Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 15, No. 2, 2023 Article ID IJIM-1585, 13 pages
DOI: http://dx.doi.org/10.30495/ijim.2023.66141.1585
Research Article

A Modification on The Exponential Cubic B-spline for Numerical Simulation of Hyperbolic Telegraph quations

AR. Haghighi *, F. Rahimian †, N. Asghary ‡§, M. Roohi ¶

Received Date: 2022-02-18 Revised Date: 2022-06-15 Accepted Date: 2022-07-08

Abstract

In this paper the differential quadrature method is implemented to find numerical solution of two and three-dimensional telegraphic equations with Dirichlet and Neumanns boundary values. This technique is according to exponential cubic B-spline functions. So, a modification on the exponential cubic B- spline is applied in order to use as a basis function in the DQ method. Therefore, the Telegraph equation (TE) is altered to a system of ordinary differential equations (ODEs). The optimized form of Runge-Kutta scheme has been implemented by four-stage and three-order strong stability preserving (SSPRK43) to solve the resulting system of ODEs. We examined the correctness and applicability of this method by four examples of the TE.

Keywords : Telegraph equation (TE); Exponential modified cubic B-spline function; SSP-RK43; Differential quadrature method.

1 Introduction

Suppose the following two dimensional second-order linear hyperbolic TE as:

$$\begin{aligned} &u_{tt}(x,y,t)+2\gamma u_t(x,y,t)+\delta^2 u(x,y,t)\\ &=u_{xx}(x,y,t)+u_{yy}(x,y,t)\\ &+f(x,y,t);(x,y)\in\Psi \end{aligned} \tag{1.1}$$

With initial conditions

$$\begin{aligned} u(x,y,0)&=u_0(x,y) \quad ;(x,y)\in\Psi\\ u_t(x,y,0)&=\nu_0(x,y) \quad ;(x,y)\in\Psi \end{aligned} \tag{1.2}$$

And the boundary values:
(a) Dirichlet boundary values

$$\begin{aligned} u(0,y,t)&=f_1(x,y) \quad u(1,y,t)=f_2(y,t)\\ u(x,y,t)&=f_3(x,y) \quad u(x,1,t)=f_4(x,t) \end{aligned} \tag{1.3}$$

(b) Neumann boundary values

$$\begin{aligned} u_x(x,y,t)&=g_1(y,t) \quad ;u_x(1,y,t)=g_2(y,t)\\ u_y(x,0,t)&=g_3(x,t) \quad ;u_y(x,1,t)=g_4(x,t) \end{aligned} \tag{1.4}$$

where $\Psi=\{(x,y);a\le x\le b;c\le y\le d\}$ is the computational square domain and $(0,T)$ is the time step and γ,δ are known constant coefficients.we suppose that $u_0,\nu_0,f_i,g_i(i=1,2,3,4)$

*Department of Mathematics, Allameh Tabatabai University, Tehran, Iran.
†Department of Mathematics, Urmia University of Technology, Urmia, Iran.
‡Corresponding author. nasim.asghary@gmail.com, Tel:+98(919)3996782.
§Faculty of Basic Sciences, Islamic Azad University, Central Tehran Branch, Tehran, Iran.
¶Department of Mathematics, School of Economics and Statistics, Guangzhou, China.

are known as smooth functions.
For $\gamma \geq 0$ and $\delta = 0$, Eq. (1.1) shows a wave equation with damping property and for $\gamma \geq \delta \geq 0$ named the TE.
The TE in three-space dimension is represented as:

$$\begin{aligned} &u_{tt}(x,y,z,t) + 2\gamma u_t(x,y,z,t) + \\ &\delta^2 u(x,y,z,t) = u_{xx}(x,y,z,t) + \\ &u_{yy}(x,y,z,t) + f(x,y,z,t), \\ &(x,y,z,t) \in \Psi; t \geq 0 \end{aligned} \tag{1.5}$$

With initial conditions

$$\begin{aligned} u(x,y,z,0) &= u_0(x,y,z) \quad ;(x,y,z) \in \Psi \\ u_t(x,y,z,0) &= \nu_0(x,y,z) \quad ;(x,y,z) \in \Psi \end{aligned} \tag{1.6}$$

And the Dirichlet boundary conditions

$$\begin{aligned} &u(0,y,z,t) = f_1(y,z,t); u(1,y,z,t) \\ &= f_2(y,z,t) \\ &u(x,0,z,t) = f_3(x,z,t); u(x,1,z,t) \\ &= f_4(x,z,t) \\ &u(x,y,0,t) = f_5(x,y,t); u(x,y,1,t) \\ &= f_6(x,y,t) \end{aligned} \tag{1.7}$$

The simulation of many physical events in different aspects of applied science described by hyperbolical partial differential equations.These types of equations have an important role in the formulation of basic equations in atomic physics and the presentation of several phenomena in applied sciences like aerospace professions, chemistry and biology are applicable [1, 2]. In the recent years, there have been many efforts in the study and development of stable numerical schemes in order to solve hyperbolical PDEs which can be pointed to [3]-[9].
One of the most practical equations in communication and electronic engineering is the TE. Actually, the connection between voltage and flows, taking into account the distance and time along the transmission line, is described by the TE. This equation was introduced in 1880 [10, 11]. So far, various methods have been proposed to compute the approximate solution of TE in two space dimension. In this case, . In [18], a comparison of basic functions in DQM named as Lagrange and cubic B-Spline interpolation, is presented to calculate the numerical answer of the TE.
In [12] a numerical procedure based on Hermit wavelets has been implemented to obtain the approximate solution of the TE. The authors of [13] achieved a proper accuracy for the numerical solution of twodimensional hyperbolic TE by using Bernoulli matrix. In [14], an accurate meshless collocation technique has been investigated in order to find the numerical answer of the two-dimensional telegraphic equations in arbitrary domains.Also, the authors in [15], obtained the approximate solution of TE in the two-space dimension subject to Dirichlet boundary conditions by using Lagranges operational approach. In [16] a continuous Galerkin method with mesh modification has been implemented for numerical solution of two-dimensional telegraph equation. A hybrid meshless method has been used in [17] for the solution of the second order hyperbolic telegraph equation in two space dimensions .The numerical solution of these types of equations by a reduced differential transform method is expressed in [18]. Besides, the authors in [19], have collocated initial and boundary values, like the Galerkin method ,in order to get approximate solution of TE in two-dimension. But the common point in most of the above methods has not been applied to the Newman boundary conditions to the 2D TE. Since most of the real world, problems depend on solving partial differential equations with Newman boundary conditions, so in this study, in addition to Dirichlet boundary conditions, Newman boundary conditions are also intended. An efficient numerical technique for the solution of PDEs is the Differential quadrature method (DQM) [**?**]. According to DQM, adding all the functional values at a certain point in the whole computational domain leads to computing the derivatives of passive functions with respect to space components. The main point of the differential quadrature method is to determine the weighting coefficients. So far, several kinds of test functions such as B-spline function, Lagrange polynomials, cubic B- spline functions, Legendre polynomials, quartic B-spline functions, trigonometric B-spline functions, modified cubic B-spline functions [11] and so on have been used in order to compute the weighting coefficients.
In this paper, a differential quadrature method is

presented based on a modified exponential cubic B-spline functions for the numerical simulation of the TE. Also, in this method by making a modification in exponential cubic B- spline functions and utilizing them as the basis functions in DQM, the unknown weighting coefficients will be determined. Then the TE reformed to an ODE system of first-order. By applying strong stability preserving time- staging Runge-Kutta method in four-step the resulting system is solved.Moreover, we confirmed the efficiency and compatibility of the method by considering four examples of the TE in two and three space- dimensions. Briefly, the most important benefits of this article are noted as follows:

1. Solve the TE by considering the Newman and Dirichlet boundary values in two- space dimensions.
2. Solve the TE with Dirichlet boundary values in three- space dimensions.
3. Use a new interpolation method named as exponential modified cubic B-spline function.
4. High accuracy compared to the above methods and high speed in calculations.

The structure of this paper is as follows: In Section 2, a modification in the exponential cubic B- spline DQM is demonstrated. The third part provides different stages of exerting method on the TE with Dirichlet and Neumanns boundary values. The fourth section has Numerical simulations and analytical comparisons. And finally, the conclusion is represented in Section 5.

2 A Modification in Exponential Cubic B- Spline DQM

The first step in two-dimensional DQM is discretizing the domainas $\Psi^1 = \{(x_i, y_j); i = 1,2,\ldots,N; j = 1,2,\ldots,M\}$ by taking space step $\Delta x = x_i - x_{i-1}$ and $\Delta y = y_j - y_{j-1}$ in the X-axis direction and Y-axis direction.

In this method, for first-order partial derivatives of the function $u(x,y,t)$ concerning x, at point x_i (keeping y_j fixed) is approximated as follow:

$$u_x(x_i, y_j, t) = \sum_{k=1}^{N} w_{ik}^{(1)} u(x_k, y_j, t); i = 1,2,\ldots,N \tag{2.8}$$

Similarly, the second-order partial derivatives of the function $u(x,y,t)$ concerning x, at point x_i (keeping y_j fixed) can approximated as follow:

$$u_{xx}(x_i, y_j, t) = \sum_{k=1}^{N} w_{ik}^{(2)} u(x_k, y_j, t); i = 1,2,\ldots,N \tag{2.9}$$

The first-order partial derivatives of the function $u(x,y,t)$ with respect to y, at the at point y_j (keeping x_i fixed) can be approximated as follow:

$$u_y(x_i, y_j, t) = \sum_{k=1}^{M} \overline{w}_{jk}^{(1)} u(x_i, y_k, t); j = 1,2,\ldots,M \tag{2.10}$$

Similarly, the second-order partial derivatives of the function $u(x,y,t)$ to concerning y, at the point y_j (keeping x_i fixed) is approximated as:

$$u_{yy}(x_i, y_j, t) = \sum_{k=1}^{M} \overline{w}_{jk}^{(2)} u(x_i, y_k, t); j = 1,2,\ldots,M \tag{2.11}$$

B-Spline functions are smooth and piecewise polynomials that these desirable features make it easier to integrate and differentiate than other interpolation methods.Since its smoothness and capability to handle local phenomena these basis functions have more influence than other basic functions.

Suppose The following exponent cubic B-spline functions [12]

$$E_j(x) = \frac{1}{h^3}\begin{cases} b_2((x_{j-2}-x) - \frac{1}{p}(\sinh(p(x_{j-2} \\ -x)))); x \in [x_{j-2}, x_{j-1}) \\ a_1 + b_1(x_j - x) + c_1 \exp(p(x_j \\ -x)) + d_1 \exp(-p(x_j \\ -x)); x \in [x_{j-1}, x_j] \\ a_1 + b_1(x - x_j) + c_1 \exp \\ (p(x - x_j)) + d_1 \\ \exp(-p(x - x_j)); x \in [x_j, x_{j+1}] \\ b_2((x - x_{j+2}) - \frac{1}{p}(\sinh(p(x \\ -x_{j+2})))); x \in [x_{j+1}, x_{j+2}) \\ 0; otherwise \end{cases} \tag{2.12}$$

Where p is the free parameter and
$a_1 = \frac{ph\cosh(ph)}{ph\cosh(ph)-\sinh(ph)}$,

$b_1 = \frac{1}{4}\left[\frac{\exp(ph)(\cosh(ph)-1)+\sinh(ph)(\exp(ph)-1)}{(ph\cosh(ph)-\sinh(ph))(1-\cosh(ph))}\right]$

$c_1 = \frac{1}{4}\left[\frac{\exp(-ph)(1-\cosh(ph))+\sinh(ph)(\exp(-ph)-1)}{(ph\cosh(ph)-\sinh(ph))(1-\cosh(ph))}\right]$,

$d_1 = \frac{1}{4}\left[\frac{\exp(ph)(\cosh(ph)-1)+\sinh(ph)(\exp(ph)-1)}{(ph\cosh(ph)-\sinh(ph))(1-\cosh(ph))}\right]$

Also the collection $\{E_0(x), E_1(x), \ldots, E_N(x), E_{N+1}(x)\}$ organized a basic set over $[a, b]$. For (2.12) Table 1 shows the values for the $E_j(x)$ and and their derivations in node points.

Table 1: Values of $E_j(x), E_j'(x), E_j''(x)$ and its derivatives at the nodal points

	$E_j(x)$
x_{j-2}	0
x_{j-1}	$\frac{\sin(ph)-ph}{2(ph\cosh(ph)-\sinh(ph))}$
x_j	1
x_{j+1}	$\frac{\sin(ph)-ph}{2(ph\cosh(ph)-\sinh(ph))}$
x_{j+2}	0
	$E_j'(x)$
x_{j-2}	0
x_{j-1}	$\frac{p(\cosh(ph)-1)}{2(ph\cosh(ph)-\sinh(ph))}$
x_j	0
x_{j+1}	$-\frac{p(\cosh(ph)-1)}{2(ph\cosh(ph)-\sinh(ph))}$
x_{j+2}	0
	$E_j''(x)$
x_{j-2}	0
x_{j-1}	$\frac{p^2(\sinh(ph))}{2(ph\cosh(ph)-\sinh(ph))}$
x_j	$\frac{p^2(\sinh(ph))}{(ph\cosh(ph)-\sinh(ph))}$
x_{j+1}	$\frac{p^2(\sinh(ph))}{2(ph\cosh(ph)-\sinh(ph))}$
x_{j+2}	0

A modification of $E_j(x)$ is considered so that the resulting matrix system is diagonally dominant.

Therefore, the modified $E_j(x)$ at nodal points are expressed as:

$$\begin{cases} \Phi_1(x) = E_1(x) + 2E_0(x) \\ \Phi_2(x) = E_2(x) + 2E_0(x) \\ \Phi_N(x) = E_i(x); i = 3, 4, \ldots, N-2 \\ \Phi_{N-1}(x) = E_{N-1}(x) - E_{N+1}(x) \\ \Phi_N(x) = E_N(x) + E_{N+1}(x) \end{cases} \quad (2.13)$$

Again, the functions $\Phi_m(x); m = 1, 2, \ldots, N$ hold a basis set over $[a, b]$.By fixing the y-axis in (2.8), we obtain weighting coefficients $w_{ik}^{(1)}$.Now by inserting the $\Phi_m(x); m = 1, 2, \ldots, N$ in (2.8), we get

$$\Phi_m'(x_i, y_j) = \sum_{k=1}^{N} w_{ik}^{(1)} \Phi_m(x_k, y_j); j = 1, 2, \ldots, M \quad (2.14)$$

Hence the (2.14) is modified to following the tridiagonal system of equations by using Table ??, (2.12) and (2.13):

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} & & & \\ \Phi_{21} & \Phi_{22} & \Phi_{23} & & \\ & \ddots & \ddots & & \\ & & \Phi_{N-1,N-1} & \Phi_{N-1,N} & \\ & & & \Phi_{N,N-1} & \Phi_{N,N} \end{bmatrix} \begin{bmatrix} w_{i,1}^{(1)} \\ w_{i,2}^{(1)} \\ \vdots \\ \vdots \\ w_{i,N-1}^{(1)} \\ w_{i,N}^{(1)} \end{bmatrix} = \begin{bmatrix} \Phi_{1,i}' \\ \Phi_{2,i}' \\ \vdots \\ \vdots \\ \Phi_{N-1,i}' \\ \Phi_{N,i}' \end{bmatrix} \quad (2.15)$$

And the first-order derivatives of the weighting coefficients $w_{i,j}^{(1)}$ can be calculated by employing the Thomas algorithm. For $i = 1$ by assuming that $c = \cosh(ph)$ and $s = \sinh(ph)$ the three diagonal systems are as:

$$\begin{bmatrix} \frac{phc-ph}{phc-s} & \frac{s-ph}{2(phc-s)} & & & & \\ 0 & 1 & \frac{s-ph}{2(phc-s)} & & & \\ & & \ddots & \ddots & & \\ & & \frac{s-ph}{2(phc-s)} & 1 & 1 & 0 \\ & & \frac{s-ph}{2(phc-s)} & \frac{phc-ph}{phc-s} & & \end{bmatrix} \begin{bmatrix} w_{11}^{(1)} \\ w_{12}^{(1)} \\ \vdots \\ \vdots \\ w_{1,N-2}^{(1)} \\ w_{1,N-1}^{(1)} \\ w_{1,N}^{(1)} \end{bmatrix} = \begin{bmatrix} \frac{-p(c-1)}{phc-s} \\ \frac{p(c-1)}{phc-s} \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.16)$$

Also, the Thomas algorithm can be utilized in the above equation to get the weighting coefficients $w_j^{(1)}; j = 1, 2, \cdots, M$.similarly, at the second knot, one has

$$\begin{aligned} &\Phi'_m(x_2, y_j) = \sum_{k=1}^{N} w_{2k}^{(1)} \Phi_m(x_k, y_j); \\ &j = 1, 2, \cdots, M \end{aligned} \tag{2.17}$$

Therefore a tridiagonal system can be produced again as:

$$\begin{bmatrix} \frac{phc-ph}{phc-s} & \frac{s-ph}{2(phc-s)} & & & \\ 0 & 1 & \frac{s-ph}{2(phc-s)} & & \\ & \frac{s-ph}{2(phc-s)} & 1 & \frac{s-ph}{2(phc-s)} & \\ & \ddots & \ddots & \ddots & \\ & \frac{s-ph}{2(phc-s)} & 1 & \frac{s-ph}{2(phc-s)} & \\ & \frac{s-ph}{2(phc-s)} & 1 & 1 & 0 \\ & & \frac{s-ph}{2(phc-s)} & \frac{phc-ph}{phc-s} & \end{bmatrix} \begin{bmatrix} w_{21}^{(1)} \\ w_{22}^{(1)} \\ \vdots \\ \vdots \\ w_{2,N-2}^{(1)} \\ w_{2,N-1}^{(1)} \\ w_{2,N}^{(1)} \end{bmatrix} = \begin{bmatrix} \frac{-p(c-1)}{2(phc-s)} \\ 0 \\ \frac{-p(c-1)}{2(phc-s)} \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.18}$$

Using the Thomas algorithm, weighting coefficients $w_{2j}^{(1)}; j = 1, 2, \cdots, M$ can be achieved.The weighting coefficients $w_{3j}^{(1)}, w_{4j}^{(1)}, \cdots, w_{N-1,j}^{(1)}$, can be computed at each grid point similarly using the Thomas algorithm. At the last grid point, we have

$$\begin{aligned} &\Phi'_m(x_N, y_j) = \sum_{k=1}^{N} \Phi_m(x_k, y_j); \\ &j = 1, 2, \cdots, M \end{aligned} \tag{2.19}$$

Which produce a tridiagonal set of the equation as:

$$\begin{bmatrix} \frac{phc-ph}{phc-s} & \frac{s-ph}{2(phc-s)} & & & & \\ 0 & 1 & \frac{s-ph}{2(phc-s)} & & & \\ & & \ddots & \ddots & & \\ & & & \frac{s-ph}{2(phc-s)} & 1 & 1 \quad 0 \\ & & & \frac{s-ph}{2(phc-s)} & \frac{phc-ph}{phc-s} & \end{bmatrix} \begin{bmatrix} w_{N1}^{(1)} \\ w_{N2}^{(1)} \\ \vdots \\ \vdots \\ w_{N,N-2}^{(1)} \\ w_{N,N-1}^{(1)} \\ w_{N,N}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ \frac{p(c-1)}{2(phc-s)} \\ 0 \\ \frac{-p(c-1)}{2(phc-s)} \end{bmatrix} \tag{2.20}$$

Finally weighting coefficients $w_{N,k}^{(1)}, k = 1, 2, \cdots, N$ is computed by solving the above system. Furthermore, weighting coefficients of second and higher orders can be calculated by using the following recurrence relations [13, 14].

$$\begin{aligned} &w_{i,j}^{(r)} = r\left[w_{i,j}^{(1)} w_{i,i}^{(r-1)} - \frac{w_{i,j}^{(r-1)}}{x_i - x_j}\right], \\ &i \neq j; i, j = 1 \\ &w_{i,i}^{(r)} = -\sum_{j=1, i\neq j}^{N} w_{i,j}^{(r)} \end{aligned} \tag{2.21}$$

where the $(r-1)$th and rth orders of partial derivations with concerning to x are shown as $w_{i,i}^{r-1}$ and $w_{i,j}^{r}$ .we can observe the weighting coefficients $\overline{w}_{ik}^{(1)}$ of first-order partial derivations with concerning to y (by fixing x-axis) in the same procedure with inserting modified $E_j(x)$ in (2.10). We can apply the following recurrence relations (from [13, 14]) in order to calculate the second and higher-order derivations.

$$\begin{aligned} &\overline{w}_{ij}^{r} = r\left[\overline{w}_{i,j}^{(1)} \overline{w}_{i,i}^{(r-1)} - \frac{\overline{w}_{i,j}^{(r-1)}}{x_i - x_j}\right], \\ &i = 1, 2, \cdots, N; r = 2, 3, \cdots, N-1 \\ &\overline{w}_{i,i}^{(r)} = -\sum_{j=1, i\neq j}^{N} \overline{w}_{i,j}^{(r)}; i = j \end{aligned} \tag{2.22}$$

Here the $(r-1)$ th and rth orders of partial derivations with concerning to y is shown as $\overline{w}_{i,j}^{(r-1)}$ and $\overline{w}_{i,j}^{(r)}$.

3 How to Implement The Method For the TE?

We first converted the two-dimensional TE (1), in the following system of PDE utilizing the transformation $\nu = u_t + 2\alpha u$

$$\begin{cases} u_t(x,y,t) = \nu(x,y,t) + 2\alpha u(x,y,t) \\ \nu_t(x,y,t) = u_{xx}(x,y,t) + u_{yy}(x,y,t) - \\ \beta^2 u(x,y,t) + f(x,y,t) \end{cases} \quad (3.23)$$

Here, first and second-order derivation of u at the nodal points $(x_i, y_j); i = 1,2,\dots,N; j = 1,2,\dots,M$ are discretized by the modified exponential cubic B- spline DQM. Therefor (3-1) converted as follows:

$$\begin{cases} \frac{du(x_i,y_j,t)}{dt} = \nu(x_i,y_j,t) + 2\alpha u(x_i,y_j,t) \\ \frac{d\nu(x_i,y_j,t)}{dt} = \sum_{k=1}^{N} w_{i,k}^{(2)} u(x_k,y_j,t) + \\ \sum_{k=1}^{N} \overline{w}_{jk}^{(2)} u(x_i,y_k,t) \\ -\beta^2 u(x_i,y_j,t) + f(x,y,t) \end{cases} \quad (3.24)$$

where $(x_i, y_j, t) \in R \times (0,t]$ and $i = 1,2,\dots,N$, $j = 1,2,\dots,M$.
Using the following initial values

$$\begin{aligned} u(x_i,y_j,0) &= u_0(x_i,y_j); (x,y) \in \Psi \\ \nu(x_i,y_j,0) &= \nu_0(x_i,y_j); (x,y) \in \Psi \end{aligned} \quad (3.25)$$

System (3.24) represents a set of 2N-ODEs. First, we calculate the vector of the initial value $[u_{i,j}^0, \nu_{i,j}^0]; i = 1,2,\dots,N, j = 1,2,\dots,M$ using (3-3).
Then to get the values of u and ν at interior grid points the system (3.24) should be solved by using the SSP-RK43 method .

3.1 How to apply the method for boundary values?

The boundary Dirichlet values are applied at the boundary points and gives the amount of u and ν at these points. When the boundary values are in the following Neumanns form:

$$\begin{cases} u_t(x,y,t) = \nu(x,y,t) + 2\alpha u(x,y,t) \\ \nu_t(x,y,t) = u_{xx}(x,y,t) + u_{yy}(x,y,t) - \\ \beta^2 u(x,y,t) + f(x,y,t) \end{cases} \quad (3.26)$$

By using DQM we discrete the conditions and achieve the answers of u and ν at the following boundary points

$$g_1(0,y,t) = \sum_{k=1}^{N} w_{1k}^{(1)} u(x_k,y_j,t); \; j = 1,2,\dots,M \quad (3.27)$$

$$g_2(1,y,t) = \sum_{k=1}^{N} w_{Nk}^{(1)} u(x_k,y_j,t); \; j = 1,2,\dots,M \quad (3.28)$$

$$g_3(x,0,t) = \sum_{k=1}^{M} \overline{w}_{1k}^{(1)} u(x_i,y_k,t); \; i = 1,2,\dots,N \quad (3.29)$$

$$g_4(x,1,t) = \sum_{k=1}^{M} \overline{w}_{Mk}^{(1)} u(x_i,y_k,t); \; i = 1,2,\dots,N \quad (3.30)$$

(3.27) and (3.28) can be written as

$$w_{1,1}^{(1)} u(x_1,y_j,t) + w_{1,N}^{(1)} u(x_N,y_j,t) = g_1(0,y,t) - \sum_{k=2}^{N-1} w_{1,k}^{(1)} u(x_k,y_j,t) \quad (3.31)$$

$$w_{N,1}^{(1)} u(x_1,y_j,t) + w_{N,N}^{(1)} u(x_N,y_j,t) = g_2(0,y,t) - \sum_{k=2}^{N-1} w_{N,k}^{(1)} u(x_k,y_j,t) \quad (3.32)$$

By solving (3.27) and (3.28) for $w_{1,j}$ and $w_{N,j}$ we get

$$w_{1,j} = \frac{w_{1,N}^{(1)}(g_2 - \sum_{k=2}^{N-1} w_{N,k}^{(1)} u) - w_{N,N}^{(1)}(g_1 - \sum_{k=2}^{N-1} w_{1,k}^{(1)} u)}{(w_{1,N}^{(1)} w_{N,1}^{(1)} - w_{1,1}^{(1)} w_{N,N}^{(1)})}, j = 1,2,\dots,M \quad (3.33)$$

$$w_{N,j} = \frac{w_{N,1}^{(1)}(g_1 - \sum_{k=2}^{N-1} w_{1,k}^{(1)} u) - w_{1,1}^{(1)}(g_2 - \sum_{k=2}^{N-1} w_{N,k}^{(1)} u)}{(w_{1,N}^{(1)} w_{N,1}^{(1)} - w_{1,1}^{(1)} w_{N,N}^{(1)})}, j = 1,2,\dots,M \quad (3.34)$$

Similarly, for $u_{i,1}$ and $u_{i,M}$ we have:

$$u_{i,1} = \frac{\overline{w}_{1,M}^{(1)}(g_4 - \sum_{k=2}^{M-1} \overline{w}_{M,k}^{(1)} u) - \overline{w}_{M,M}^{(1)}(g_3 - \sum_{k=2}^{M-1} \overline{w}_{1,k}^{(1)} u)}{(\overline{w}_{1,M}^{(1)} \overline{w}_{M,1}^{(1)} - \overline{w}_{1,1}^{(1)} \overline{w}_{M,M}^{(1)})}, j = 1,2,\dots,M \quad (3.35)$$

$$u_{i,M} = \frac{\overline{w}_{M,1}^{(1)}(g_3 - \sum_{k=2}^{M-1} \overline{w}_{1,k}^{(1)} u) - \overline{w}_{1,1}^{(1)}(g_4 - \sum_{k=2}^{M-1} \overline{w}_{M,k}^{(1)} u)}{(\overline{w}_{1,M}^{(1)} \overline{w}_{M,1}^{(1)} - \overline{w}_{1,1}^{(1)} \overline{w}_{M,M}^{(1)})}; \; i = 1,2,\dots,M \quad (3.36)$$

4 Numerical Simulation

Here, in order to show the effectiveness and accuracy of the numerical scheme, we applied the suggested numerical technique on some testing problems. These examples are selected because their exact answers are known and already existed in the literature and can be easily compared with our results. Also, the numerical simulations are provided in the Matlab software. Moreover the following formulas are used to compute the relative errors and root means square (RMS) errors.

$$L_2 = \left(\sum_{j=1}^{N} \left| u_j^{exact} - u_j^* \right|^2 \right)^{\frac{1}{2}} \quad (4.37)$$

$$L_\infty = max_{j=1}^{N} \left| u_j^{exact} - u_j^* \right| \quad (4.38)$$

$$RMS = \left(\sum_{i=1}^{N} \frac{e_i^2}{N} \right)^{\frac{1}{2}} \quad (4.39)$$

That u_j^* indicates a numerical solution for u at point j.

Example 4.1. *Consider a TE as follows:*

$$u_{tt} + 2\gamma u_t + \delta^2 u = u_{xx} + u_{yy} + f(x, y, t); \quad 0 \le x, y \le 1, t \ge 0 \quad (4.40)$$

With the following initial and Dirichlet boundary values

$$u(x, y, 0) = \sin x \sin y; u_t(x, y, 0) = 0 \quad (4.41)$$

$$\begin{cases} u(0, y, t) = 0; 0 \le y \le 1; x = 0 \\ u(1, y, t) = \cos t \sin(1) \sin y; \\ 0 \le y \le 1; x = 1 \\ u(x, 0, t) = 0; 0 \le x \le 1; y = 0 \\ u(x, 1, t) = \cos t \sin x \sin(1); \\ 0 \le x \le 1; y = 1 \end{cases} \quad (4.42)$$

the exact solution of this example in case $f(x, y, t) = 2(\cos t - \sin t) \sin x$ is given by $u(x, y, t) = \cos t \sin x \sin y$ [15].

The above example is solved with $\gamma = 1$ and $\delta = 1$. In Table 2, L_2 and L_∞ error norms are reported with $\Delta t = 0.01, h_x = h_y = 0.1$. Next, we take $\Delta t = 0.001$ and$h_x = h_y = 0.05$ and the

Table 2: Error norms of Example 4.1 with $\Delta t = 0.01$ and $h_x = h_y = 0.05$

T	The method in [?]		
	L_2	L_∞	Relative error
1	$9.722e-4$	$2.2746e-3$	$5.9762e-3$
2	$2.2877e-4$	$6.0818e-3$	$7.4720e-4$
3	$1.0926e-3$	$2.8706e-3$	$8.5019e-3$
7	$7.2867e-4$	$1.8781e-3$	$3.1572e-3$
T	present method		
	L_2	L_∞	Relative error
1	$1.2305e-5$	$3.2417e-6$	$2.5960e-6$
2	$1.3130e-6$	$4.2118e-6$	$3.9841e-6$
3	$4.1228e-5$	$5.7123e-5$	$3.5587e-5$
7	$2.2279e-6$	$5.5127e-5$	$5.3124e-6$

Table 3: Error norms of Example 4.1 with $\Delta t = 0.001$ and $h_x = h_y = 0.05$

T	The method in [15]		
	L_2	L_∞	R error
1	$9.8870e-5$	$2.4964e-4$	$6.2977e-4$
2	$1.2148e-4$	$3.2296e-4$	$1.0025e-3$
3	$3.7627e-5$	$9.9310e-5$	$1.3078e-4$
7	$7.2867e-4$	$1.8781e-3$	$3.1572e-3$
T	present method		
	L_2	L_∞	R error
1	$5.2140e-7$	$2.5127e-7$	$9.3184e-6$
2	$6.7448e-6$	$4.9972e-6$	$5.3420e-5$
3	$2.6025e-7$	$3.1782e-8$	$6.3327e-6$
7	$2.2279e-6$	$5.5127e-5$	$5.3124e-6$

L_2, L_∞ and relative errors at different time steps are reported in Table 3. In addition, a comparison of the present method and results in [15] is noted in Table 2 and Table 3 . One can see the results of the proposed method perform much better than the answers of [15] and are in a better agreement with the exact solution.

Example 4.2. *We consider the linear 3D hyperbolic TE*

$$u_{tt} + 2\gamma u_t + \delta^2 u = u_{xx} + u_{yy} + u_{zz} + f(x, y, z, t); \quad 0 \le x, y, z \le 1, t \ge 0 \quad (4.43)$$

where $f(x, y, z, t) = (\delta^2 - 4) \cos(t) - 2 \sin(t) \sinh(x) \sinh(y) \sinh(z)$ and $\gamma = 10$,

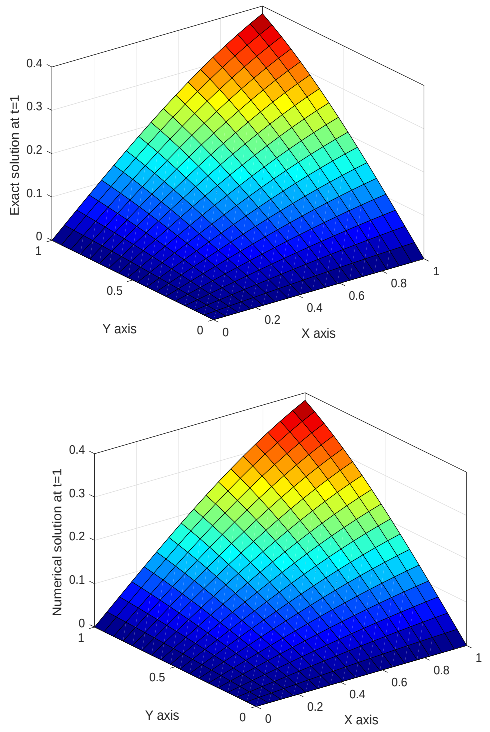


Figure 1: A comparison between the numerical and exact solution of Example 4.2 at $t = 1$

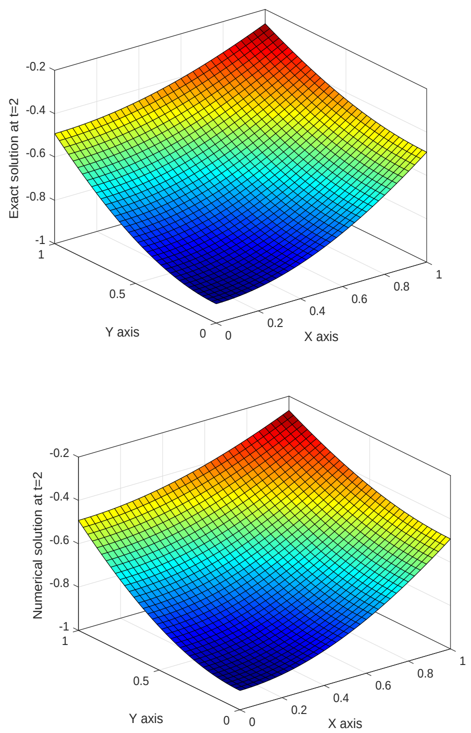
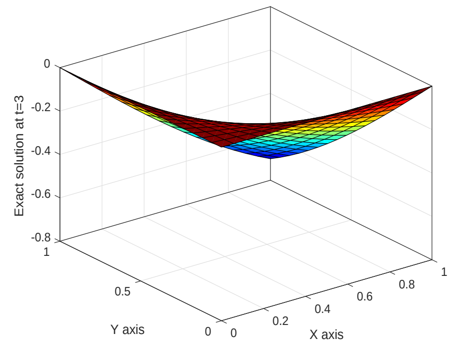


Figure 2: A comparison between the numerical and exact solution of Example 4.1 at $t = 2$



=2.5in, height=2in,keepaspectratio=false]fig3n

Figure 3: A comparison between the numerical and exact solution of Example 4.1 at $t = 3$

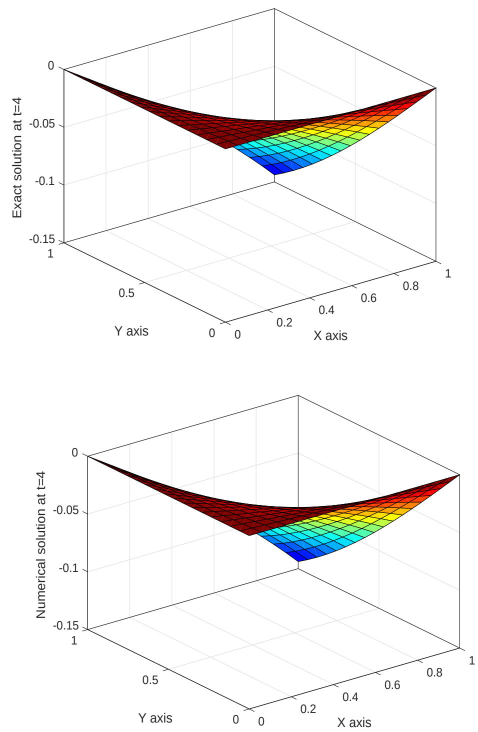


Figure 4: A comparison between the numerical and exact solution of Example 4.1 at $t = 4$

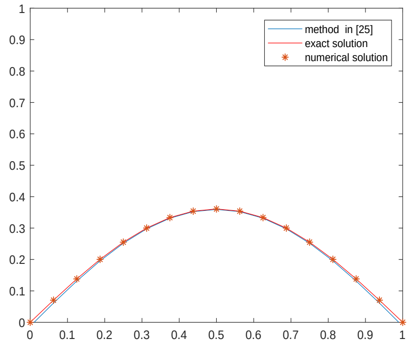


Figure 5: A comparison between the numerical and exact solution of Example 4.1 at $t = 1$ and the method in [15].

$\delta = 5$ with initial conditions

$$\begin{aligned} u(x, y, z, 0) &= \sinh(x)\sinh(y)\sinh(z) \\ u_t(x, y, z, 0) &= 0 \end{aligned} \tag{4.44}$$

The exact solution is given by $u(x, y, z, t) = \cos(t)\sinh(x)\sinh(y)\sinh(z)$ Table 4 and Table 5 show a comparison between the method in [19] with the present method in terms of relative and L_2, L_∞ error norms at $h_x = h_y = \frac{1}{10}, \frac{1}{16}, \frac{1}{20}, \Delta t = 0.001$. The results indicate that the new method produces much better results in comparison with [19].

Errors	N=10	
	$T = 1$	$T = 2$
L_2	$5.7008e-4$	$5.0974e-4$
L_∞	$1.200e-3$	$8.6344e-4$
R error		
Errors	N=16	
	$T = 1$	$T = 2$
L_2	$1.4645e-4$	$5.9825e-5$
L_∞	$2.1278e-4$	$1.7305e-4$
R error	$2.4400e-4$	$1.2936e-4$
Errors	N=20	
	$T = 1$	$T = 2$
L_2	$1.3244e-4$	$5.5135e-5$
L_∞	$4.9567e-5$	$1.9493e-5$
R error		

Table 4: error norms of Example 4.2 at time $t = 1, t = 2$ and $\Delta t = 0.001$ with the proposed numerical method

Errors	N=10	
	$T = 1$	$T = 2$
L_2	$2.7301e-6$	$7.2481e-5$
L_∞	$4.8311e-5$	$2.2035e-6$
Errors	N=16	
	$T = 1$	$T = 2$
L_2	$2.9512e-7$	$1.1907e-5$
L_∞	$1.9417e-6$	$4.5129e-6$
Errors	N=20	
	$T = 1$	$T = 2$
L_2	$3.6493e-6$	$2.5700e-6$
L_∞	$2.6307e-6$	$5.2175e-6$

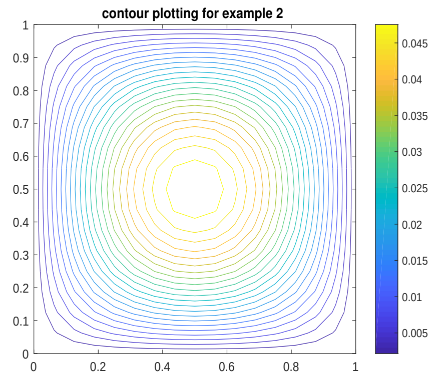


Figure 6: Contour plot of Example 4.2 with $dt = 0.001$

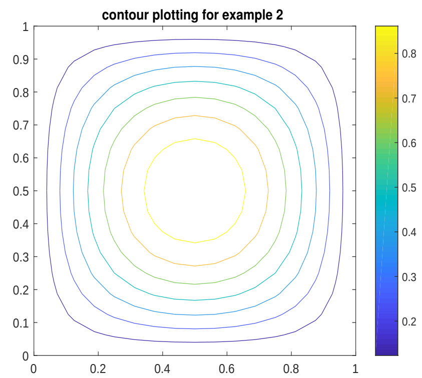


Figure 7: Contour plot of Example 4.2 with $dt = 0.01$

Example 4.3. *suppose the following 3D second-order hyperbolic TE*

$$u_{tt}+2u_t+u=u_{xx}+u_{yy}+u_{zz}; 0\leq x,y\leq 1, t\geq 0 \quad (4.45)$$

with $0\leq x,y,z\leq 1$ and the initial conditions

$$\begin{aligned} u(x,y,z,0)&=\sinh(x)\sinh(y)\sinh(z)\\ u_t(x,y,z,0)&=-\sinh(x)\sinh(y)\sinh(z) \end{aligned} \quad (4.46)$$

The exact solution is obtained by $u(x,y,z,t)=\exp^{-t}\sinh(x)\sinh(y)\sinh(z)$. Table 6 reports a comparison between the scheme in [14] with the present method in terms of L_2 and L_∞ errors at $h_x=h_y=h_z=\frac{1}{20}$ and $\Delta t=0.001$ at different times $t=5,10,15$.The results indicate that the new scheme produces much better results in comparison with [19].

Table 5: L_2 and L_∞ error norms of Example 4.3 at $\Delta t=0.001$ and $h_x=h_y=h_z=\frac{1}{20}$.

T	L_2	L_∞
5	3.2600×10^{-6}	2.7895×10^{-6}
10	1.6252×10^{-6}	9.13549×10^{-7}
2	2.2106×10^{-9}	5.7118×10^{-9}
T	L_2 in [14]	L_∞ in [14]
5	5.0130×10^{-4}	1.9019×10^{-5}
10	9.088×10^{-6}	7.900×10^{-6}
2	1.2490×10^{-8}	500740×10^{-8}

Example 4.4. *We suppose the second-order hyperbolic TE* (1.1)*, with* $0\leq x,y\leq 1,\gamma=\delta=1$ *and the initial condition*

$$\begin{aligned} u(x,y,0)&=\sin(\pi x)\sin(\pi y)\\ u_t(x,y,0)&=-\sin(\pi x)\sin(\pi y) \end{aligned} \quad (4.47)$$

With boundary conditions

$$\begin{aligned} &\tfrac{\partial u}{\partial x}(0,y,t)=-\pi\exp^{-t}\sin(\pi y); 0\leq y\leq 1; x=0\\ &\tfrac{\partial u}{\partial x}(x,1,t)=-\pi\exp^{-t}\sin(\pi x); 0\leq x\leq 1; y=1\\ &u(x,0,t)=0; 0\leq x\leq 1; y=0\\ &u(1,y,t)=0; 0\leq x\leq 1; x=1 \end{aligned} \quad (4.48)$$

Table 6: RMS and relative errors of Example 4.4 and a comparison of relative errors with other numerical methods

T	present method	method in [16]
0.5	5.2512×10^{-7}	1.00788×10^{-4}
1	1.2858×10^{-7}	9.13549×10^{-4}
2	6.9107×10^{-7}	1.19624×10^{-4}
3	3.9943×10^{-8}	1.32672×10^{-5}
5	2.7133×10^{-7}	1.32777×10^{-5}
10	1.3561×10^{-8}	3.35744×10^{-6}
T	method [17](ML)	method in [17]
0.5	7.040×10^{-5}	3.701×10^{-5}
1	9.088×10^{-5}	7.900×10^{-5}
2	4.820×10^{-4}	1.216×10^{-4}
3	1.400×10^{-3}	8.302×10^{-4}
5	...	...
10	...	...

The exact solution is achieved by $u(x,y,t)=-\exp^{-t}\sin(\pi x)\sin(\pi y); t\geq 0$.
Table 6 shows a comparison between the methods in [16] and [17] with the present method in terms of relative errors at $h_x=h_y=0.05$ and $\Delta t=0.001$. The results indicate the new method has a better accuracy in comparison with [16, 17] .

5 Conclusion

In this article, the numerical approximations of two and three-dimensional hyperbolic TE were computed by an exponential modified cubic B-spline DQM. During our study, we have used a DQM for discretizing both of the spatial derivatives and SSP-RK43 method for discretizing time. After discretizing we get an ODE system which solved by the SSP- RK43 method. The obtained numerical results were in good agreement with the exact solutions. A trivial outcome of this article is that accurate results will be achieved by using a little number of grid points instead of a lot of grid points, which not only can reduce the computations but also lead to saving in the com-

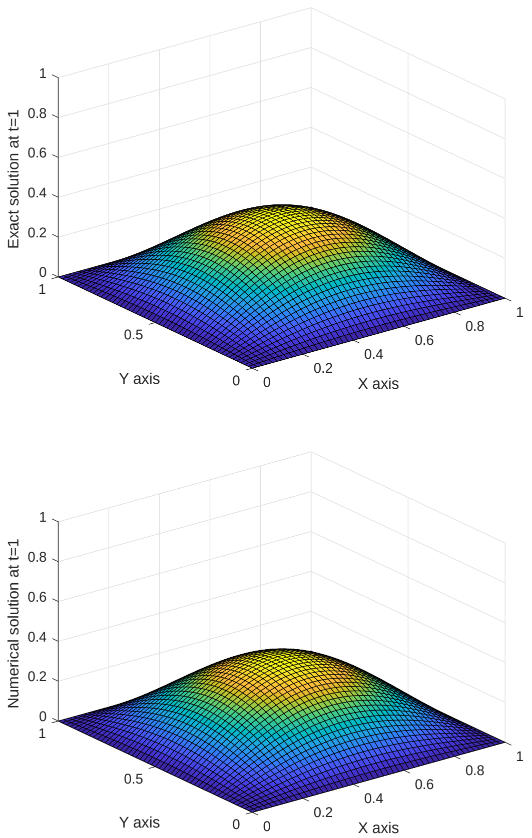


Figure 8: A comparison between the numerical and exact solution of Example 4.4 at $t = 1$.

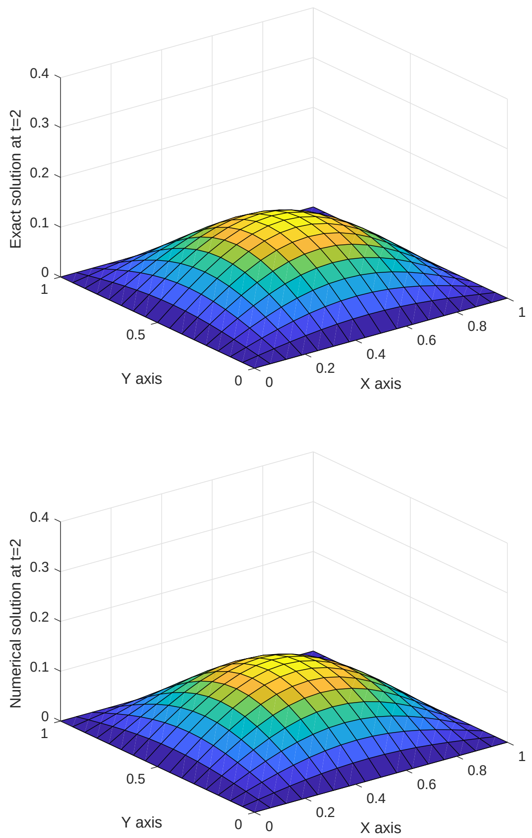


Figure 9: A comparison between the numerical and exact solution of Example 4.4 at $t = 2$.

puter running time.

References

[1] M. Lakestani, B. N .Saray, Numerical solution of telegraph equation using interpolating scaling functions, *Computers and Mathematics with Applications* 60 (2010) 19464-1972.

[2] Q. C. Nguyen, M. Piao, K. S. Hong, Multivariable adaptive control of the rewinding process of a roll-to-roll system governed by hyperbolic partial differential equations, *International Journal of Control, Automation and Systems* 5 (2018) 2177-2186.

[3] E. H. Doha, R. M. Hafez, Y. H. Youssri, Shifted Jacobi spectral-Galerkin method for solving hyperbolic partial differential equations, *Computers and Mathematics with Applications* 78 (2019) 889-904.

[4] Y. H. Youssri, R. M. Hafez, Exponential Jacobi spectral method for hyperbolic partial differential equations, *Mathematical Sciences* 13 (2019) 347-354.

[5] K. E. Bicer, S. Yalcinbas, A matrix approach to solving hyperbolic partial differential equations using Bernoulli polynomials, *Filomat* 30 (2016) 993-1000.

[6] G. Capdeville, Compact high-order numerical schemes for scalar hyperbolic partial differential equations, *Journal of Computational and Applied Mathematics* 363 (2020) 171-210.(2020)

[7] A. Ascanelli, A. Su, Random-field solutions to linear hyperbolic stochastic partial differential equations with variable coefficients, *Stochastic Processes and their Applications* 128 (2018) 2605-2641.

[8] R. Jiwari, Lagrange interpolation and modified cubic B-spline differential quadrature methods for solving hyperbolic partial differential equations with Dirichlet and Neumann boundary conditions, *Computer Physics Communications* 193 (2015) 55-65.

[9] C. Helzel, A Third-Order Accurate Wave Propagation Algorithm for Hyperbolic Partial Differential Equations, *Communications on Applied Mathematics and Computation* 1-25 (2020)

[10] A. Mohebbi and M.Dehghan , High order compact solution of the onespacedimensional linear hyperbolic equation, *Numerical Methods for Partial Differential Equations: An International Journal* 24 (2008) 1222-1235.

[11] R. Mohanty, New unconditionally stable difference schemes for the solution of multidimensional telegraphic equations, *International Journal of Computer Mathematics* 86 (2009) 2061-2071.

[12] B. Ola, A numerical procedure based on Hermite wavelets for two-dimensional hyperbolic telegraph equation, *Engineering with Computers* 34 (2018) 741-755.

[13] S. Singh, V. K. Patel, V. K. Singh, E.Tohidi, Application of Bernoulli matrix method for solving two-dimensional hyperbolic telegraph equations with Dirichlet boundary conditions, *Computers and Mathematics with Applications*, 75 (2018) 2280-2294.

[14] J. Lin , F. Chen, Y. Zhang, J. Lu, An accurate meshless collocation technique for solving two-dimensional hyperbolic telegraph equations in arbitrary domains, *Engineering Analysis with Boundary Elements* 108 (2019) 372-384.

[15] V. Devi, R. K. Maurya, S. Singh, V. K. Singh, Lagranges operational approach for the approximate solution of two-dimensional hyperbolic telegraph equation subject to Dirichlet boundary conditions, *Applied Mathematics and Computation* 367 (2020) 124-127.

[16] Z. Zhao , H. Li, Y. Liu, Analysis of a continuous Galerkin method with mesh modification for two-dimensional telegraph equation, *Computers and Mathematics with Applications* 79 (2020) 588-602.

[17] Y. Zhou , W. Qu, Y. Gu, H. Gao, A hybrid meshless method for the solution of the second order hyperbolic telegraph equation in two space dimensions, *Engineering Analysis with Boundary Elements* 115 (2020) 21-27.

[18] V. K. Srivastava, M. K. Awasthi, R. K. Chaurasia, Reduced differential transform method to solve two and three dimensional second order hyperbolic telegraph equations, *Journal of king Saud university-Engineering sciences* 29 (2017) 166-171.

[19] M. Kara, A Galerkin-like scheme to solve two-dimensional telegraph equation using collocation points in initial and boundary conditions, *Computers and Mathematics with Applications* 74 (2017) 3242-3249.

Ahmad Reza Haghighi is a Professor in the Department of Mathematics, Allameh Tabatabai University, Tehran, Iran. He completed his Ph. D degree in applied mathematics from Pune University, India. His research interest includes Bio-Mathematics, Computational Fluid Dynamics, Partial Differential Equations and Control.

Forouzan Rahimian was born in Shiraz, Iran. She received her B.Sc. degree in pure mathematics from Azad University of Shiraz branch. Also, she holds master degree in applied mathematics from Urmia University of Technology, Urmia, Iran. Her research areas include Numerical analysis, Partial Differential Equations.

Nasim Asghary is an associate professor in Mathematics Education in the Department of Science and congruent technologies, Islamic Azad Univercity, Central Tehran Branch,Tehran,Iran. Her research interests is Algebraic Thinking, Teacher training and Differential equestions.

Majid Roohi is a research assistant in the Department of Electrical and Computer Engineering (ECE) at Aarhus University, Denmark. His research interests include Control theory, Numerical analysis, Power electronics, Deep learning, and Big data.