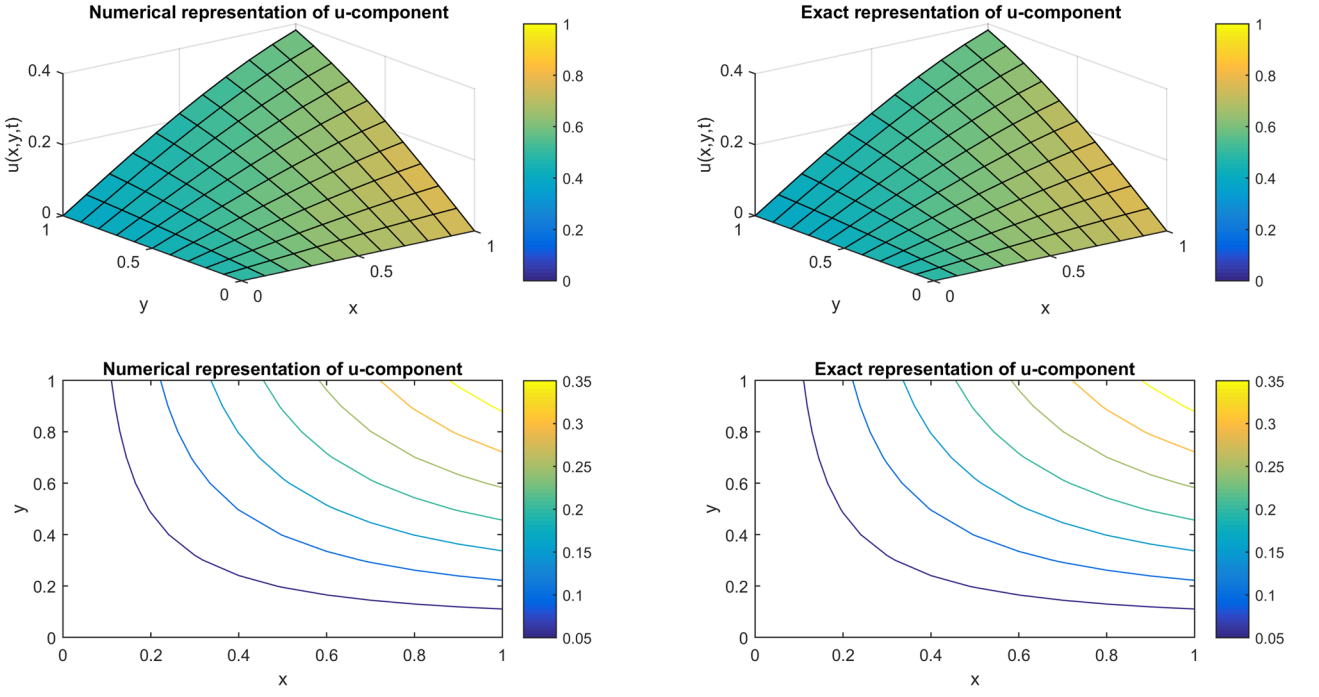

Figure 10: Surface and contour representation of $u(x, t)$ at $t = 1$ with $N = 11$, $\alpha = 1$, $\beta = 1$, $\tau = 0.5$, and $\Delta t = 0.0001$ using Method II regarding Example 5.

Table 13: Comparison of errors at $t = 1$, 2, and 3 for $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 5

t	[8]		Method I	
	L_2	L_∞	L_2u	$L_\infty u$
1	9.97×10^{-4}	2.27×10^{-3}	8.17×10^{-5}	4.75×10^{-5}
2	1.09×10^{-3}	2.87×10^{-3}	1.31×10^{-5}	6.75×10^{-6}
3	2.29×10^{-4}	6.08×10^{-4}	7.33×10^{-5}	5.04×10^{-5}
t	[8]		Method II	
	L_2	L_∞	L_2u	$L_\infty u$
1	9.97×10^{-4}	2.27×10^{-3}	8.17×10^{-5}	4.75×10^{-5}
2	1.09×10^{-3}	2.87×10^{-3}	1.31×10^{-5}	6.75×10^{-6}
3	2.29×10^{-4}	6.08×10^{-4}	7.33×10^{-5}	5.03×10^{-5}

Table 14: Comparison of errors at $t = 1$, 2, and 3 for $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 5

	[32]		Method I	
t	L_2	L_∞	L_2u	$L_\infty u$
1	3.75×10^{-6}	4.57×10^{-6}	8.17×10^{-5}	4.75×10^{-5}
2	4.47×10^{-6}	5.61×10^{-6}	1.31×10^{-5}	6.75×10^{-6}
3	3.74×10^{-6}	6.28×10^{-6}	7.33×10^{-5}	5.04×10^{-5}
	[32]		Method II	
t	L_2	L_∞	L_2u	$L_\infty u$
1	3.75×10^{-6}	4.57×10^{-6}	8.17×10^{-5}	4.75×10^{-5}
2	4.47×10^{-6}	5.61×10^{-6}	1.31×10^{-5}	6.75×10^{-6}
3	3.74×10^{-6}	6.28×10^{-6}	7.33×10^{-5}	5.03×10^{-5}

Table 15: Comparison of errors with at $t = 1$, 2 and 3 for $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 5

t	Relative error [8]	Relative error [32]	Relative error u Method I	Relative error u Method II
1	5.98×10^{-3}	2.47×10^{-5}	1.24×10^{-4}	1.24×10^{-4}
2	8.50×10^{-3}	3.83×10^{-5}	1.94×10^{-5}	1.94×10^{-5}
3	7.47×10^{-4}	1.34×10^{-5}	6.31×10^{-5}	6.31×10^{-5}

Boundary conditions are obtained from the following exact solution.

Exact solution: $u(x, y, t) = \exp(-t) \sin(\pi x) \sin(\pi y)$ and $v(x, y, t) = -\exp(-t) \sin(\pi x) \sin(\pi y)$.

In Figure 17, surface and contour plots for u are matched at $t = 1$ using Method I. In Figure 18, surface and contour plots for u are compared at $t = 1$ using Method II. In Table 24, L_2u and $L_\infty u$ are compared with [8] using Methods I and II. Methods I and II have generated the better results.

Example 10. The present example considered 2D hyperbolic telegraph equation is as follows [18]: $u_{tt} + 4\pi u_t + 2\pi^2 u = u_{xx} + u_{yy} + 2\pi t \sin\pi(x + y)\exp(-(x + y)t) + [(x + y - 2\pi)^2 - 2t^2]\sin(\pi x)\sin(\pi y)\exp(-(x + y)t)$.

I.C.s: $u(x, y, 0) = \sin(\pi x) \sin(\pi y)$ $v(x, y, 0) = (x + y) \sin(\pi x) \sin(\pi y)$.

Boundary conditions can be extracted from the provided exact solution.

Exact solution: $u(x, y, t) = \exp(-(x + y)t) \sin(\pi x) \sin(\pi y)$.

$u(x, y, t) = -(x + y) \exp(-(x + y)t)\sin(\pi x)\sin(\pi y)$.

Domain of computation: $[0, 1] \times [0, 1]$.

In Figures 19 and 20, surface and contour plots are provided for u and $t = 1$ using Methods I and II.

Need for SSPRK 43 method:

The SSP method [48,50,51] is also termed as TVD (time variation diminishing) implemented for tackling the ODE system. The mth-order RK method is defined as follows; $u^{(0)} = u^n$

$u^{(i)} = \sum_{\theta=1}^{i-1}[\alpha_{i,\theta}u^\theta + \Delta t\beta_{i,\theta}u^\theta]$,

$\alpha_{i,\theta} \geq 0$, $i = 1, 2, 3, \ldots$ $u^{(n+1)} = u^{(m)}$.

For the consistent system $\sum_{\theta=0}^{i-1}\alpha_{i,\theta} = 1$, as per the hyperbolic conversation law:

$$\frac{\partial u}{\partial t} = -g(u)_x,$$

where $g(u)_x$ (spatial derivative) is discretized by TVD FD scheme, which lead to an ODE system.

$$\frac{\mathrm{d}u}{\mathrm{d}t} = L(u).$$

It is considered that there is discretization in space: $L(u)$ has the attribute that it is combined with the forward, first-order Euler time discretization. The CFL condition is provided as: $\Delta t \leq c\Delta t_F E$

Total variation for the multistep time discretization or SSP RK (higher-order) is to obtain the temporal accuracy

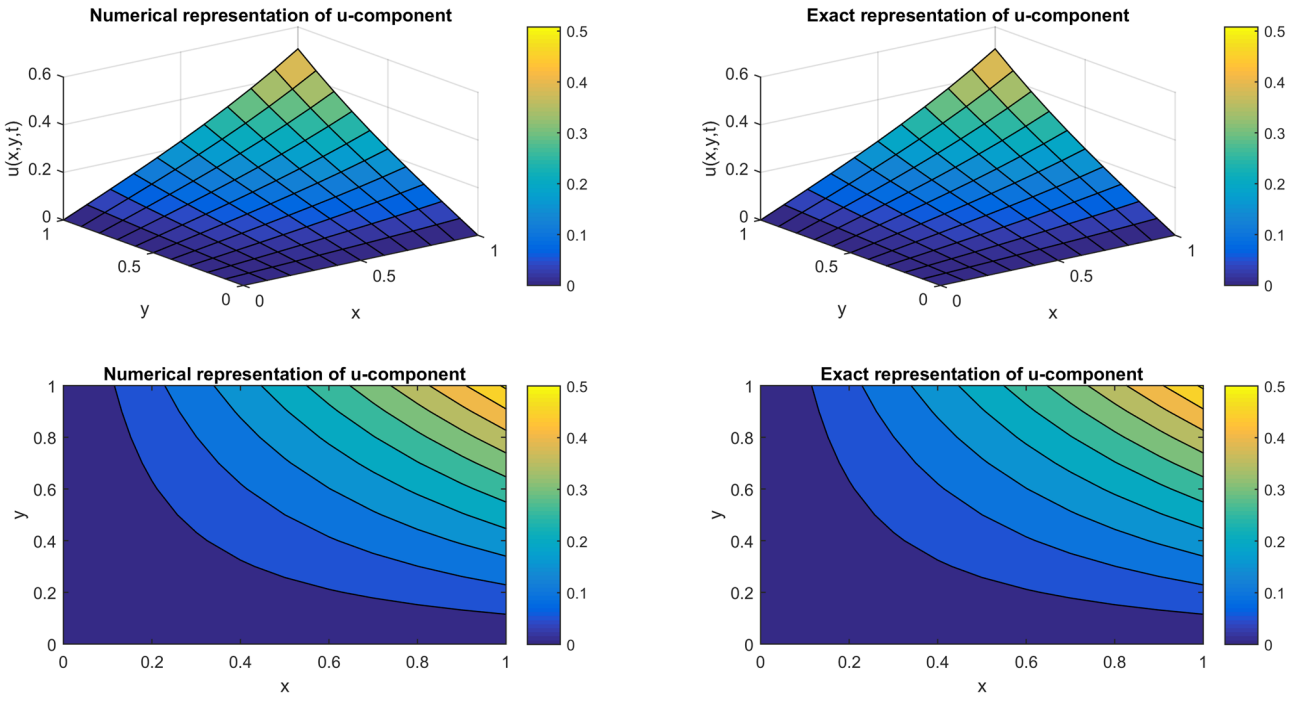

Figure 11: Surface and contour plots for u component at $t = 1$ for $N = 11$, $\alpha = 10$, $\beta = 5$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method I regarding Example 6.

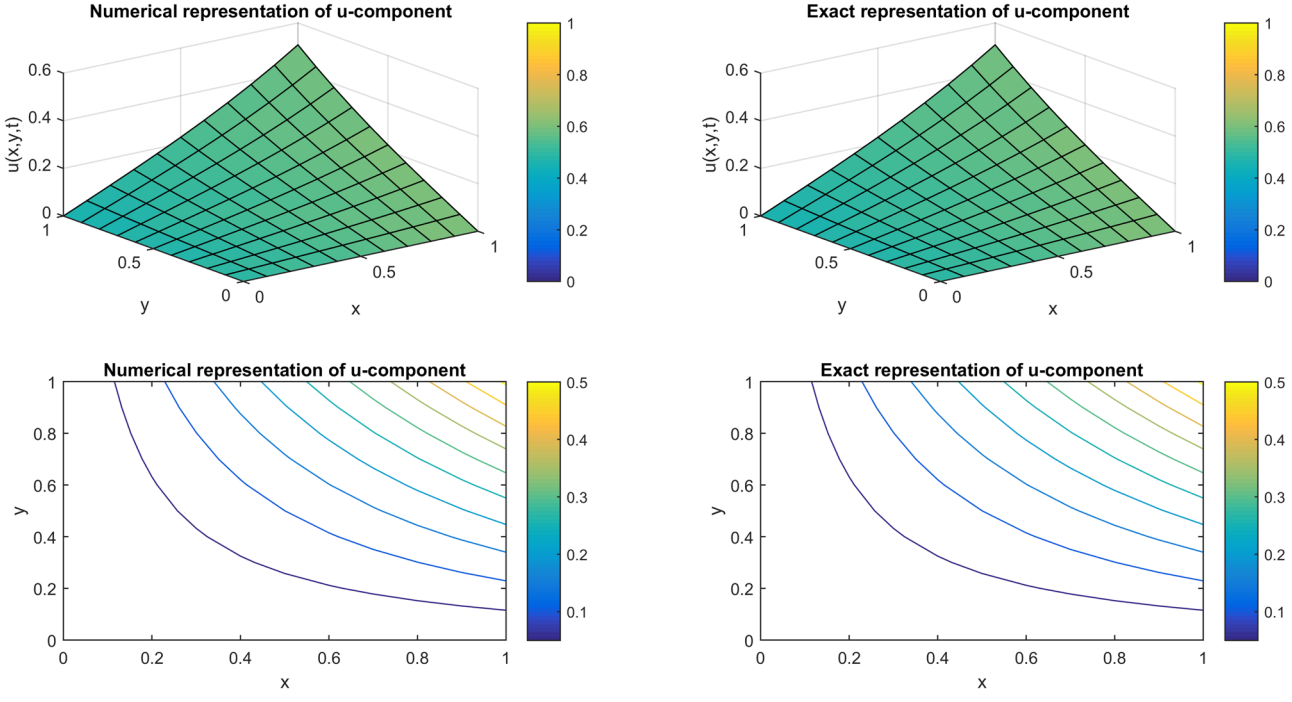

Figure 12: Surface and contour plots for u component at $t = 1$ for $N = 11$, $\alpha = 10$, $\beta = 5$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 6.

Table 16: Comparison of errors for $\alpha = 10$, $\beta = 0$, $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 6

	[8]		Method I	
t	L_2	L_∞	L_2u	$L_\infty u$
0.5	9.30×10^{-5}	4.23×10^{-4}	4.27×10^{-5}	4.27×10^{-5}
1	6.37×10^{-5}	2.58×10^{-4}	3.11×10^{-5}	2.84×10^{-5}
2	2.55×10^{-5}	9.58×10^{-5}	2.22×10^{-5}	1.26×10^{-5}
	[8]		Method II	
t	L_2	L_∞	L_2u	$L_\infty u$
0.5	9.30×10^{-5}	4.23×10^{-4}	4.27×10^{-5}	4.04×10^{-5}
1	6.37×10^{-5}	2.58×10^{-4}	3.11×10^{-5}	2.84×10^{-5}
2	2.55×10^{-5}	9.58×10^{-5}	2.22×10^{-5}	1.26×10^{-5}

Table 17: Comparison of errors for $\alpha = 10$, $\beta = 5$, $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 6

	[8]		Method I	
t	L_2	L_∞	L_2u	$L_\infty u$
0.5	1.61×10^{-4}	2.47×10^{-4}	3.24×10^{-5}	3.58×10^{-5}
1	1.53×10^{-5}	3.31×10^{-4}	1.88×10^{-5}	2.21×10^{-5}
2	4.65×10^{-5}	1.14×10^{-5}	8.05×10^{-6}	8.18×10^{-6}
	[8]		Method II	
t	L_2	L_∞	L_2u	$L_\infty u$
0.5	1.61×10^{-4}	2.47×10^{-4}	3.24×10^{-5}	3.58×10^{-5}
1	1.53×10^{-5}	3.31×10^{-4}	1.88×10^{-5}	2.21×10^{-5}
2	4.65×10^{-5}	1.14×10^{-5}	8.05×10^{-6}	8.18×10^{-6}

Table 18: Comparison of relative errors for $\alpha = 10$, $\beta = 5$, $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 6

t	Relative error [8]	Relative error [18]	Relative error u Method 1	Relative error u Method II
0.5	1.11×10^{-4}	1.12×10^{-4}	2.24×10^{-5}	2.24×10^{-5}
1	1.33×10^{-4}	1.81×10^{-4}	2.74×10^{-5}	2.74×10^{-5}
2	3.20×10^{-4}	4.73×10^{-4}	3.46×10^{-5}	3.45×10^{-5}

with the higher-order to fetch the substantial stability property.

5 Stability and convergence analysis

5.1 Stability analysis

In the traditional regime, the Von Neumann stability approach can be easily implemented. DQM discretized models can not be tackled using the Von Neumann stability analysis method. To solve this purpose, the matrix stability analysis method is used extensively [52,53]. In this piece of research, stability is discussed using the matrix stability analysis method. The time-dependent system is as follows: $\frac{\partial g}{\partial t} = L(g)$. Along with the I.C.s, L is the spatial nonlinear differential operator. After the discretization with UAT DQM or UAH DQM, and the linearization of the nonlinear term, the following ODE system will be fetched:

$\frac{\mathrm{d}}{\mathrm{d}t}u = [D]\{u\} + \{c\}$.

$\{u\} \rightarrow$ unknown vector of the functional value.

$\{c\} \rightarrow$ nonhomogeneous part.

$[D] \rightarrow$ coefficient matrix.

In the more compact form,

$$\frac{\mathrm{d}u}{\mathrm{d}t} = Du + H. \tag{31}$$

Or

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -2\alpha & A \\ I & O \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} O_1 \\ H_i(t) \end{pmatrix}.$$

O is the null matrix. I is the unit matrix.

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix},$$

is a solution vector. H is the vector for the nonhomogeneous part, O_1 is the zero vector, and $H_i(t)$ is the column vector. The stability of the fetched system (31) is directly connected to the stability of the numerical regime, which is proposed to solve it. If the obtained ODE system is unstable, then any stable numerical regime can not converge. The stability of the ODE system depends upon the eigenvalues of the coefficient matrix D. If all the eigenvalues of D consist of the real part, then the ODE system will be stable. $\lambda \rightarrow$ eigenvalue of D. $u_1, u_2 \rightarrow$ the corresponding eigenvectors of $(N - 2)$ order each. Let

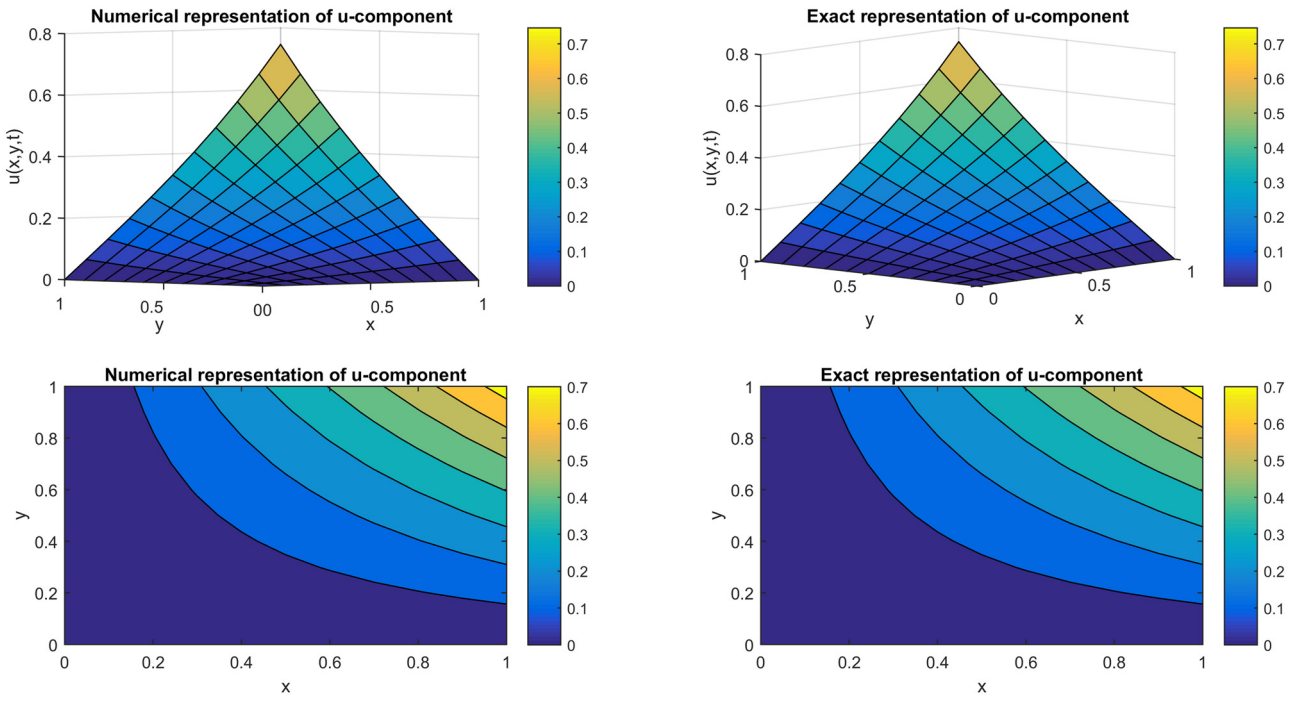

Figure 13: Surface and contour plots for u component at $t = 1$ for $N = 11$, $\alpha = 50$, $\beta = 5$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method I regarding Example 7.

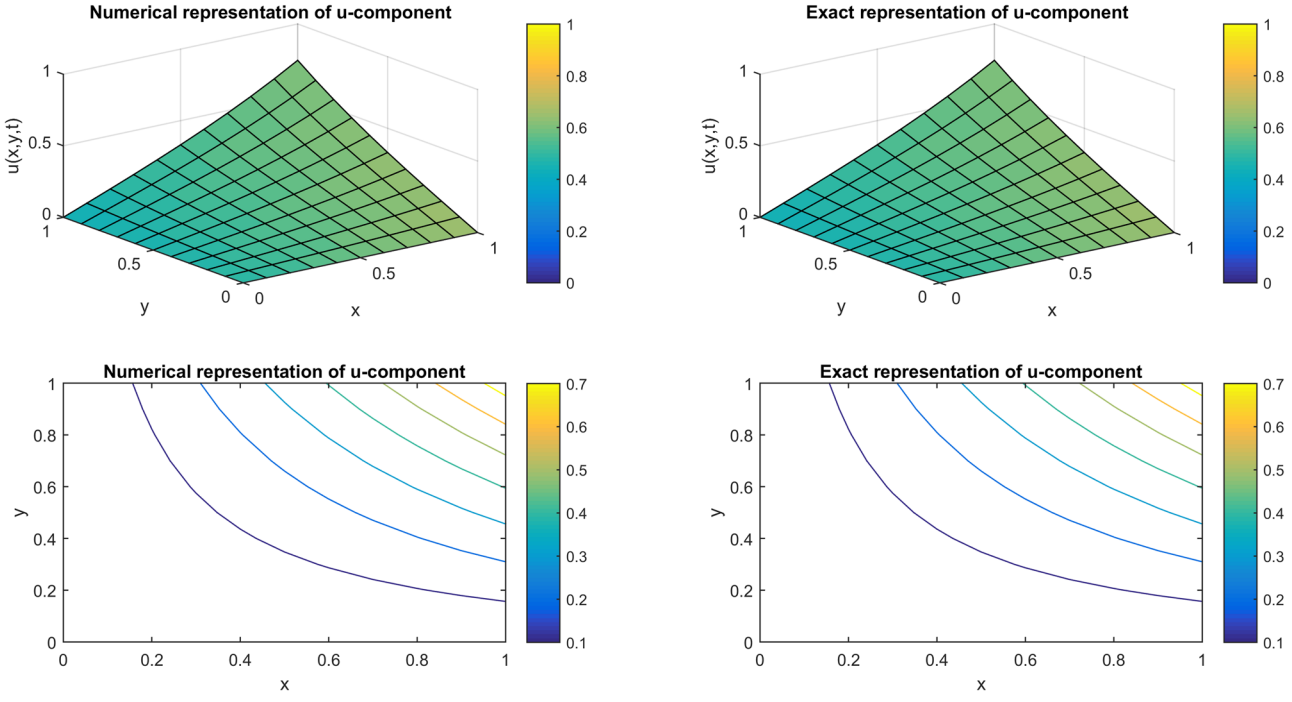

Figure 14: Surface and contour plots for u component at $t = 1$ for $N = 11$, $\alpha = 50$, $\beta = 5$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 7.

Table 19: Comparison of errors for $\alpha = 10$, $\beta = 5$, $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 7

	[8]		Method I	
t	L_2	L_∞	L_2u	$L_\infty u$
0.5	6.96×10^{-6}	3.76×10^{-4}	4.55×10^{-5}	6.57×10^{-5}
1	1.72×10^{-4}	5.64×10^{-4}	1.79×10^{-5}	2.02×10^{-5}
2	1.65×10^{-4}	5.13×10^{-4}	6.41×10^{-5}	7.38×10^{-5}
	[8]		Method II	
t	L_2	L_∞	L_2u	$L_\infty u$
0.5	6.96×10^{-6}	3.76×10^{-4}	4.54×10^{-5}	6.56×10^{-5}
1	1.72×10^{-4}	5.64×10^{-4}	1.79×10^{-5}	2.02×10^{-5}
2	1.65×10^{-4}	5.13×10^{-4}	6.41×10^{-5}	7.38×10^{-5}

Table 20: Comparison of errors for $\alpha = 50$, $\beta = 5$, $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 7

	[8]		Method I	
t	L_2	L_∞	L_2u	$L_\infty u$
0.5	9.88×10^{-5}	3.70×10^{-4}	3.12×10^{-5}	5.81×10^{-5}
1	1.68×10^{-4}	5.69×10^{-4}	3.05×10^{-5}	2.33×10^{-5}
2	1.71×10^{-4}	5.26×10^{-4}	4.62×10^{-5}	6.09×10^{-5}
	[8]		Method II	
t	L_2	L_∞	L_2u	$L_\infty u$
0.5	9.88×10^{-5}	3.70×10^{-4}	3.12×10^{-5}	5.81×10^{-5}
1	1.68×10^{-4}	5.69×10^{-4}	3.05×10^{-5}	2.33×10^{-5}
2	1.71×10^{-4}	5.26×10^{-4}	4.62×10^{-5}	6.09×10^{-5}

Table 21: Comparison of relative errors for $\alpha = 10$, $\beta = 5$, $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 7

t	Relative error [8]	Relative error [18]	Relative error u Method I	Relative error u Method II
0.5	1.38×10^{-5}	2.01×10^{-5}	2.84×10^{-5}	2.84×10^{-5}
1	3.60×10^{-5}	4.60×10^{-5}	1.43×10^{-5}	1.43×10^{-5}
2	4.47×10^{-5}	4.45×10^{-5}	8.94×10^{-5}	8.94×10^{-5}

$$\begin{pmatrix} -2\alpha & A \\ I & O \end{pmatrix}.$$

As per the definition of eigenvalues and eigenvectors:

$$\begin{pmatrix} -2\alpha & A \\ I & O \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \lambda\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

Table 22: Comparison of relative errors for $\alpha = 50$, $\beta = 5$, $N = 21$, $\Delta t = 0.0001$, and $\tau = 0.5$ regarding Example 7

t	Relative error [8]	Relative error [18]	Relative error u Method I	Relative error u Method II
0.5	1.27×10^{-5}	2.87×10^{-5}	1.55×10^{-5}	1.55×10^{-5}
1	3.51×10^{-5}	7.01×10^{-5}	2.56×10^{-5}	2.56×10^{-5}
2	4.64×10^{-5}	6.78×10^{-5}	6.29×10^{-5}	6.29×10^{-5}

$$-2\alpha u_1 + Au_2 = \lambda u_1(a),$$
$$Iu_1 = \lambda u_2(b).$$
$$-2\alpha[\lambda u_2] + Au_2 = \lambda[\lambda u_2].$$
$$Au_2 = (\lambda^2 + 2\alpha\lambda)u_2.$$
$$A \overset{\text{eigen value}}{\rightarrow} \lambda^2 + 2\alpha\lambda,$$

where u_2 is the corresponding eigenvector. Considered $\lambda^2 + 2\alpha\lambda = \mu$.

$$\lambda^2 + 2\alpha\lambda - \mu = 0.$$
$$\lambda = -\alpha \pm \sqrt{\alpha^2 + \mu}\,.$$

It is noticeable from the aforementioned equation that the real part of the eigenvalues of D is always negative. Therefore, the developed system of ODEs is stable. Hence, stability condition is verified.

Convergence analysis

Let $\mathrm{UATB}_k(x)/\mathrm{UAHB}_k(x) k = -1, \ldots, n+1$ UAT tension B-spline/UAH tension B-spline is the set of basis functions, continuously differentiable twice and L.I. in the provided universe if discourse. Cubic UAT tension B-spline/UAH tension B-spline is $(n+3)$ dimensional subspace of $C^2[a, b]$. By using the notion of $\mathrm{UATB}_k(x)/\mathrm{UAHB}_k(x)$:

$$u^{(p)}(x_i) = \sum_{j=1}^{n} w_{ij}^{(p)} u(x_j). \tag{32}$$

$$\mathrm{UATB}_k'(x_i)/\mathrm{UAHB}_k'(x_i) = \sum_{j=0}^{n} w_{ij}^{(1)} \mathrm{UATB}_k(x_j)/\mathrm{UAHB}_k(x_j), \tag{33}$$

$$\mathrm{UATB}_k''(x_i)/\mathrm{UAHB}_k''(x_i) = \sum_{j=0}^{n} w_{ij}^{(2)} \mathrm{UATB}_k(x_j)/\mathrm{UAHB}_k(x_j). \tag{34}$$

By solving the system, $w_{ij}^{(p)}$ can be obtained, and the derivative of the given function can be obtained at the node points. Cubic uniform algebraic hyperbolic tension B-spline $h(x)$ is given as follows:

$$h(x) = \sum_{k=-1}^{(n+1)} m_k \mathrm{UATB}_k(x)/\mathrm{UAHB}_k(x), \tag{35}$$

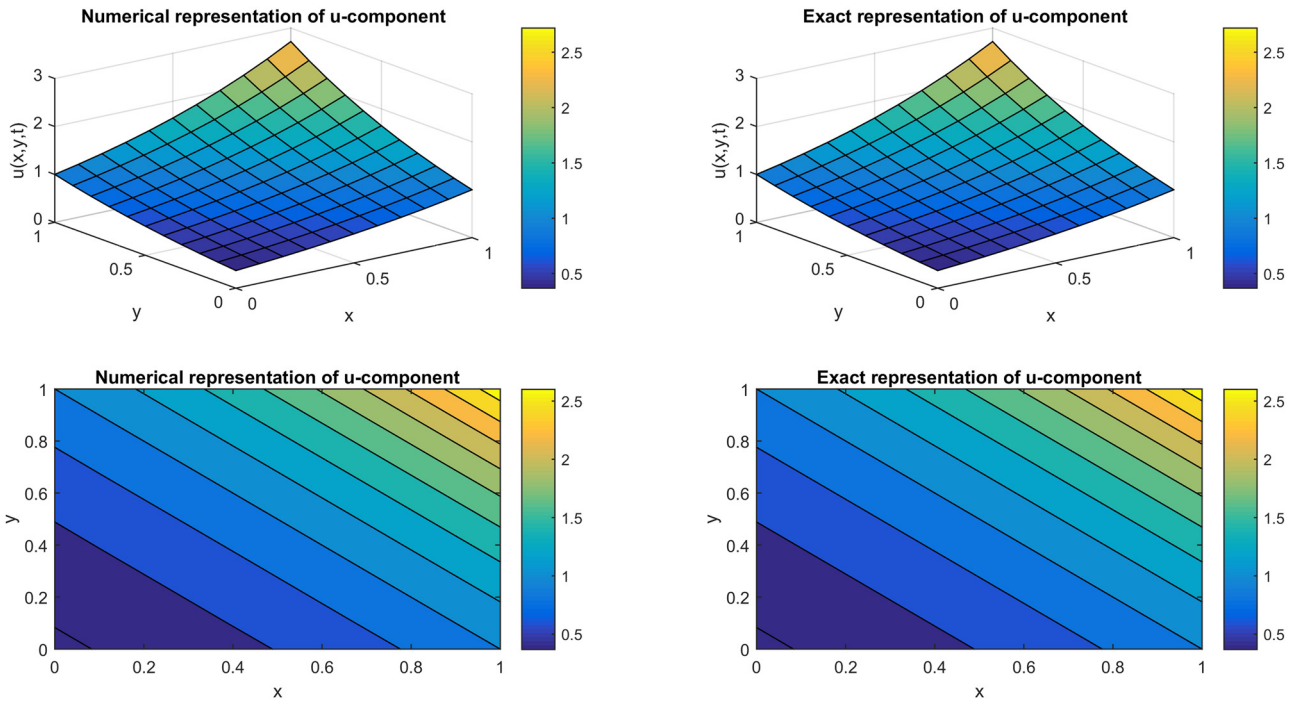

Figure 15: Surface and contour plots for u component at $t = 1$ for $N = 11$, $\alpha = 1$, $\beta = 1$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method I regarding Example 8.

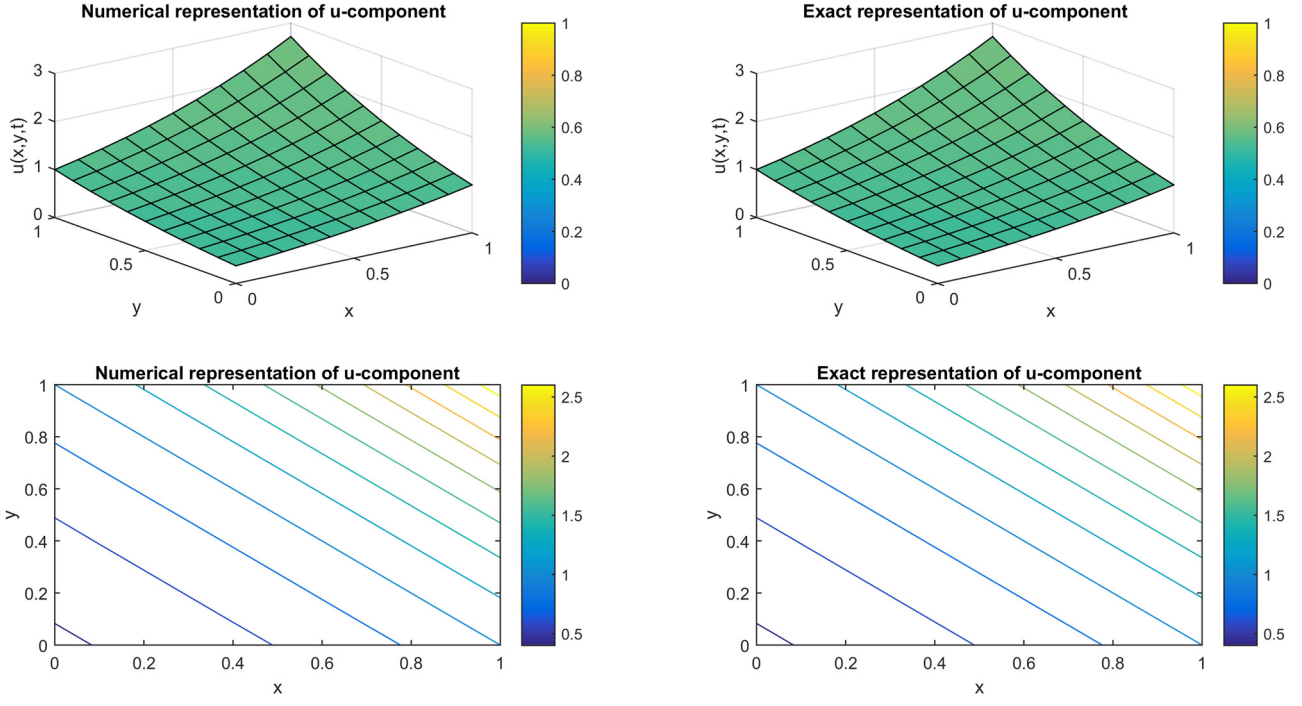

Figure 16: Surface and contour plots for u component at $t = 1$ for $N = 11$, $\alpha = 1$, $\beta = 1$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 8.

Table 23: Comparison of errors for $N = 11$, $\Delta t = 0.0001$, $\alpha = 1$, $\beta = 1$, and $\tau = 0.5$ regarding Example 8

	[8]		Method I	
t	L_2	L_∞	L_2u	$L_\infty u$
1	1.44×10^{-2}	3.00×10^{-2}	1.51×10^{-3}	8.22×10^{-4}
2	1.39×10^{-3}	3.97×10^{-3}	6.71×10^{-4}	4.40×10^{-4}
3	1.30×10^{-3}	2.22×10^{-3}	1.09×10^{-4}	1.28×10^{-4}
	[8]		Method II	
t	L_2	L_∞	L_2u	$L_\infty u$
1	1.44×10^{-2}	3.00×10^{-2}	1.51×10^{-3}	8.21×10^{-4}
2	1.39×10^{-3}	3.97×10^{-3}	6.70×10^{-4}	4.40×10^{-4}
3	1.30×10^{-3}	2.22×10^{-3}	1.09×10^{-4}	1.27×10^{-4}

where m_k can be determined by using the interpolation conditions given as follows [31]:

$$h(x_i) = u(x_i), \quad i = 0, 1, \ldots, n. \tag{36}$$

$$h''(a) = u''(a) - \frac{h^2}{12}u^{(4)}(a), \tag{37}$$

$$h''(b) = u''(b) - \frac{h^2}{12}u^{(4)}(b). \tag{38}$$

We can construct the approximation of $u(x)$ as follows:

$$u'(x_i) \equiv U'(x_i) = \sum_{j=-1}^{(n+1)} w_{ij}^{(1)} U(x_j), \quad i = -1, \ldots, (n+1). \tag{39}$$

Theorem. *Let $v(x) \in C^6[a, b]$. Then the aforementioned two approximations of $u'(x_i)$ and $u''(x_i)$ have the following error bounds for $i = -1, \ldots, n+1$.*

$$|u'(x_i) - U'(x_i)| = O(h^4),$$

$$|u''(x_i) - U''(x_i)| = O(h^4).$$

Proof. Let $h(x)$ is the cubic hyperbolic B-spline interpolant of $u(x)$ the by using the, and we will obtain the following,

$$|u'(x_i) - U'(x_i)| = |(u'(x_i) - h'(x_i)) + (h'(x_i) - U'(x_i))|, \tag{40}$$

$$|u'(x_i) - U'(x_i)| \le |u'(x_i) - h'(x_i)| + |h'(x_i) - U'(x_i)|, \tag{41}$$

where the first term of RHS of the aforementioned equation is having order four by ref. [31]. For the second term, it is considered that,

$$h'(x_i) - U'(x_i) = \sum_{k=-1}^{(n+1)} m_k \mathrm{UATB}_k'(x_i) \mathrm{UAHB}_k'(x_i) - \sum_{j=-1}^{(n+1)} w_{ij}^{(1)} U(x_j),$$

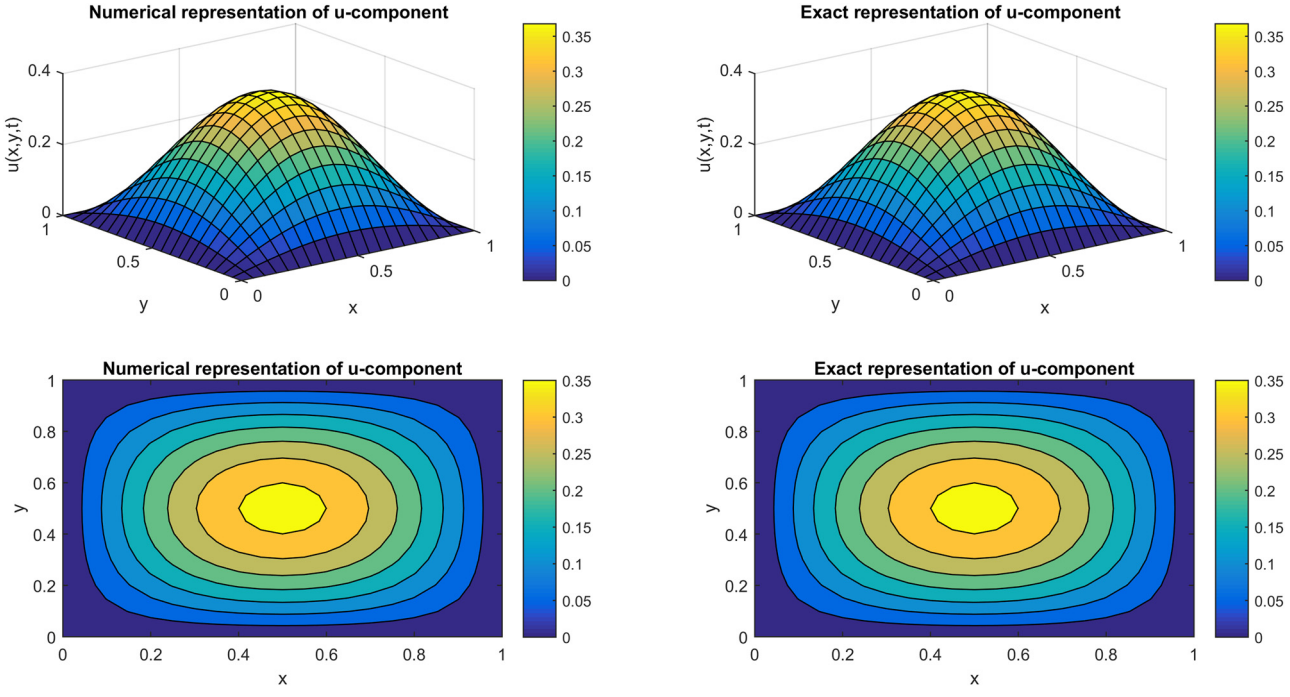

Figure 17: Surface and contour plots for u component at $t = 1$ for $N = 21$, $\alpha = 1$, $\beta = 1$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method I regarding Example 9.

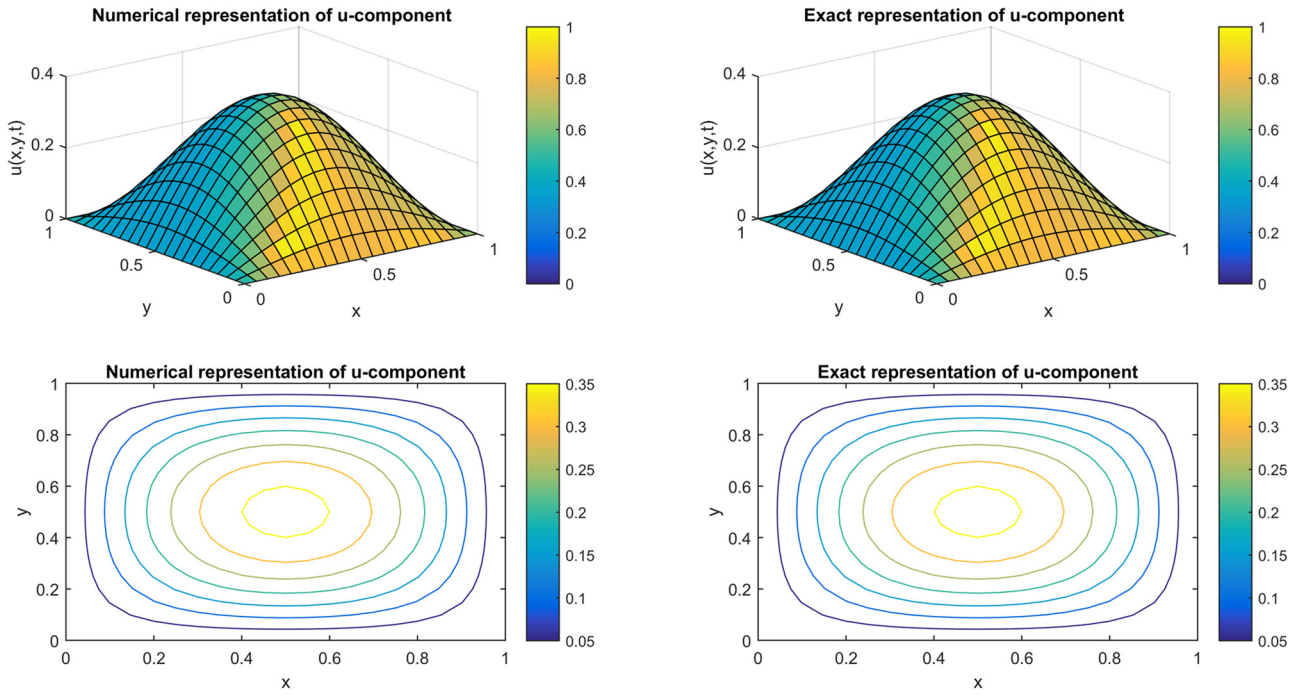

Figure 18: Surface and contour plots for u component at $t = 1$ for $N = 21$, $\alpha = 1$, $\beta = 1$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 9.

Table 24: Comparison of errors for $N = 21$, $\Delta t = 0.0001$, $\alpha = 1$, $\beta = 1$, and $\tau = 0.5$ regarding Example 9

	[8]		Method I	
t	L_2	L_∞	L_2u	$L_\infty u$
1	1.61×10^{-3}	3.70×10^{-3}	2.09×10^{-4}	9.00×10^{-5}
2	2.63×10^{-3}	5.71×10^{-3}	9.13×10^{-5}	4.18×10^{-5}
3	5.38×10^{-4}	1.25×10^{-3}	1.51×10^{-5}	8.20×10^{-6}

$$h'(x_i) - U'(x_i) = \sum_{k=-1}^{(n+1)} m_k \sum_{j=-1}^{(n+1)} w_{ij}^{(1)} \mathrm{UATB}_k(x_j) \mathrm{UAHB}_k(x_j) - \sum_{j=-1}^{(n+1)} w_{ij}^{(1)} U(x_j),$$

$$h'(x_i) - U'(x_i) = \sum_{j=-1}^{(n+1)} w_{ij}^{(1)} [\sum_{k=-1}^{(n+1)} m_k \mathrm{UATB}_k(x_j) \mathrm{UAHB}_k(x_j) - U(x_j)],$$

$$h'(x_i) - U'(x_i) = w_{i1}^{(1)}[h(x_1) - U(x_1)] + w_{in}^{(1)}[h(x_n) - U(x_n)] + \sum_{j=-1, j\neq i, j\neq n}^{(n+1)} w_{ij}^{(1)}[h(x_j) - U(x_j)],$$

$$h'(x_i) - U'(x_i) = O(h^4) + O(h^4) + 0 = O(h^4).$$

Similarly,

$$|u''(x_i) - U''(x_i)| = |(u''(x_i) - h''(x_i)) + (h''(x_i) - U''(x_i))|, \tag{42}$$

$$|u''(x_i) - U''(x_i)| \leq |u''(x_i) - h''(x_i)| + |h''(x_i) - U''(x_i)|, \tag{43}$$

where the first term of RHS of the aforementioned equation is having order four by ref. [31]. For the second term, it is considered that,

$$h''(x_i) - U''(x_i) = \sum_{k=-1}^{(n+1)} m_k \mathrm{UATB}_k''(x_i) \mathrm{UAHB}_k''(x_i) - \sum_{j=-1}^{(n+1)} w_{ij}^{(2)} U(x_j),$$

$$h''(x_i) - U''(x_i) = \sum_{k=-1}^{(n+1)} m_k \sum_{j=-1}^{(n+1)} w_{ij}^{(2)} \mathrm{UATB}_k(x_j) \mathrm{UAHB}_k(x_j) - \sum_{j=-1}^{(n=1)} w_{ij}^{(2)} U(x_j),$$

$$h''(x_i) - U''(x_i) = \sum_{j=-1}^{(n+1)} w_{ij}^{(2)} \left[\sum_{k=-2}^{(n+1)} m_k \mathrm{UATB}_k(x_j) \mathrm{UAHB}_k(x_j) - U(x_j) \right],$$

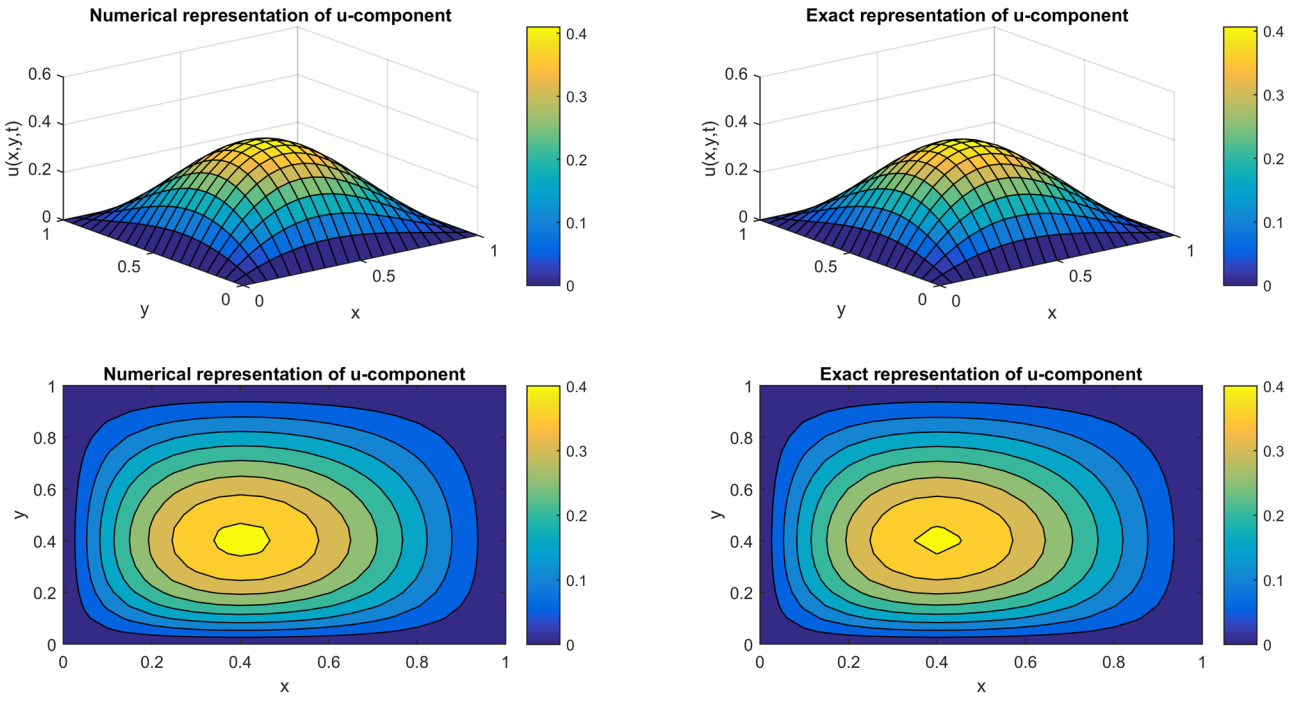

Figure 19: Surface and contour plots for u component at $t = 1$ for $N = 21$, $\alpha = 1$, $\beta = 1$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method I regarding Example 10.

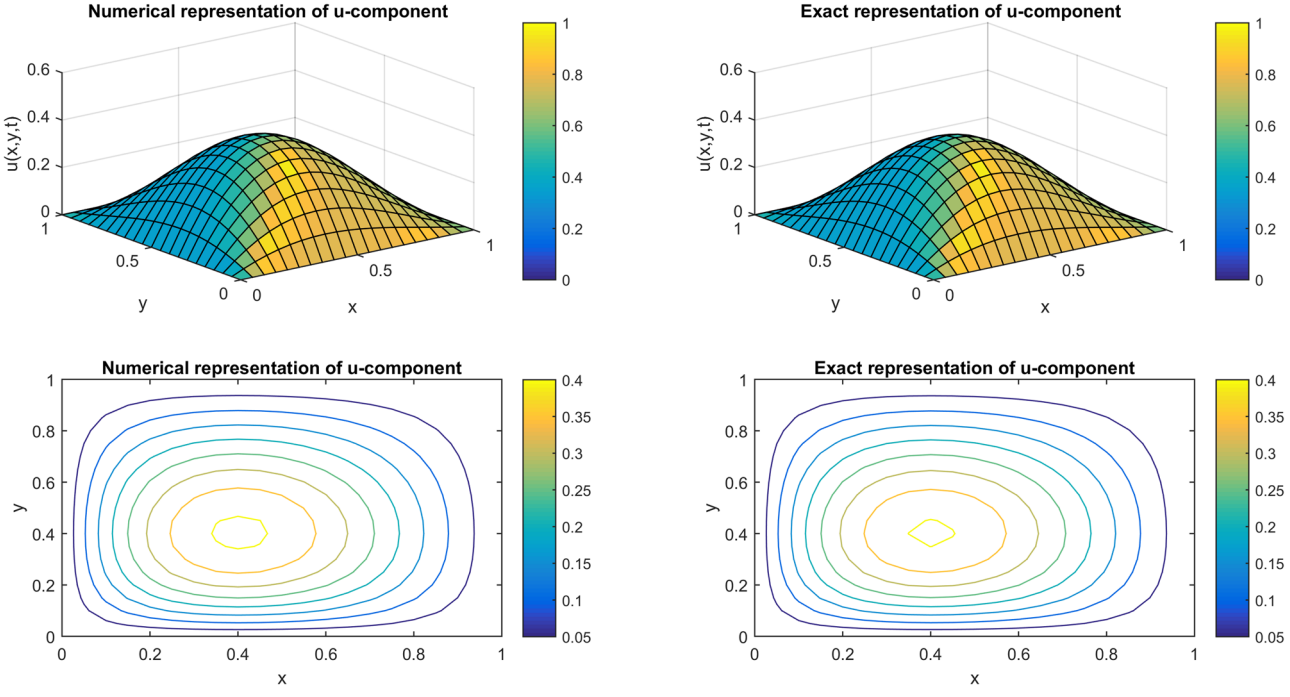

Figure 20: Surface and contour plots for u component at $t = 1$ for $N = 21$, $\alpha = 1$, $\beta = 1$, $\Delta t = 0.0001$, and $\tau = 0.5$ using Method II regarding Example 10.

$$h''(x_i) - U''(x_i) = w_{i1}^{(2)}[g(x_1) - U(x_1)] + w_{in}^{(2)}[g(x_n) - U(x_n)] + \sum_{j=-1, j\neq i, j\neq n}^{(n+1)} w_{ij}^{(2)}[g(x_j) - U(x_j)],$$

$$h''(x_i) - U''(x_i) = O(h^4) + O(h^4) + 0 = O(h^4).$$

6 Conclusion

In the current study, two regimes known as UAT tension B-spline DQM and UAH tension B-spline regimes are used to numerically approximate 1D and 2D hyperbolic telegraph equations. Ten numerical examples in all are discussed in the current study; the first four examples concern the 1D hyperbolic telegraph equation and the remaining six instances concern the 2D hyperbolic telegraph equation. The validity of the anticipated regimes is examined after graphic and tabular analysis of these examples. The current regimes have produced more often than not better and acceptable results. The main focus of this research is the comparison of two innovative regimes, and it is noted that UAH tension B-spline DQM is marginally superior than UAT tension B-spline DQM, meaning that Method II is marginally superior to Method I.

Robustness of these methods is claimed *via* the comparison of results. It is observed that in Table 4 Method I has produced better results than [49]. In Table 6, Method II has generated much better results than [49] for L_2 error. In Tables 9 and 10, results of this study are better than [18] at $t = 1.5$. In Table 13, produced results are far better than [8] at a wide range of time levels. Stability and convergence analysis are also provided. These methods are completely novel to solve 1D and 2D hyperbolic telegraph equations. Finding analytical solutions is not an easy task in most of the cases, and therefore, these developed regimes are important contributions in the literature. These developed regimes will surely be helpful to tackle other equations of importance.

Funding information: Author state that no funding involved.

Author contribution: Author has accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: Author do not have any conflict of interest.

Future scope of the work: Numerous partial differential equations, such as fractional equations, integro differential equations, partial integro differential equations, and many others, cannot be resolved analytically, as the future scope of the work.

Data availability statement: All data are included within the manuscript.

References

[1] Pascal H. Pressure wave propagation in a fluid flowing through a porous medium and problems related to interpretation of Stoneley's wave attenuation in acoustical well logging. Int J Eng Sci. 1986;24(9):1553–70.
[2] Böhme G. Non-Newtonian fluid mechanics. Elsevier; 2012.
[3] Evans DJ, Bulut H. The numerical solution of the telegraph equation by the alternating group explicit (AGE) method. Int J Comput Math. 2003;80(10):1289–97.
[4] Jordan P, Meyer MR, Puri A. Causal implications of viscous damping in compressible fluid flows. Phys Rev E. 2000;62(6):7918.
[5] Mohanty R. New unconditionally stable difference schemes for the solution of multi-dimensional telegraphic equations. Int J Comput Math. 2009;86(12):2061–71.
[6] Lakestani M, Saray BN. Numerical solution of telegraph equation using interpolating scaling functions. Comput Math Appl. 2010;60(7):1964–72.
[7] Arora G, Singh BK. Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method. Appl Math Comput. 2013;224:166–77.
[8] Mittal R, Bhatia R. A numerical study of two dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method. Appl Math Comput. 2014;244:976–97.
[9] Ersoy O, Dag I. Numerical solutions of the reaction diffusion system by using exponential cubic B-spline collocation algorithms. Open Phys. 2015;13(1):414–27.
[10] Mittal R, Bhatia R. Numerical solution of second order one dimensional hyperbolic telegraph equation by cubic B-spline collocation method. Appl Math Comput. 2013;220:496–506.
[11] Dehghan M, Ghesmati A. Solution of the second-order one-dimensional hyperbolic telegraph equation by using the dual reciprocity boundary integral equation (DRBIE) method. Eng Anal Boundary Elements. 2010;34(1):51–9.
[12] Arora G, Mittal R, Singh B. Numerical solution of BBM-Burger equation with quartic B-spline collocation method. J Eng Sci Technol. 2014;9:104–16.
[13] Bülbül B, Sezer M. Taylor polynomial solution of hyperbolic type partial differential equations with constant coefficients. Int J Comput Math. 2011;88(3):533–44.
[14] Gao F, Chi C. Unconditionally stable difference schemes for a one-space-dimensional linear hyperbolic equation. Appl Math Comput. 2007;187(2):1272–6.

[15] Dehghan M, Yousefi S, Lotfi A. The use of He's variational iteration method for solving the telegraph and fractional telegraph equations. Int J Numer Meth Biomed Eng. 2011;27(2):219–31.
[16] Sharifi S, Rashidinia J. Numerical solution of hyperbolic telegraph equation by cubic B-spline collocation method. Appl Math Comput. 2016;281:28–38.
[17] Dehghan M, Ghesmati A. Combination of meshless local weak and strong (MLWS) forms to solve the two dimensional hyperbolic telegraph equation. Eng Anal Boundary Elements. 2010;34(4):324–36.
[18] Jiwari R, Pandit S, Mittal R. A differential quadrature algorithm to solve the two dimensional linear hyperbolic telegraph equation with Dirichlet and Neumann boundary conditions. Appl Math Comput. 2012;218(13):7279–94.
[19] Ding H, Zhang Y. A new fourth-order compact finite difference scheme for the two-dimensional second-order hyperbolic equation. J Comput Appl Math. 2009;230(2):626–32.
[20] Koksal ME, Senol M, Unver AK. Numerical simulation of power transmission lines. Chinese J Phys. 2019;59:507–24.
[21] Ashyralyev A, Koksal M. On the numerical solution of hyperbolic PDEs with variable space operator. Numer Meth Partial Differ Equ. 2009;25:1084–96.
[22] Ashyralyev A, Koksal M. Stability of a second order of accuracy difference scheme for hyperbolic equation in a hilbert space. Discrete Dynam Nature Soc. 2007;2007:1–25.
[23] Asif M, Haider N, Al-Mdallal Q, Khan I. A Haar wavelet collocation approach for solving one and two-dimensional second-order linear and nonlinear hyperbolic telegraph equations. Numer Meth Partial Differ Equ. 2020;36(6):1962–81.
[24] Wang F, Hou E, Ahmad I, Ahmad H, Gu Y. An efficient meshless method for hyperbolic telegraph equations in (1+1) dimensions. Cmes-comput Model Eng. 2021;128(2):687–98.
[25] Zhou Y, Qu W, Gu Y, Gao H. A hybrid meshless method for the solution of the second order hyperbolic telegraph equation in two space dimensions. Eng Anal Boundary Elements. 2020;115:21–7.
[26] Khan H, Shah R, Baleanu D, Kumam P, Arif M. Analytical solution of fractional-order hyperbolic telegraph equation, using natural transform decomposition method. Electronics. 2019;8(9):1015.
[27] Wang F, Hou E. A direct meshless method for solving two-dimensional second-order hyperbolic telegraph equations. J Math. 2020;2020:1–9.
[28] Lin J, Chen F, Zhang Y, Lu J. An accurate meshless collocation technique for solving two-dimensional hyperbolic telegraph equations in arbitrary domains. Eng Anal Boundary Elements. 2019;108:372–84.
[29] Ahmad I, Seadawy AR, Ahmad H, Thounthong P, Wang F. Numerical study of multi-dimensional hyperbolic telegraph equations arising in nuclear material science via an efficient local meshless method. Int J Nonlinear Sci Numer Simulat. 2022;23(1):115–22.
[30] Ahmad I, Ahmad H, Abouelregal AE, Thounthong P, Abdel-Aty M. Numerical study of integer-order hyperbolic telegraph model arising in physical and related sciences. European Phys J Plus. 2020;135(9):1–14.
[31] Arora G, Joshi V. Comparison of numerical solution of 1D hyperbolic telegraph equation using B-spline and trigonometric B-spline by differential quadrature method. Indian J Sci Technol. 2016;9(45):1–8.
[32] Singh BK, Kumar P. An algorithm based on a new DQM with modified extended cubic B-splines for numerical study of two dimensional hyperbolic telegraph equation. Alexandr Eng J. 2018;57(1):175–91.
[33] Bellman R, Kashef B, Casti J. Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations. J Comput Phys. 1972;10(1):40–52.
[34] Quan J, Chang C. New insights in solving distributed system equations by the quadrature method-I. Analysis. Comput Chem Eng. 1989;13(7):779–88.
[35] Shu C. Differential quadrature and its application in engineering. London, UK: Springer Science & Business Media; 2012.
[36] Tamsir M, Srivastava VK, Jiwari R. An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers' equation. Appl Math Comput. 2016;290:111–24.
[37] Tamsir M, Meetei MZ, Msmali AH. Hyperbolic B-spline function-based differential quadrature method for the approximation of 3D wave equations. Axioms. 2022;11(11):597.
[38] Tamsir M, Dhiman N. DQM based on the modified form of CTB shape functions for coupled Burgers' equation in 2D and 3D. Int J Math Eng Manag Sci. 2019;4(4):1051.
[39] Shukla H, Tamsir M, Jiwari R, Srivastava VK. A numerical algorithm for computation modelling of 3D nonlinear wave equations based on exponential modified cubic B-spline differential quadrature method. Int J Comput Math. 2018;95(4):752–66.
[40] Tamsir M, Dhiman N, Srivastava VK. Cubic trigonometric B-spline differential quadrature method for numerical treatment of Fisherâ's reaction-diffusion equations. Alexandr Eng J. 2018;57(3):2019–26.
[41] Shukla H, Tamsir M. An exponential cubic B-spline algorithm for multi-dimensional convection-diffusion equations. Alexandr Eng J. 2018;57(3):1999–2006.
[42] Shukla H, Tamsir M, Srivastava VK, Rashidi MM. Modified cubic B-spline differential quadrature method for numerical solution of three-dimensional coupled viscous Burger equation. Modern Phys Lett B. 2016;30(11):1650110.
[43] Shukla H, Tamsir M. Extended modified cubic B-spline algorithm for nonlinear Fisheras reaction-diffusion equation. Alexandr Eng J. 2016;55(3):2871–9.
[44] Shukla H, Tamsir M, Srivastava VK. Numerical simulation of two dimensional sine-Gordon solitons using modified cubic B-spline differential quadrature method. AIP Adv. 2015;5(1):017121.
[45] Wang G, Fang M. Unified and extended form of three types of splines. J Comput Appl Math. 2008;216(2):498–508.
[46] Alinia N, Zarebnia M. A new tension B-spline method for third-order self-adjoint singularly perturbed boundary value problems. J Comput Appl Math. 2018;342:521–33.
[47] Alinia N, Zarebnia M. A numerical algorithm based on a new kind of tension B-spline function for solving Burgers-Huxley equation. Numer Algorithms. 2019;82(4):1121–42.
[48] Mittal R, Dahiya S. Numerical simulation of three-dimensional telegraphic equation using cubic B-spline differential quadrature method. Appl Math Comput. 2017;313:442–52.

[49] Dehghan M, Shokri A. A numerical method for solving the hyperbolic telegraph equation. Numer Meth Partial Differ Equ Int J. 2008;24(4):1080–93.
[50] Spiteri RJ, Ruuth SJ. A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J Numer Anal. 2002;40(2):469–91.
[51] Shu CW. Total-variation-diminishing time discretizations. SIAM J Scientif Stat Comput. 1988;9(6):1073–84.
[52] Tomasiello S. Stability and accuracy of the iterative differential quadrature method. Int J Numer Meth Eng. 2003;58(9):1277–96.
[53] Tomasiello S. Numerical solutions of the Burgers-Huxley equation by the IDQ method. Int J Comput Math. 2010;87(1):129–40.