Computational and Applied Mathematics (2022) 41:225
https://doi.org/10.1007/s40314-022-01927-x

A reliable and fast mesh-free solver for the telegraph equation

Neslişah İmamoğlu Karabaş[1] · Sıla Övgü Korkut[2] · Gurhan Gurarslan[3] · Gamze Tanoğlu[1]

Received: 9 December 2021 / Revised: 30 April 2022 / Accepted: 23 May 2022 /
Published online: 1 July 2022

Abstract

In the presented study, the hyperbolic telegraph equation is taken as the focus point. To solve such an equation, an accurate, reliable, and efficient method has been proposed. The developed method is mainly based on the combination of a kind of mesh-free method and an adaptive method. Multiquadric radial basis function mesh-free method is considered on spatial domain and the adaptive fifth-order Runge–Kutta method is used on time domain. The validity and the performance of the proposed method have been checked on several test problems. The approximate solutions are compared with the exact solution, it is shown that the proposed method has more preferable to the other methods in the literature.

Keywords Hyperbolic partial differential equations · Adaptive method · Mesh-free method · Telegraph equation

Mathematics Subject Classification 00A69 · 65M15 · 65M20 · 65M50

Communicated by Abdellah Hadjadj.

Sıla Övgü Korkut, Gurhan Gurarslan and Gamze Tanoğlu contributed equally to this work.

✉ Neslişah İmamoğlu Karabaş
neslisahimamoglu@gmail.com

Sıla Övgü Korkut
silaovgu@gmail.com

Gurhan Gurarslan
gurarslan@pau.edu.tr

Gamze Tanoğlu
gamzetanoglu@iyte.edu.tr

1 Department of Mathematics, İzmir Institute of Technology, Gülbahçe Campus, İzmir 35430, Turkey

2 Department of Engineering Sciences, İzmir Katip Celebi University, Balatcik Campus, İzmir 35620, Turkey

3 Department of Civil Engineering, Pamukkale University, Kinikli Campus, Denizli 20070, Turkey

1 Introduction

Many considerable phenomena in aerodynamic flows, flows of fluids and contaminants through porous media, atmospheric flows, signal-propagation, etc. are modeled by hyperbolic partial differential equations. Obtaining a solution for nonlinear hyperbolic partial differential equations is getting attractive than those in elliptic and parabolic ones. Due to the fact that any numerical solutions for linear systems can be adapted to the nonlinear ones, any contribution done for linear systems is so crucial.

One of the most popular hyperbolic equations is the telegraphic equation. Many remarkable studies have been done for obtaining stable methods for the numerical solution of the two-dimensional telegraphic equations for decades. The authors in Mohanty and Jain (2001) have presented an unconditionally stable method based on alternating direction implicit (ADI) scheme. Mohanty has proposed the idea of operator splitting in Mohanty (2004) while an implicit finite difference scheme combined by ADI strategy in Mohanty (2009) to obtain an unconditionally stable scheme. An element-free method based on least square approximation has been proposed in Cheng and Ge (2009). Compact finite difference method has been combined with an implicit collocation method in Dehghan and Mohebbi (2009), a meshless method using the radial basis functions combined with finite difference approximation in time in Dehghan and Shokri (2009), comparative work on meshless local weak–strong method, and meshless local Petrov–Galerkin method in which both methods are combined with Crank–Nicolson method in time in Dehghan and Ghesmati (2010). In Abbasbandy et al. (2014) Abbasbandy et al. have used both two classes of mesh-free methods based on the RBF, direct and indirect RBF collocation methods with their localized versions in space where the θ-weighted method used for time variable. In addition to these studies, one can see the applications of a differential quadrature method combined with RK4 in Jiwari et al. (2012), modified cubic B-splines combined with SSP-RK43 in Mittal and Bhatia (2014), modified extended B-splines combined with SSP-RK54 in Singh and Kumar (2018), and a spectral collocation method in Hafez (2018). More recently, the use of mesh-free methods is getting more attention among engineers and applied scientists for not only parabolic but also for hyperbolic equations such as the telegraph equation. For instance, in Lin et al. (2019), Crank–Nicolson scheme has been combined by the meshless method to solve the two-dimensional telegraph equation, Houbolt method has been combined with the meshless method in Zhou et al. (2020), a local differential quadrature method utilizing the radial basis functions has been combined by an explicit time integrator in Ahmad et al. (2020a), direct meshless method based on the isotropic radial basis function has been applied by treating time variable regularly during the whole solution process in Wang and Hou (2020), multi-wavelet Galerkin method has been presented in Jebreen et al. (2021).

All of above-mentioned investigations are generally focused on the variety of the spatial discretization. The novelty of the present study is essentially based on the time discretization technique. Unlike the aforementioned studies, the current study presents a new unconditionally stable algorithm based on the hybrid of meshless and adaptive Runge–Kutta method to solve the telegraph equation.

The main focus of the study is second-order hyperbolic equation such that

$$\begin{aligned}\frac{\partial^2 u}{\partial t^2} + 2\alpha(x, y)\frac{\partial u}{\partial t} + (\gamma(x, y))^2 u = \lambda_1(x, y)\frac{\partial^2 u}{\partial x^2} + \lambda_2(x, y)\frac{\partial^2 u}{\partial y^2} \\ + f(x, y, t), \quad \mathbf{x} = (x, y) \in \Omega, \ t > t_0,\end{aligned} \tag{1}$$

subjects to the following conditions

$$u(\mathbf{x},t)=g_D(.,t),\quad \text{on} \quad \partial\Omega,\quad t>t_0, \tag{2}$$

$$u(\mathbf{x},t_0)=\psi_0(\mathbf{x}),\quad \mathbf{x}\in\Omega, \tag{3}$$

$$\frac{\partial u}{\partial t}\,|_{(\mathbf{x},t_0)}=\psi_1(\mathbf{x}),\quad \mathbf{x}\in\Omega. \tag{4}$$

The rest of the study has been organized as follows: in Sect. 2 we give the reader the content in terms of brief descriptions of the applied methods in both time and space. The theoretical details are demonstrated in Sect. 3 by discussing the convergence of the numerical scheme. Section 4 is dedicated to present several numerical examples to highlight the efficiency and accuracy of the proposed numerical scheme by comparing to the existing methods in the literature.

2 Numerical scheme

The main purpose of this section is to describe the proposed numerical scheme by briefly touching upon the discretization techniques in both space and time variables. To this end, Eq. (1) is rearranged as a system of equations form with the help of changing variables such that

$$\begin{aligned}
&u_t(x,y,t)=w,\\
&w_t(x,y,t)=\lambda_1(x,y)\frac{\partial^2 u}{\partial x^2}+\lambda_2(x,y)\frac{\partial^2 u}{\partial y^2}-2\alpha(x,y)\,w-(\gamma(x,y))^2u+f(x,y,t),\\
&\quad u(\mathbf{x},t_0)=\psi_0(\mathbf{x}),\quad \mathbf{x}=(x,y)\in\Omega,\\
&\quad w(\mathbf{x},t_0)=\psi_1(\mathbf{x}),\quad \mathbf{x}=(x,y)\in\Omega.
\end{aligned} \tag{5}$$

For the sake of simplicity of expressions define the variable $\boldsymbol{Y}=[u,w]^T$, we have

$$\begin{aligned}
\boldsymbol{Y}_t(\mathbf{x},t)&=\begin{bmatrix}0 & 1\\ \lambda_1(\mathbf{x})\,\partial_x^2+\lambda_2(\mathbf{x})\,\partial_y^2 & 0\end{bmatrix}\boldsymbol{Y}(\mathbf{x},t)\\
&\quad+\begin{bmatrix}0 & 0\\ -(\gamma(\mathbf{x}))^2 & -2\alpha(\mathbf{x})\end{bmatrix}\boldsymbol{Y}(\mathbf{x},t)+\boldsymbol{F}(\mathbf{x},t),
\end{aligned} \tag{6}$$

$$\boldsymbol{Y}(\mathbf{x},t_0)=\boldsymbol{Y}_0(\mathbf{x}), \tag{7}$$

where $\boldsymbol{Y}_0(\mathbf{x})=[\psi_0(\mathbf{x}),\psi_1(\mathbf{x})]^T$, $\boldsymbol{F}(\mathbf{x},t)=[0,f(\mathbf{x},t)]$ and ∂_x^2 and ∂_y^2 denote the partial derivatives with respect to x and y, respectively. Due to the nature of numerical process, by employing the discretization for the spatial domain the Eq. (6) reduces to initial value problem which will be solved by an efficient, reliable and compatible method. The upcoming subsections are given to describe those methods, respectively.

2.1 MQ-RBF spatial discretization

The key point of the current study emphasizes the application of the adaptive Runge–Kutta method. However, to solve the system, the spatial discretization is required, as well. For this purpose, we use the meshless method with radial basis functions (RBFs) first introduced by Kansa (1990). RBFs are more preferable to many other discretizations due to its advantages.

For several applications of the meshless methods, the authors refer the interested reader to more recent studies such as (Siraj-ul-Islam 2015; Aziz et al. 2018; Siraj-ul-Islam and Haider 2018; Ahmad et al. 2020b; Seydaoğlu 2022).

Let $\phi : \mathbb{R}^2 \to \mathbb{R}$ be a univariate function which has infinity support. Due to the choice of multiquadric (MQ) which is one of the commonly-used RBFs, $\phi(\boldsymbol{r}) = (\boldsymbol{r}^2 + c^2)^{\beta/2}$ where β is an odd integer and c is the shape parameter. Here, $\boldsymbol{r}$ stands for the Euclidean norm in $\mathbb{R}^2$. The choice of shape parameters has a crucial role on the solution. Therefore, the shape parameter, c can vary whereas the constant β is fixed to 1 throughout the study. For the sake of integrity, let $U(\mathbf{x}, t)$ and $W(\mathbf{x}, t)$ denote the approximate solution of $u(\mathbf{x}, t)$ and $w(\mathbf{x}, t)$ such that

$$\begin{aligned} u(\mathbf{x}, t) \approx U(\mathbf{x}, t) &= \sum_{i}^{N_p} \zeta_{ij}(t)\phi_i(\boldsymbol{r}_j), \\ w(\mathbf{x}, t) \approx W(\mathbf{x}, t) &= \sum_{i}^{N_p} \eta_{ij}(t)\phi_i(\boldsymbol{r}_j), \end{aligned} \tag{8}$$

where N_p represents the number of collocation points. Notice that $\boldsymbol{r}_j = \left(\left\|\mathbf{x} - \mathbf{x}_j\right\|\right)$ where $\mathbf{x}_j = [j\Delta x, j\Delta y]^T$ such that $\Delta x = \frac{b-a}{N_x}$, and $\Delta y = \frac{d-c}{N_y}$ for $\Omega = [a, b]\times[c, d]$. Equation (8) can be expressed simply as follows:

$$\begin{aligned} U(\mathbf{x}, t) = \mathbf{\Phi}^T(\boldsymbol{r})\boldsymbol{\zeta}(t) \quad \nabla U(\mathbf{x}, t) = (\nabla\mathbf{\Phi}(\boldsymbol{r}))^T \boldsymbol{\zeta}(t) \quad \nabla^2 U(\mathbf{x}, t) = \left(\nabla^2\mathbf{\Phi}(\boldsymbol{r})\right)^T \boldsymbol{\zeta}(t), \\ W(\mathbf{x}, t) = \mathbf{\Phi}^T(\boldsymbol{r})\boldsymbol{\eta}(t) \quad \nabla W(\mathbf{x}, t) = (\nabla\mathbf{\Phi}(\boldsymbol{r}))^T \boldsymbol{\eta}(t) \quad \nabla^2 W(\mathbf{x}, t) = \left(\nabla^2\mathbf{\Phi}(\boldsymbol{r})\right)^T \boldsymbol{\eta}(t). \end{aligned} \tag{9}$$

where

$$\Phi(\mathbf{x}) = \begin{pmatrix} \phi_1(r_1) & \phi_1(r_2) & \ldots & \phi_1(r_{N_p}) \\ \phi_2(r_1) & \phi_2(r_2) & \ldots & \phi_2(r_{N_p}) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \phi_{N_p}(r_1) & \phi_{N_p}(r_2) & \ldots & \phi_{N_p}(r_{N_p}) \end{pmatrix}$$

In Eq. (9) ∇ stands for the gradient operator. Notice that at a fixed time, $t = t_n$, $\boldsymbol{\zeta}(t) = \boldsymbol{\zeta}(t_n) = \boldsymbol{\zeta}^n$, and $\boldsymbol{\eta}(t) = \boldsymbol{\eta}(t_n) = \boldsymbol{\eta}^n$ which can be computed such that

$$\boldsymbol{\zeta}^n = \left(\mathbf{\Phi}^T(\mathbf{x})\right)^{-1} U(x, t_n), \qquad \boldsymbol{\eta}^n = \left(\mathbf{\Phi}^T(\mathbf{x})\right)^{-1} W(x, t_n).$$

Using the notation $\nabla_{\mathbf{x}}^k$ to denote the kth gradient of the function for $k = 0, 1, 2$ with the property $\nabla^0 = I$ where I denotes the identity matrix, one can be expressed

$$\begin{aligned} \nabla_{\mathbf{x}}^k U(\mathbf{x}, t_n) &= \underbrace{\left(\nabla_{\mathbf{x}}^k \mathbf{\Phi}(\mathbf{x})\right)^T \left(\mathbf{\Phi}^T(\mathbf{x})\right)^{-1}}_{D^k(\mathbf{x})} U(\mathbf{x}, t_n), \\ \nabla_{\mathbf{x}}^k W(\mathbf{x}, t_n) &= \underbrace{\left(\nabla_{\mathbf{x}}^k \mathbf{\Phi}(\mathbf{x})\right)^T \left(\mathbf{\Phi}^T(\mathbf{x})\right)^{-1}}_{D^k(\mathbf{x})} W(\mathbf{x}, t_n). \end{aligned} \tag{10}$$

It is crucial to emphasize that the size of $D^k(\mathbf{x})$ for $k = 0, 1, 2$ is $N_x N_y \times N_x N_y$ due to $\mathbf{x} = (x, y) \in \mathbb{R}^2$. This section is finalized by rewriting Eq. (6) as follows:

$$\begin{aligned} \boldsymbol{Y}_t(\mathbf{x}, t) = {} & \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ D^2(\mathbf{x}) & \mathbf{0} \end{bmatrix} \boldsymbol{Y}(\mathbf{x}, t) \\ & + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\mathrm{diag}\left((\gamma(\mathbf{x}))^2\right) & -\mathrm{diag}(2\alpha(\mathbf{x})) \end{bmatrix} \boldsymbol{Y}(\mathbf{x}, t) + \boldsymbol{F}(\mathbf{x}, t), \end{aligned} \tag{11}$$

$$\boldsymbol{Y}(\mathbf{x}, t_0) = \boldsymbol{Y}_0(\mathbf{x}), \tag{12}$$

where $D^2(\mathbf{x}) = \lambda_1(\mathbf{x}) \frac{\partial^2 D}{\partial x^2}(\mathbf{x}) + \lambda_2(\mathbf{x}) \frac{\partial^2 D}{\partial y^2}(\mathbf{x})$. Notice that $\mathbf{0}$ denotes the zero matrix.

2.2 Fifth-order adaptive Runge–Kutta time integrator

Once the meshless method implemented successfully on the spatial domain, the system of partial differential equations given in Eq. (6) is converted into a system of ordinary differential equations (ODEs) as in Eq. (11). The main purpose of the present section is to describe a fifth-order adaptive Runge–Kutta formula which is also known as DOPRI5, (Dormand and Prince 1980). Due to the advantages of DOPRI5, a reliable, accurate, and efficient solution has been proposed to solve such hyperbolic equations, approximately. Equation (11) can be rewritten clearly as follows:

$$\boldsymbol{Y}_t(\mathbf{x}, t) = G(\mathbf{x}, t, \boldsymbol{Y}), \tag{13}$$

$$\boldsymbol{Y}(\mathbf{x}, t_0) = \boldsymbol{Y}_0(\mathbf{x}), \tag{14}$$

where

$$G(\mathbf{x}, t, \boldsymbol{Y}) = \underbrace{\begin{bmatrix} 0 & I \\ D^2(\mathbf{x}) & 0 \end{bmatrix} \boldsymbol{Y}(\mathbf{x}, t) + \begin{bmatrix} 0 & 0 \\ -\mathrm{diag}\left((\gamma(\mathbf{x}))^2\right) & -\mathrm{diag}(2\alpha(\mathbf{x})) \end{bmatrix}}_{A} \boldsymbol{Y}(\mathbf{x}, t) + \boldsymbol{F}(\mathbf{x}, t).$$

Consider Eq. (13), the approximated solution via the DOPRI5 can be obtained as follows:

$$\begin{aligned} k_1 &= G(\mathbf{x}, t_n, Y_n) \\ k_s &= G\left(\mathbf{x},\ t_n + \omega_s \Delta t_n,\ Y_n + \Delta t_n \sum_{l=1}^{s-1} \varphi_{s,l} k_l\right), \quad s = 2, 3, \ldots, 7, \\ Y_{n+1} &= Y_n + \Delta t_n \sum_{s=1}^{7} \chi_s k_s, \end{aligned} \tag{15}$$

where n and s denote the time and stage indexes, respectively. Moreover, k_s stands for the approximated slope matrix, Δt_n is the adapted time step at $t = t_n$. Furthermore, the Butcher table, (Butcher 1964), can be seen in Table 1 which introduces the required coefficients ω_s, $\varphi_{s,l}$, and χ_s.

DOPRI5 is a kind of adaptive methods. The adaptivity of the method comes from controlling the error of the method with a tolerance at each time step. For the sake of intelligibility, the pseudocode is given in Algorithm 1.

As in Bahar and Gurarslan (2020), the ratio $\frac{\Delta t_{n+1}}{\Delta t_n}$ has been limited to [0.1, 10] and the tolerance has been accepted as 10^{-6}. Before ending this section, the advantages of the DOPRI5 can be summarized as follows:

Table 1 Butcher table for DOPRI5

ω	φ						
0	0						
$\frac{1}{5}$	$\frac{1}{5}$	0					
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$	0				
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$	0			
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$	0		
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$	0	
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
χ^T	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
$\hat{\chi}^T$	$\frac{5179}{57600}$	0	$\frac{7571}{166954}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Algorithm 1 Pseudocode for the DOPRI5

procedure DOPRI5($Y, tolerance, t_0, t_{\text{final}}$)
  Define ω_s, $\varphi_{s,l}$, and χ_s,
  $\mathbf{x} = (x, y)$ grid points, $t = t_0$,
  Initialize $n = 0$, Δt_0
  while $t < t_{\text{final}}$, **do**
    Calculate k_s, for $s = 1, 2, \ldots, 7$, ▷ in Eq. (15)
    Compute $Y = Y + \Delta t_p \sum_{s=1}^{7} \chi_s k_s$
    Define $e_{n+1} = \left\| \Delta t_n \sum_{s=1}^{7} \left(\chi_s - \hat{\chi}_s\right) \right\|_\infty$
    if $e_{n+1} > tolerance$ **then**
      Define $\Delta t_{n+1} = 0.9 \Delta t_n \left(\frac{tolerance}{e_{n+1}}\right)^{1/5}, \quad e_{n+1} \le tolerance$
    else
      Store $Y_{n+1} := Y$ and $n := n + 1$
    end if
  end while
  Assign $u = Y(1, :)$
end procedure

- the method is unconditionally stable since error propagation is always controlled by a user-defined tolerance,
- being a member of the family of Runge–Kutta method makes the DOPRI5 more reliable,
- DOPRI5 is an explicit scheme which makes the method easily adaptable for both linear and nonlinear equations,
- the coefficients of the DOPRI5 obtained by minimizing the error of the fifth-order solution which leads to the higher order of accuracy.

On the other hand, the DOPRI5 has theoretically seven stages which can be considered a disadvantage. However, the fact that the use of the last stage is evaluated at the same point as the first stage. This helps to reduce the stages to six per step, computationally.

3 Convergence results

The proposed method introduced in Sect. 2 undoubtedly gives an approximate solution. However, any numerical method will be analyzed by the concepts of consistency and stability to guarantee the method will approximate to the exact solution, eventually. The main focus of the current section is to give these theoretical results of the proposed method. To do this, the following required auxiliary theorems are given.

Theorem 1 (Schaback and Wendland 2006; Li and Chen 2008) *Let* $[\beta]$ *stand for the smallest integer greater than or equal to* β. *The multiquadrics,* $\phi(\boldsymbol{r}) = \left(\boldsymbol{r}^2 + c^2\right)^{\beta/2}$, $\beta > 0$ *where* β *is an odd number, are conditionally positive definite of order* $m \leq \left[\frac{\beta}{2}\right]$ *on* $\mathbb{R}$

Theorem 2 (Li and Chen 2008, Theorem 10.1) *Assume that* ϕ *is conditionally positive definite of order* m *on* $\Omega \subset \mathbb{R}^d$, *and that the set of points* $Z = \{z_1, \ldots, z_N\} \in \Omega$ *is* $\prod_{m-1}\left(\mathbb{R}^d\right)$ *unisolvent, i.e., the zero polynomial is the only polynomial from* $\prod_{m-1}\left(\mathbb{R}^d\right)$ *that vanishes on* $Z = \{z_1, \ldots, z_N\}$. *Then for any* $f \in \mathcal{C}(\Omega)$, *there is exactly one function such that*

$$I_{f,X}(z) = \sum_{j=1}^{N} \alpha_j \phi\left(\|z - z_{\boldsymbol{j}}\|\right) + p(z)$$

where a polynomial $p(z) \in \prod_{m-1}\left(\mathbb{R}^d\right)$ *provided that*

$$I_{f,X}(z_i) = f(z_i),$$

and

$$\sum_{j=1}^{N} \alpha_j q\left(\|z - z_j\|\right) = 0, \quad 1 \leq i \leq N, \quad \forall q \in \prod_{m-1}\left(\mathbb{R}^d\right).$$

Here, $\prod_{m-1}\left(\mathbb{R}^d\right)$ *denotes the* d*-variable polynomial of degree at most* m.

Theorem 3 (Schaback 1999) *The error bound of interpolation for the choice of multiquadric radial basis functions is defined by*

$$\left\|I_{f,X} - f\right\|_{L_2(\Omega)} \leq C e^{-\delta/h}, \delta > 0, \tag{16}$$

for some constant C *where* $h = \sup_{\mathbf{x}\in\Omega} \min_{z\in Z} \|\mathbf{x} - z\|_2$

Assumption 1 Let $\nabla^2_{\mathbf{x}} \approx \left(\nabla^2_{\mathbf{x}} \boldsymbol{\Phi}(\mathbf{x})\right)^T \left(\boldsymbol{\Phi}^T(\mathbf{x})\right)^{-1}$. Due to the variability of the shape parameter we can write

$$\left\|\left(\nabla^2_{\mathbf{x}} \boldsymbol{\Phi}(\mathbf{x})\right)^T \left(\boldsymbol{\Phi}^T(\mathbf{x})\right)^{-1}\right\| \leq K_1(c),$$

where K_1 is a function of the shape parameter.

Our main references on the meshless methods and their analysis are Li and Chen (2008), Sarra and Kansa (2009) and references therein. The theorems stated above help to put the convergence result of approximation in the spatial domain. We, further, need to prove the convergence result of the proposed method considering the time integrator. To do so, we first give the required hypothesis.

Hypothesis 1 Let $\boldsymbol{F}(\mathbf{x}, t) \in \mathcal{C}(\Omega) \cap \mathcal{C}^2([0, t_{\text{final}}])$ such that

$$\left\|\frac{\partial^i \boldsymbol{F}}{\partial t^i}(\mathbf{x}, t_n) - \frac{\partial^i \boldsymbol{F}}{\partial t^i}(\mathbf{x}, t_{n-1})\right\| \leq Q_i, \quad i = 0, 1, \text{ and } \Delta t_{n-1}, \; n = 1, 2, \ldots, N_t.$$

With the help of Remark 1, $G(\mathbf{x}, t, \boldsymbol{Y})$ satisfies the Lipschitz property such that

$$\|G(\mathbf{x}, t, \boldsymbol{Y}) - G(\mathbf{x}, t, \mathbf{Z})\| \leq K_1(c)\, \|\boldsymbol{Y} - \mathbf{Z}\| + Q_0 \Delta t_{n-1}, \tag{17}$$

and

$$\left\| \frac{\partial G}{\partial t}(\mathbf{x}, t, \boldsymbol{Y}) - \frac{\partial G}{\partial t}(\mathbf{x}, t, \boldsymbol{Z}) \right\| \leq K_1(c) \left\| \boldsymbol{Y} - \boldsymbol{Z} \right\| + Q_1 \Delta t_{n-1}, \tag{18}$$

for some constants Q_0 and Q_1.

Under the lights of all mentioned theorems, remark, and hypothesis, Theorem 4 states the local error bound of the proposed method. For the sake of clarity of the analysis, Eq. (1) is considered for homogeneous case, that is $\boldsymbol{F}(\mathbf{x}, t) \equiv 0$. In this case, throughout the analysis, the values of Q_0 and Q_1 are accepted as 0.

Theorem 4 *Suppose that Hypothesis 1 fulfilled. The local error bound of the DOPRI5 for the homogeneous case of Eq. (1) is*

$$\|\boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n\| \leq \|\boldsymbol{Y}_n - \boldsymbol{Y}_{n-1}\| + \mathcal{O}\left(\Delta t_{n-1}^{s-1}\right).$$

Proof Let $\Delta t_p = \sigma_p \Delta t,\ \sigma_p \in (0, 1)$, where $t_p = t_{p-1} + \sigma_p \Delta t$ for $p = 1, 2, \ldots, N_{\max}$, where $N_{\max}$ is the maximum iteration when the tolerance holds. Let Y_n denote the approximate solution of Eq. (13)–(14) at $t = t_n$. The local error bound is obtained via the standard procedure, that is, $\|Y_{n+1} - Y_n\|$ where

$$\boldsymbol{Y}_{n+1} = \boldsymbol{Y}_n + \Delta t_n \sum_{s=1}^{7} \chi_s \, k_s^n, \tag{19}$$

$$\boldsymbol{Y}_n = \boldsymbol{Y}_{n-1} + \Delta t_{n-1} \sum_{s=1}^{7} \chi_s \, k_s^{n-1}. \tag{20}$$

By means of triangle inequality after subtracting Eq. (20) from Eq. (19) and taking the norms, one can obtain

$$\|\boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n\| \leq \|\boldsymbol{Y}_n - \boldsymbol{Y}_{n-1}\| + \left\| \Delta t_n \sum_{s=1}^{7} \chi_s \, k_s^n - \Delta t_{n-1} \sum_{s=1}^{7} \chi_s \, k_s^{n-1} \right\|.$$

Using the relation of $\frac{\Delta t_n}{\Delta t_{n-1}} = \rho$ where $\rho \in [0.1, 10]$ one can be said that $\Delta t_n = \rho \Delta t_{n-1}$ which leads to

$$\|\boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n\| \leq \|\boldsymbol{Y}_n - \boldsymbol{Y}_{n-1}\| + \left\| \rho \Delta t_{n-1} \sum_{s=1}^{7} \chi_s \, k_s^n - \Delta t_{n-1} \sum_{s=1}^{7} \chi_s \, k_s^{n-1} \right\|. \tag{21}$$

By choosing a constant C_1 depending on the values of $\max\{1, \rho\}$ and the upper bound of $\sum_{s=1}^{7} \chi_s$ Eq. (21) can be reduced to Eq. (22)

$$\|\boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n\| \leq \|\boldsymbol{Y}_n - \boldsymbol{Y}_{n-1}\| + C_1 \Delta t_{n-1} \underbrace{\left\| k_s^n - k_s^{n-1} \right\|}_{\text{I}}. \tag{22}$$

Notice that I needs to describe. By virtue of Taylor expansion, k_s^n and k_s^{n-1} for $s = 2, 3, \ldots, 7$ can be written as follows:

$$k_s^n = G\left(\mathbf{x}, t_n, \boldsymbol{Y}_n\right) + \frac{\partial G}{\partial t}\left(\mathbf{x}, t_n, \boldsymbol{Y}_n\right) \omega_s \Delta t_n + \frac{\partial G}{\partial Y}\left(\mathbf{x}, t_n, \boldsymbol{Y}_n\right) \Delta t_n \sum_{j_1=1}^{s-1} \varphi_{s, j_1} k_{j_1},$$

$$
\begin{aligned}
k_s^{n-1} &= G\left(\mathbf{x}, t_{n-1}, \boldsymbol{Y}_{n-1}\right) + \frac{\partial G}{\partial t}\left(\mathbf{x}, t_{n-1}, \boldsymbol{Y}_{n-1}\right)\omega_s \Delta t_{n-1} \\
&\quad + \frac{\partial G}{\partial Y}\left(\mathbf{x}, t_{n-1}, \boldsymbol{Y}_{n-1}\right)\Delta t_{n-1}\sum_{j_1=1}^{s-1}\varphi_{s,j_1}k_{j_1}.
\end{aligned}
\tag{23}
$$

For the sake of understandability of notations, we use G^p, G_t^p, and G_Y^p to represent $G\left(\mathbf{x}, t_p, \boldsymbol{Y}_p\right), \frac{\partial G}{\partial t}\left(\mathbf{x}, t_p, \boldsymbol{Y}_p\right)$, and $\frac{\partial G}{\partial Y}\left(\mathbf{x}, t_p, \boldsymbol{Y}_p\right)$, respectively. Moreover, we define $\hbar_s^n$ to denote $G^n + G_t^n\omega_s\Delta t_n$. Substituting $k_{j_1},\ \ j_1 = 1, \ldots, s-1$ leads to nested summations. More precisely,

$$
\begin{aligned}
k_s^n &= \hbar_s^n + G_Y^n\Delta t_n\sum_{j_1=1}^{s-1}\varphi_{s,j_1}(\hbar_{j_1}^n + G_Y^n\Delta t_n\sum_{j_2=1}^{j_1-1}\varphi_{j_1,j_2}\Bigg(\hbar_{j_2}^n \\
&\quad + \cdots + G_Y^n\Delta t_n\sum_{j_5=1}^{j_4-1}\varphi_{j_4,j_5}\left(\hbar_{s-1}^n + G_Y^n\Delta t_n\varphi_{1,1}k_1\right)\Bigg)\ldots\Bigg).
\end{aligned}
\tag{24}
$$

It is noted that k_s^{n-1} can be obtained by writing $n-1$ instead of n in Eq. (24). The bound of $\left\|\hbar_s^n - \hbar_s^{n-1}\right\|$ is defined by

$$
\left\|\hbar_s^n - \hbar_s^{n-1}\right\| \le M_1\left\|\boldsymbol{Y_n} - \boldsymbol{Y_{n-1}}\right\|, \tag{25}
$$

where M_1 can be obtained by the values of $K_1(c)$, Δt_{n-1} and $\max_{1\le s\le 7}\omega_s$. After doing some tedious calculations through Eq. (25), ① can be obtained as follows:

$$
① = \left\|k_s^n - k_s^{n-1}\right\| \le M_1\left\|\boldsymbol{Y}_n - \boldsymbol{Y}_{n-1}\right\| + \mathcal{O}\left(\Delta t_{n-1}^{s-1}\right).
$$

Substituting ① in Eq. (22) one can be obtained that

$$
\left\|\boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n\right\| \le \left(1 + \Delta t_{n-1}M_1\right)\left\|\boldsymbol{Y}_n - \boldsymbol{Y}_{n-1}\right\| + \mathcal{O}\left(\Delta t_{n-1}^{s-1}\right), \tag{26}
$$

which concludes the proof. □

Theorem 4 guaranteed that there is no error propagation of the proposed method over time. In addition to this conclusion, the stability of the proposed method required for convergence is discussed in Theorem 5.

Theorem 5 *Let Y_n denote the approximate solution obtained by the proposed method at $t = t_n,\ \ n = 1, 2, \ldots, N_t$ where N_t stands for the final time step. For an appropriate choice of the shape parameter satisfying $K_1(c)\Delta t_n < 1$, the proposed method remains stable with the bound*

$$
\left\|\boldsymbol{Y}_n\right\| \le \kappa\left\|\boldsymbol{Y}_0\right\| \tag{27}
$$

where the constant κ depends on $K_1(c)$ and Δt_n

Proof Before starting the proof, it is important to note the essential requirement bound playing a crucial role on completing the current proof. Thus, we first describe, briefly, the upper bound of k_s^n defined generally in Eq. (24) such that

$$
\left\|k_s^n\right\| \le W_1K_1(c)\sum_{j=1}^{s-1}\left(\Delta t_nK_1(c)\right)^j\left\|\boldsymbol{Y}_n\right\| + W_2K_1(c)\sum_{j=1}^{s-1}\mathcal{O}\left(\Delta t_n^{j+1}\right). \tag{28}
$$

Here both W_1 and W_2 are some constants. Then, due to use of the induction technique to discuss the stability of the proposed method, we start with giving the bound of one-step solution as follows:

$$
\begin{aligned}
\|\boldsymbol{Y}_1\| &\leq \|\boldsymbol{Y}_0\| + \Delta t_n \sum_{s=1}^{7} \chi_s \left\|k_s^0\right\|, \\
&\leq W_1 K_1(c) \sum_{j=1}^{s-1} (\Delta t_n K_1(c))^j \|\boldsymbol{Y}_0\| + W_2 K_1(c) \sum_{j=1}^{s-1} \mathcal{O}\left(\Delta t_n^{j+1}\right),
\end{aligned} \tag{29}
$$

where Y_0 is the prescribed initial condition. One can be seen in Eq. (29) the possibility of instability can arise just from the amplification factor of the proposed method which is $W_1 K_1(c) \sum_{j=1}^{s-1} (\Delta t_n K_1(c))^j$. This means that the second summation of Eq. (29) can be negligible as $\Delta t_n \to 0$. As the process progresses inductively, we obtain

$$
\begin{aligned}
\|\boldsymbol{Y}_n\| &\leq \left(W_1 K_1(c) \sum_{j=1}^{s-1} (\Delta t_n K_1(c))^j \right)^n \|\boldsymbol{Y}_0\|, \\
&\leq \underbrace{\frac{1}{1 - K_1(c)\Delta t_{n-1}}}_{\kappa} \|\boldsymbol{Y}_0\|,
\end{aligned} \tag{30}
$$

provided that $\|K_1(c)\Delta t_{n-1}\| < 1$. □

Both Theorems 3 and 5 highlights the vital role of the shape parameter of the approximate solution. Even though the unconditional stability property of the DOPRI5, Theorem 5 has stated that the proposed method is stable when the correct choice of shape parameter satisfying $K_1(c)\Delta t_n < 1$ is chosen. However, it is important to emphasize that under an appropriate choice of c, the proposed method has a long-time behavior and an accurate solution.

4 Computational results

The current section is dedicated to testing the performance and accuracy of the proposed method for several benchmark problems. The considered numerical examples cover various cases, such as constant coefficient and variable coefficient hyperbolic telegraph equation in both homogeneous and inhomogeneous ones. Throughout the section, the errors are measured in both relative error (RE) and root mean square error (RMSE) sense as follows:

$$
\mathrm{RE} = \sqrt{\frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left\|u\left(x_i, y_j, t_{\text{final}}\right) - U\left(x_i, y_j, t_{\text{final}}\right)\right\|.^2}{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left\|u\left(x_i, y_j, t_{\text{final}}\right)\right\|^2}},
$$

$$
\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left\|u\left(x_i, y_j, t_{\text{final}}\right) - U\left(x_i, y_j, t_{\text{final}}\right)\right\|^2}{N_x N_y}},
$$

where $u(x, y, t)$ and $U(x, y, t)$ stand for the exact solution and numerical solution, respectively. Throughout this section, $\alpha(x, y)$, $\gamma(x, y)$, $\lambda_1(x, y)$, and $\lambda_2(x, y)$ are denoted by α, γ, λ_1, and λ_2, respectively, in case of constant values selections. All computations have

Table 2 Comparison of root mean square errors for Example 1

t_{final}	Proposed method	GA (Dehghan and Salehi 2012)	IMQ (Dehghan and Salehi 2012)	PDQM (Jiwari et al. 2012) $N_x = N_y = 20$	PRBF (Rostamy et al. 2017) $N_x = N_y = 20$
0.5	1.8803e−05	1.68748e−05	5.08565e−05	9.26110e−05	4.468e−06
1	9.1985e−06	1.37797e−05	6.16056e−06	8.03601e−05	1.358e−05
2	1.2709e−05	1.23889e−05	2.84682e−06	9.05079e−05	3.375e−05
3	1.7571e−05	1.04864e−05	6.68808e−06	8.89250e−05	5.305e−05
5	4.3141e−05	–	–	9.00659e−05	9.318e−05
10	9.4235e−05	–	–	8.98311e−05	–
20	2.2277e−04	–	–	–	–

been executed on Intel Core i7-6700HQ 2.60Ghz and 16GB of RAM and implemented via the MATLAB-2018b programming language.

Example 1 Firstly, the two-dimensional telegraph equation given in Eq. (1) over the square domain $\Omega = [0, 1] \times [0, 1]$ where $\alpha = \gamma = 1$ and $\lambda_1 = \lambda_2 = 1$. The exact solution of the equation is

$$u(x, y, t) = x^2 + y^2 + t.$$

The initial conditions are as

$$u(x, y, 0) = x^2 + y^2, \quad u_t(x, y, 0) = 1$$

where the boundary conditions are taken from the exact solution. One can be obtained by the use of chosen parameters that $f(x, y, t) = x^2 + y^2 + t - 2$.

For obtaining the tables of Example 1, the spatial domain has discretized into 400 grid points by taking $N_x = N_y = 20$ and the shape parameter, $c = 0.38$. Table 2 presents a comparison of the root mean square errors for the proposed method to both boundary knot method combined with analog equation method using Gaussian (GA), inverse multiquadric (IMQ) RBFs in Dehghan and Salehi (2012), polynomial differential quadrature method (PDQM) in Jiwari et al. (2012), and pseudospectral radial basis functions (PRBF) in Rostamy et al. (2017).

It can be seen from Table 2 that the proposed method may record better results than not only those given in Table 2 but also the references therein. Nevertheless, it is noted that our errors are not as good enough as IMQ in Dehghan and Salehi (2012). This does not mean that the proposed method is not preferable. Table 3 emphasizes the efficiency of the proposed method. In this context, the CPU times of the proposed method are compared to the values existing in the literature.

Table 3 is evidence that the proposed method returns a fast solution compared to those in the literature. This property makes the proposed method more attractive. Moreover, Table 4 emphasizes the efficiency of the proposed method for both less and more collocation points. The listed values of Table 4 are obtained for $t_{\text{final}} = 1$ with the shape parameter as $c = 0.3$ for various number of collocation points.

Furthermore, the physical compatibility of the proposed method is illustrated in Fig. 1 for a finer mesh where $N_x = N_y = 25$ and the shape parameter $c = 0.3$ by comparing with the exact solution at $t = t_{\text{final}} = 3$ for $\alpha = \beta = 1$.

Table 3 Comparison of CPU times in second for Example 1

t_{final}	Proposed method	GA (Dehghan and Salehi 2012)	IMQ (Dehghan and Salehi 2012)	PDQM (Jiwari et al. 2012) $N_x = N_y = 20$	PRBF (Rostamy et al. 2017) $N_x = N_y = 20$
0.5	0.1	7.7	57.9	5	1.7
1	0.13	8.1	59.8	9	3.4
2	0.21	8.7	60.2	18	6.7
3	0.33	9.1	62.8	27	9.9
5	0.54	–	–	45	15.5
10	1.27	–	–	90	–
20	2.49	–	–	–	–

Table 4 Accuracy of Example 1 for fixed shape parameter $c = 0.3$ at $t_{\text{final}} = 1$

$N_x = Ny$	RMSE	RE
10	7.6789e−04	5.8831e−04
20	3.7866e−05	2.9178e−05
25	7.9944e−06	6.1668e−06

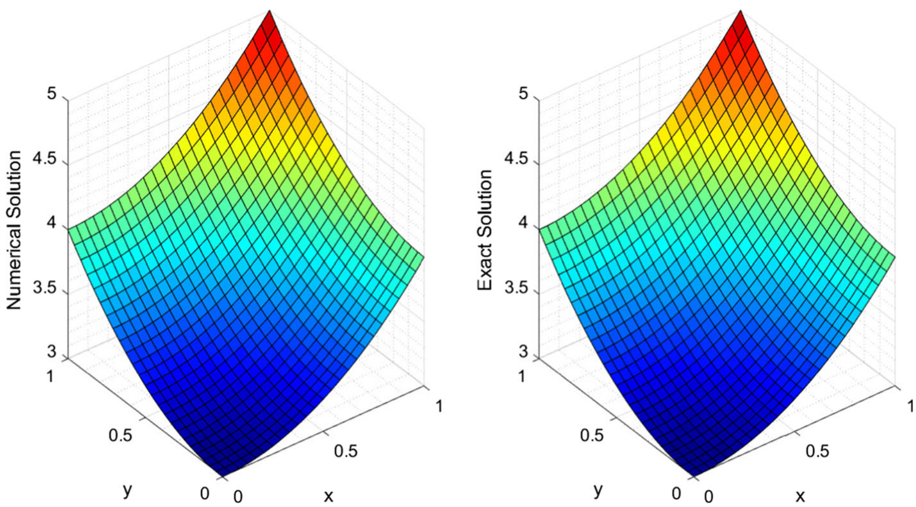

Fig. 1 Numerical and exact solutions of Example 1 at $t_{\text{final}} = 3$

Example 2 As the second problem, Eq. (1) is studied on $(x, y) \in \Omega = [0, 1]^2$ by taking $\lambda_1 = \lambda_2 = 1$ for the various choices of α and γ values. The initial and boundary conditions are chosen form the exact solution which defined by

$$u(x, y, t) = e^{-t} \sinh(x) \sinh(y)$$

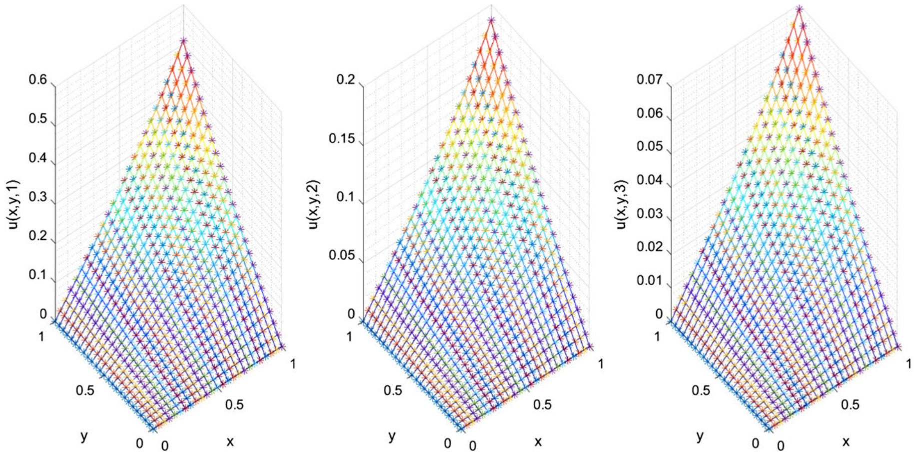

Fig. 2 Numerical and exact solution of Example 2 with $\alpha = \beta = 1$ at $t_{\text{final}} = 1, 2, 3$

The choices of different parameters can vary the $f(x, y, t)$. Throughout this example, the following cases have occurred:

$$f(x,y,t) = \begin{cases} 4e^{-t}\sinh(x)\sinh(y) & \alpha(x,y) = 10,\ \gamma(x,y) = 5; \\ -21e^{-t}\sinh(x)\sinh(y), & \alpha(x,y) = 10,\ \gamma(x,y) = 0; \\ 2e^{-t}\sinh(x)\sinh(y), & \alpha(x,y) = \gamma(x,y) = 1. \end{cases}$$

All computational results of Example 2 are recorded for $N_x = N_y = 25$ where the shape parameter is chosen as $c = 0.3$. Comparative work is presented in Table 5. The results obtained by the proposed method are compared to local meshless differential quadrature method (LMM) in Ahmad et al. (2020a), "isotropic" space-time radial basis function method (DMM1) in Wang and Hou (2020), (PRBF) in Rostamy et al. (2017), PDQM in Jiwari et al. (2012), and the modified B-spline differential quadrature method (MBDQ) in Mittal and Bhatia (2014).

It is crucial to emphasize that the LMM method in Ahmad et al. (2020a) also uses multiquadric RBF, whereas its errors are not as good enough as the proposed method. Besides, Tables 5 and 6 present the RMSE comparison for the proposed method to GA an IMQ in Dehghan and Salehi (2012).

Tables 5 and 6 declare that the proposed method is more accurate than the other numerical methods in the literature. The listed errors evidence how well the proposed method fits the exact values over time. Furthermore, the efficiency of the methods is demonstrated in Table 7. For this purpose, the CPU time of the algorithm has been compared to those available values in the literature.

It is important to underline that the reported run time for MBDQ in Mittal and Bhatia (2014) is chosen for the similar N_x and N_y choices to do a reliable comparison. It can be concluded from Table 5, 6 and 7 that the proposed method records better results in both accuracy and efficiency. All these results also highlight that the proposed method is preferable to the other numerical methods in the literature.

Before ending Example 2, Fig. 2 is illustrated to visualize that the proposed method is in good agreement with the exact solution at $\alpha = \gamma = 1$ for various choices of t_{final}.

Table 5 Comparison of RE and RMSE of several methods for the selected α and γ values in Example 2 at different times.

	t_{final}	Proposed method		LMM (Ahmad et al. 2020a) $N_x = N_y = 30$	DMM1 (Wang and Hou 2020) $N_x = N_y = 9$	PQDM (Jiwari et al. 2012) $N_x = N_y = 20$	MBDQ (Mittal and Bhatia 2014) $N_x = N_y = 20$	PRBF (Rostamy et al. 2017) $N_x = N_y = 20$
		RMSE	RE	RMSE	RMSE	RMSE	RE	RMSE
$\alpha = 10,\ \gamma = 5$	0.5	8.1368e−06	7.6102e−06	3.5821e−06	4.44e−07	3.30338e−05	1.1088e−04	3.005e−05
	1	4.9903e−06	7.6953e−06	3.2568e−06	4.38e−06	3.23359e−05	1.3266e−04	2.511e−05
	2	1.6018e−06	6.7140e−06	1.4538e−06	1.77e−05	3.11642e−05	3.1954e−04	1.098e−05
	3	4.6282e−07	5.2734e−06	–	–	3.06864e−05	1.3024e−04	4.218e−06
	5	9.0105e−09	7.5861e−06	–	–	3.04414e−05	1.4439e−04	5.858e−07
	10	2.9038e−09	3.6283e−05	–	–	3.04032e−05	–	–
$\alpha = 10,\ \gamma = 0$	0.5	9.6647e−06	9.0393e−06	4.4207e−06	8.80e−07	3.30746e−05	3.4675e−04	3.925e−05
	1	6.6628e−06	1.0274e−05	5.3563e−06	1.75e−06	3.33838e−05	3.9146e−04	4.350e−05
	2	2.3393e−06	9.8056e−06	3.8287e−06	1.86e−05	3.41356e−05	4.2739e−04	3.071e−05
	3	7.2028e−07	8.2070e−06	–	–	3.49446e−05	4.5140e−04	1.829e−05
	5	4.0096e−08	3.3758e−06	–	–	3.57123e−05	5.0758e−04	9.898e−06
	10	4.1308e−09	5.1615e−05	–	–	3.559175e−05	–	–

Table 6 Comparison of root mean square errors for Example 2 for $\alpha = \gamma = 1$ at various t_{final}

t_{final}	Proposed method	GA (Dehghan and Salehi 2012)	IMQ (Dehghan and Salehi 2012)
0.5	1.1530e−05	3.38485e−05	1.76044e−05
1	5.7258e−06	6.67377e−06	6.96266e−06
2	1.8546e−06	3.17489e−05	3.05566e−05
3	1.3345e−06	4.32162e−05	4.63057e−05

Example 3 As another benchmark problem, Eq. (1) is studied on $(x, y) \in \Omega = [0, 1]^2$ by the choice of $\alpha = \beta = 1$, and $\lambda_1 = \lambda_2 = 1$. The exact solution of Eq. (1) is given by

$$u(x, y, t) = \ln(1 + x + y + t),$$

where $f(x, y, t) = \frac{2}{1+x+y+t} + \ln(1 + x + y + t) + \frac{1}{(1+x+y+t)^2}$.

For a more reliable comparison, 100 grid points are chosen by taking $N_x = N_y = 10$ as it is done in the other studies. It is observed that the shape parameter can vary for various values of t_{final} in Example 3. In Table 8, the attained errors are compared with the results of DMM1 in Wang and Hou (2020), PQDM in Jiwari et al. (2012), the MBDQ in Mittal and Bhatia (2014), the meshless local weak–strong method via moving least square method (MLWS-MLS), and the meshless local Petrov–Galerkin via moving least square method (MLPG-MLS) in Dehghan and Ghesmati (2010).

As mentioned above, the obtained errors are recorded for $c = 1.2, 1.2, 1.1, 1.1, 0.9, 0.85$, and 0.8 where $t_{\text{final}} = 0.5, 1, 2, 3, 5, 10$, and 20, respectively. Table 8 declares that the proposed method achieves the best errors compared to other methods. In addition to Table 8, the reported available data for the maximum elapsed times to return the numerical result are listed in Table 9 in seconds.

Similar to the previous test problems, the recorded results in Table 9 promote the preferability of the proposed method. Besides all tables for Example 3, Fig. 3 indicates that the physical shape of the proposed method for finer mesh also fits well with the exact solution.

Example 4 All the above discussing examples are inhomogeneous. As another test problem, Eq. (1) is discussed by considering $\lambda_1 = \lambda_2 = 1$ for different α and γ values which lead to homogeneous and inhomogeneous cases. Initial conditions are given as follows:

$$u(x, y, 0) = \sin(\pi x)\sin(\pi y), \quad u_t(x, y, 0) = -\sin(\pi x)\sin(\pi y)$$

where

$$f(x, y, t) = \begin{cases} 2\pi^2 e^{-t} \sin\pi x \sin\pi y & \alpha = \gamma = 1; \\ 0, & \alpha = \pi^2 + 1, \;\; \gamma = 1. \end{cases}$$

For both choices the exact solution of Example 4 is defined as

$$u(x, y, t) = e^{-t} \sin(\pi x)\sin(\pi y)$$

To see the validity of the proposed method on the determined case, the errors are controlled in RE and RMSE sense. All the attained results for $\alpha = \gamma = 1$ are compared by the methods available in the literature and are presented in Table 10. The computational results are obtained for the choice of $N_x = N_y = 25$ where the shape parameter is chosen as $c = 0.25$.

Even though the computed errors are similar to those in the literature, comparing the elapsed times to return the results one can be seen in Table 10 that the proposed method is

Table 7 Comparison of elapsed time to return a numerical solution for Example 2 for different choices of α, and γ values at different times

t_{final}	$\alpha = 10\ \gamma = 5$				$\alpha = 10\ \gamma = 0$				$\alpha = 1\ \gamma = 1$		
	Proposed method	MBDQ (Mittal and Bhatia 2014) $N_x = N_y = 20$	PQDM (Jiwari et al. 2012) $N_x = N_y = 20$	PRBF (Rostamy et al. 2017) $N_x = N_y = 20$	Proposed method	MBDQ (Mittal and Bhatia 2014) $N_x = N_y = 20$	PQDM (Jiwari et al. 2012) $N_x = N_y = 20$	PRBF (Rostamy et al. 2017) $N_x = N_y = 20$	Proposed method	GA (Dehghan and Salehi 2012)	IMQ (Dehghan and Salehi 2012)
0.5	0.36	0.47	6	5.7	0.34	0.52	6	6	0.26	9.6	59.8
1	0.59	1.1	12	10	0.58	0.98	12	12	0.5	10.6	60.5
2	1.05	1.1	25	22	1.05	1.8	25	20	0.86	10.8	70.4
3	1.41	2.8	37	36	1.37	2.2	37	33	1.27	11.3	72.5
5	1.92	4.3	62	62	1.99	4.5	62	59	–	–	–
10	3.64	–	124	–	3.73	–	124	–	–	–	–

Table 8 Comparison of L_∞ and RMSE of several methods for $\alpha = 1,\ \gamma = 1$ for Example 3 at different times

t_{final}	Proposed method		DMM1 (Wang and Hou 2020) $N_x = N_y = 30$	PQDM (Jiwari et al. 2012) $N_x = N_y = 20$	MBDQ (Mittal and Bhatia 2014) $N_x = N_y = 20$	MLWS-MLS (Dehghan and Ghesmati 2010) $N_x = N_y = 20$	MLPG-MLS (Dehghan and Ghesmati 2010) $N_x = N_y = 20$
	RE	RMSE	RMSE	RMSE	RE	RE	RE
0.5	2.3439e−07	6.7364e−07	2.67e−06	4.91809e−05	1.1088e−03	7.939e−05	9.991e−05
1	4.7148e−07	1.6287e−06	6.06e−06	5.39567e−05	1.3266e−03	9.098e−05	7.198e−05
2	6.7545e−07	2.9523e−06	4.27e−05	4.95636e−05	3.1954e−04	8.705e−04	8.784e−04
3	1.9442e−06	9.8778e−06	2.34e−05	4.96391e−05	1.3024e−04	9.931e−04	4.801e−04
5	2.2886e−06	1.4072e−05	3.45e−05	4.42952e−05	8.4225e−05	4.703e−03	6.091e−04
10	3.3514e−06	2.6329e−05	–	4.32592e−05	2.9624e−05	7.302e−03	9.498e−04
20	3.8382e−06	3.7515e−05	–	–	–	–	–

Table 9 Comparison of CPU times of several methods for $\alpha = 1,\ \gamma = 1$ for Example 3 at different times

t_{final}	Proposed method	MBDQ (Mittal and Bhatia 2014) $N_x = N_y = 20$	PQDM (Jiwari et al. 2012) $N_x = N_y = 20$	MLWS-MLS (Dehghan and Ghesmati 2010) $N_x = N_y = 20$	MLPG-MLS (Dehghan and Ghesmati 2010) $N_x = N_y = 20$
0.5	0.03	0.5	6	9.2	21.0
1	0.04	1.1	11	12.9	36.2
2	0.04	2.0	22	25.7	49.1
3	0.05	2.8	32	38.1	66.8
5	0.07	7.0	54	49.8	82.0
10	0.1	9.6	108	62.0	97.3
20	0.17	–	–	–	–

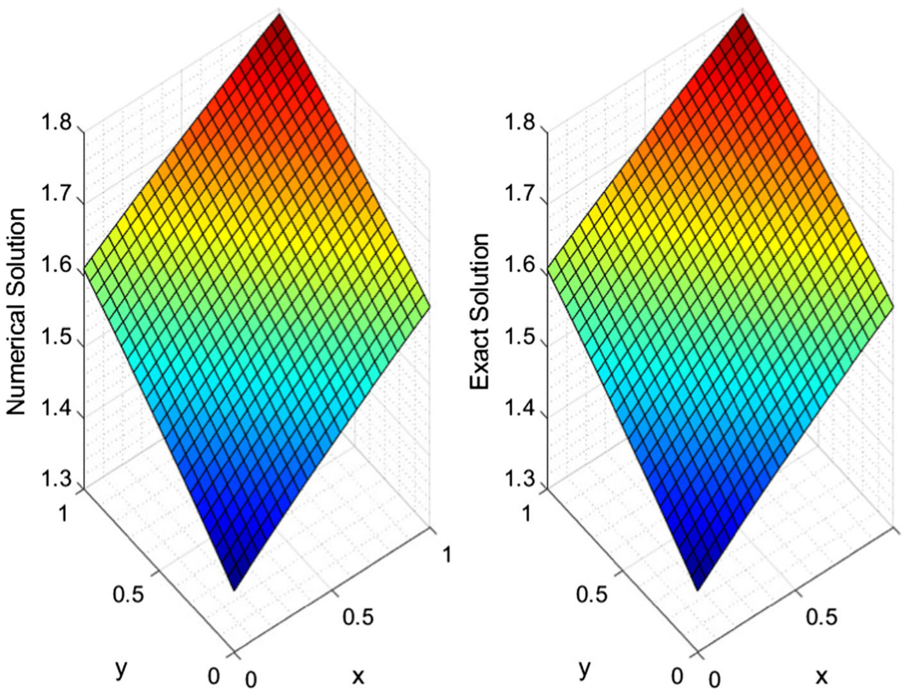

Fig. 3 Numerical and exact solutions of Example 3 at $t_{\text{final}} = 3$

more preferable to the methods existing in the literature. Moreover, Eq. (1) is also solved for the choice of parameters $\alpha = \pi^2 + 1$ and $\gamma = 1$, which leads to a homogeneous equation. To the best of the authors' knowledge, there are no existing results for the homogeneous ones in the literature. Therefore, the performance of the proposed method is checked by the exact solution by taking $N_x = N_y = 25$ and the shape parameter $c = 0.32$. Table 11 presents the CPU time in addition to errors described in both RE and RMSE sense (Fig. 4).

Table 10 Error and CPU times of Example 4 at different times

t_{final}	Proposed method			GA (Dehghan and Salehi 2012)		IMQ (Dehghan and Salehi 2012)		MLWS-MLS (Dehghan and Ghesmati 2010) $N_x = N_y = 20$		MLPG-MLS (Dehghan and Ghesmati 2010) $N_x = N_y = 20$	
	RE	RMSE	CPU-time	RMSE	CPU-time	RMSE	CPU-time	RE	CPU-time	RE	CPU-time
0.5	8.5359e−05	1.0945e−04	0.24	2.63185e−04	8.3	2.06466e−04	49.4	7.040e−05	8.8	3.701e−05	23.8
1	9.0439e−05	1.2845e−05	0.45	6.34934e−05	8.4	1.13532e−05	55.7	9.088e−05	11.5	7.900e−05	30.7
2	1.9773e−04	5.6573e−05	0.95	2.99709e−05	8.8	2.97719e−05	56.7	4.820e−04	24.0	1.216e−04	52.2
3	8.1936e−05	8.6241e−06	1.29	4.53846e−05	9.2	4.51602e−05	57.6	1.400e−03	37.7	8.302e−04	76.9

Table 11 Error and CPU-time of Example 4 for $\alpha = \pi^2 + 1,\ \gamma = 1$ at different times

t_{final}	RE	RMSE	CPU-time
0.5	2.8729e−05	3.6838e−05	0.27
1	4.4575e−05	3.4667e−05	0.47
2	7.2881e−05	2.0852e−05	0.84
3	9.4424e−05	9.9385e−06	1.1
5	1.2350e−04	1.7593e−06	1.62

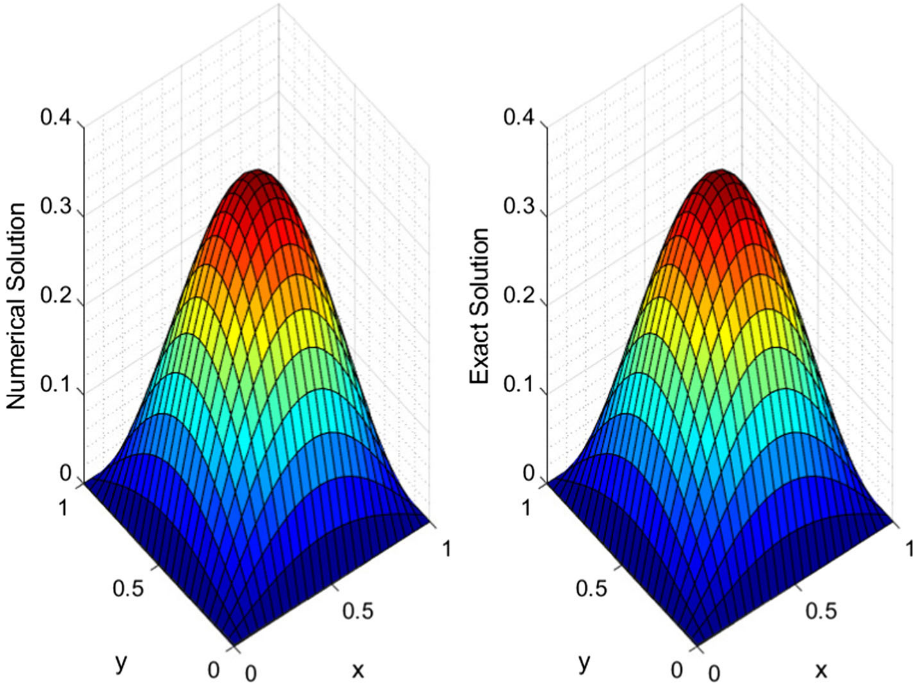

Fig. 4 Numerical and exact solutions of Example 4 at $t_{\text{final}} = 1$

Moreover, the physical behavior of the proposed method is also compared to the exact solution which is exhibited in Fig. 5 at various final times.

Example 5 As our last example, the variable coefficient telegraph equation is considered. To do so, in Eq. (1) $\alpha(x, y) = e^{x+y}$, $\gamma(x, y) = \sin x + y$. Moreover, $\lambda_1(x, y)$ and $\lambda_2(x, y)$ are taken as $\left(1 + x^2\right)$ and $\left(1 + y^2\right)$, respectively. That is, we have

$$u_{tt} + 2e^{x+y}u_t + \sin^2(x + y)u = \left(1+x^2\right)u_{xx} + \left(1 + y^2\right)u_{yy} + f(x, y, t),\ \ 0 < x, y < 1. \tag{31}$$

whose exact solution is defined by

$$u(x, y, t) = e^{-t}\sinh(x)\sinh(y)$$

Equation (31) is solved on $(x, y) \in \Omega = [0, 1]^2$ by discretizing the domain 100 grid points, that is $N_x = N_y = 10$. The obtained results are compared to those computed by operator

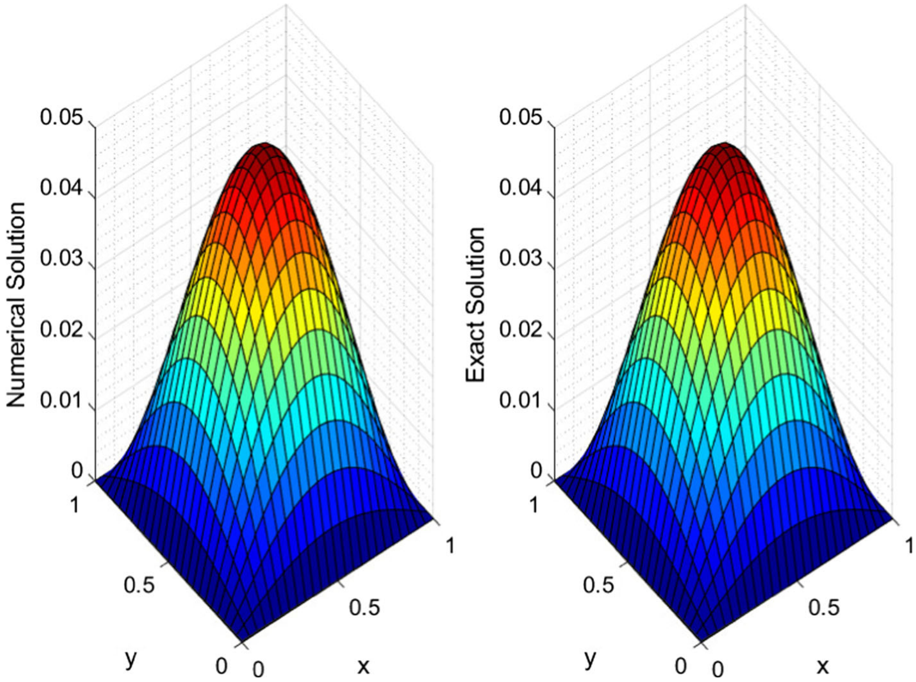

Fig. 5 Numerical and exact solutions of Example 4 at $t_{\text{final}} = 3$

Table 12 The RMSE of several methods and their CPU times for $\alpha = 1,\ \gamma = 1$ in Example 5 at different times

t_{final}	Proposed method		MSRBF (Dehghan and Shokri 2009) $dx = dy = 0.1$ $dt = 0.001$		OS (Mohanty 2004) $dx = dy = 1/64$	
	RMSE	CPU-time	RMSE	CPU-time	RMSE	CPU-time
0.5	2.0237e−06	0.05	1.2136e−05	1	–	–
1	1.6833e−06	0.05	4.1208e−06	1	0.6791e−04	–
2	2.0001e−06	0.06	1.5090e−06	2	0.2206e−04	–
3	1.4813e−06	0.07	5.5563e−07	3	–	–
5	2.3913e−05	0.09	2.0448e−07	3	–	–

splitting method combined with an unconditionally stable difference scheme (OS) in Mohanty (2004) and the meshless method with the help of thin plate spline radial basis functions (MSRBF) in Dehghan and Shokri (2009). All errors are listed in Table 12 with the CPU times of the methods.

The values of the proposed method in Table 12 are computed by taking the varied shape parameter as $c = 1.2, 1.2, 1, 0.85, 0.75$ for $t_{\text{final}} = 0.5, 1, 2, 3, 5$, respectively. One can be inferred from Table 12 that the proposed method can be preferable to the other methods for some values of t_{final} by taking into account both efficiency and accuracy of the solution. Figure 6 furthermore depicts the physical behavior of the proposed method agrees with the exact solution at final time, $t_{\text{final}} = 3$.

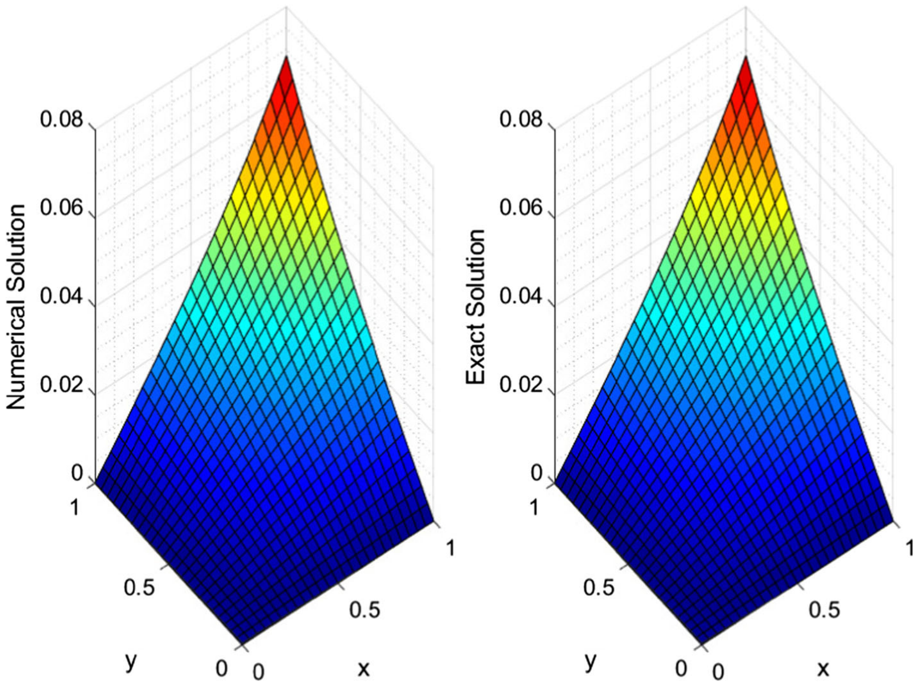

Fig. 6 Numerical and exact solutions of Example 5 for $N_x = N_y = 25$ with shape parameter $c = 0.25$ at $t_{\text{final}} = 3$

5 Conclusion

In this study, a reliable, accurate, and efficient method has been proposed for solving the hyperbolic telegraph equation. The equation solved by the proposed method is constructed by a combination of mesh-free RBF method and DOPRI5, one of the unconditionally stable methods. After introducing the proposed method, detailed convergence results have been studied with the concepts of consistency and stability. It has been shown theoretically that the shape parameter has a vital role in the approximate solutions. The study has been enriched by considering several benchmark problems. From a computational point of view, the crucial role of the shape parameter has been also emphasized upon various examples. It is reported that the shape parameter can be varied for not only different examples but also various values of t_{final} on the same example. All recorded results have been compared to those available in the literature. Furthermore, as mentioned in the theoretical part, the long-time behavior of the proposed method has also been confirmed. In a conclusion, all presented tables and figures support the preferability of the proposed method not only because of its accuracy but also because of its efficiency.

Acknowledgements The authors are grateful to the Editor-in-Chief and the anonymous referees for their valuable and constructive comments/suggestions to improve the original version.

References

Abbasbandy S, Roohani Ghehsareh H, Hashim I, Alsaedi A (2014) A comparison study of meshfree techniques for solving the two-dimensional linear hyperbolic telegraph equation. Eng Anal Bound Elem 47:10–20. https://doi.org/10.1016/j.enganabound.2014.04.006

Ahmad I, Ahmad H, Abouelregal AE, Thounthong P, Abdel-Aty M (2020) Numerical study of integer-order hyperbolic telegraph model arising in physical and related sciences. Eur Phys J Plus 135:759. https://doi.org/10.1140/epjp/s13360-020-00784-z

Ahmad I, Ahmad H, Thounthong P, Chu Y-M, Cesarano C (2020) Solution of multi-term time-fractional PDE models arising in mathematical biology and physics by local meshless method. Symmetry 12:1195. https://doi.org/10.3390/sym12071195

Aziz I, Siraj-ul-Islam Haider N (2018) Meshless and multi-resolution collocation techniques for steady state interface models. Int J Comput Methods 15(01):1750073. https://doi.org/10.1142/S0219876217500736

Bahar E, Gurarslan G (2020) B-spline method of lines for simulation of contaminant transport in groundwater. Water 12(6):1607. https://doi.org/10.1140/epjp/i2017-11529-2

Butcher JC (1964) Implicit Runge-Kutta processes. Math Comput 18:50–64. https://doi.org/10.1016/j.amc.2012.01.006

Cheng RJ, Ge HK (2009) Element-free Galerkin (EFG) method for a kind of two-dimensional linear hyperbolic equation. Chin Phys B 18(10):4059–4064. https://doi.org/10.1088/1674-1056/18/10/001

Dehghan M, Ghesmati A (2010) Combination of meshless local weak and strong (MLWS) forms to solve the two dimensional hyperbolic telegraph equation. Eng Anal Bound Elem 34(4):324–336. https://doi.org/10.1016/j.enganabound.2009.10.010

Dehghan M, Mohebbi A (2009) High order implicit collocation method for the solution of two-dimensional linear hyperbolic equation. Numer Methods Partial Differ Equ 25(1):232–243. https://doi.org/10.1002/num.20341

Dehghan M, Salehi R (2012) A method based on meshless approach for the numerical solution of the two-space dimensional hyperbolic telegraph equation. Math Methods Appl Sci 35(10):1220–1233. https://doi.org/10.1002/num.20357

Dehghan M, Shokri A (2009) A meshless method for numerical solution of a linear hyperbolic equation with variable coefficients in two space dimensions. Numer Methods Partial Differ Equ 25(2):494–506. https://doi.org/10.1002/num.20357

Dormand JR, Prince PJ (1980) A family of embedded Runge-Kutta formulae. J Comput Appl Math 6(1):19–26. https://doi.org/10.1016/0771-050X(80)90013-3

Hafez RM (2018) Numerical solution of linear and nonlinear hyperbolic telegraph type equations with variable coefficients using shifted Jacobi collocation method. Comput Appl Math 37:5253–5273. https://doi.org/10.1007/s40314-018-0635-1

Jebreen HB, Cano YC, Dassios I (2021) An efficient algorithm based on the multi-wavelet Galerkin method for telegraph equation. J AIMS Math 6(2):1296–1308. https://doi.org/10.3934/math.2021080

Jiwari R, Pandit S, Mittal RC (2012) A differential quadrature algorithm to solve the two dimensional linear hyperbolic telegraph equation with Dirichlet and Neumann boundary conditions. Appl Math Comput 218(13):7279–7294. https://doi.org/10.1016/j.amc.2012.01.006

Kansa EJ (1990) Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations. Comput Math Appl 19(8):147–161. https://doi.org/10.1016/0898-1221(90)90271-K

Li J, Chen Y-T (2008) Computational partial differential equations using MATLAB. CRC Press, Boca Raton

Lin J, Chen F, Zhang Y, Lu J (2019) An accurate meshless collocation technique for solving two-dimensional hyperbolic telegraph equations in arbitrary domains. Eng Anal Bound Elem 108:372–384. https://doi.org/10.1016/j.enganabound.2019.08.012

Mittal RC, Bhatia R (2014) A numerical study of two dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method. Appl Math Comput 244:976–997. https://doi.org/10.1016/j.amc.2014.07.060

Mohanty RK (2004) An operator splitting method for an unconditionally stable difference scheme for a linear hyperbolic equation with variable coefficients in two space dimensions. Appl Math Comput 152(3):799–806. https://doi.org/10.1016/S0096-3003(03)00595-2

Mohanty RK (2009) New unconditionally stable difference schemes for the solution of multi-dimensional telegraphic equations. Int J Comput Math 86(12):2061–2071. https://doi.org/10.1080/00207160801965271

Mohanty RK, Jain MK (2001) An unconditionally stable alternating direction implicit scheme for the two space dimensional linear hyperbolic equation. Numer Methods Partial Differ Equ 17(6):684–688. https://doi.org/10.1002/num.1034

Rostamy D, Emamjome M, Abbasbandy S (2017) A meshless technique based on the pseudospectral radial basis functions method for solving the two-dimensional hyperbolic telegraph equation. Eur Phys J Plus 132:263. https://doi.org/10.1140/epjp/i2017-11529-2

Sarra SA, Kansa EJ (2009) Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations. Adv Comput Mech 2(2):220

Schaback R (1999) Improved error bounds for scattered data interpolation by radial basis functions. Math Comput 68(225):201–216

Schaback R, Wendland H (2006) Kernel techniques: from machine learning to meshless methods. Acta Numerica 15:543–639. https://doi.org/10.1017/S0962492906270016

Seydaoğlu M (2022) A meshless two-stage scheme for the fifth-order dispersive models in the science of waves on water. Ocean Eng 250:111014. https://doi.org/10.1016/j.oceaneng.2022.111014

Singh BK, Kumar P (2018) An algorithm based on a new DQM with modified extended cubic B-splines for numerical study of two dimensional hyperbolic telegraph equation. Alexandria Eng J 57(1):175–191. https://doi.org/10.1016/j.aej.2016.11.009

Siraj-ul-Islam, Aziz I, Zaheer-ud-Din (2015) Meshless methods for multivariate highly oscillatory Fredholm integral equations. Eng Anal Bound Elem 53:100–112. https://doi.org/10.1016/j.enganabound.2014.12.007

Siraj-ul-Islam, Haider N, Aziz I (2018) Meshless and multi-resolution collocation techniques for parabolic interface models. Appl Math Comput 335:313–332. https://doi.org/10.1016/j.amc.2018.04.044

Wang F, Hou E (2020) A direct meshless method for solving two-dimensional second-order hyperbolic telegraph equations. J Math 2020:8832197. https://doi.org/10.1155/2020/8832197

Zhou Y, Qu W, Gu Y, Gao H (2020) A hybrid meshless method for the solution of the second order hyperbolic telegraph equation in two space dimensions. Eng Anal Bound Elem 115:21–27. https://doi.org/10.1016/j.enganabound.2020.02.015