East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.280118.100518

Vol. **9**, No. 1, pp. 195-211
February 2019

# An Unconditionally Stable Numerical Method for Two-Dimensional Hyperbolic Equations

Swarn Singh[1,*], Suruchi Singh[2] and Rajni Arora[3]

[1]*Department of Mathematics, Sri Venkateswara College, University of Delhi, New Delhi 110021, India.*
[2]*Department of Mathematics, Aditi Mahavidyalaya, University of Delhi, New Delhi 110039, India.*
[3]*Department of Mathematical Sciences, University of Delhi, New Delhi 110007, India.*

*Received 28 January 2018; Accepted (in revised version) 10 May 2018.*

**Abstract.** A collocation method based on exponential B-splines for two-dimensional second-order non-linear hyperbolic equations is studied. The initial equation is split into a system of coupled equations, each of which is transformed into a system of ordinary differential equations. The corresponding differential equations are solved by SSP-RK(2,2) method. It is shown that the method under consideration is unconditionally stable. Numerical experiments demonstrate its efficiency and accuracy.

**AMS subject classifications**: 65M60, 65M06, 65N30, 65N06

**Key words**: Collocation method, SSP-RK(2,2), telegraph equation, tri-diagonal solver, unconditional stability.

## 1. Introduction

We consider the second order two-dimensional non-linear hyperbolic equation

$$u_{tt} = u_{xx} + u_{yy} - 2\alpha u_t - \beta^2 u + g(x,y,t) + f(u), \quad a < x < b, \quad c < y < d, \quad t > 0 \tag{1.1}$$

with the initial conditions

$$u(x,y,0) = \phi(x,y), \quad u_t(x,y,0) = \psi(x,y), \quad a \le x \le b, \quad c \le y \le d \tag{1.2}$$

and the Dirichlet boundary conditions

$$\begin{aligned} u(a,y,t) &= f_1(y,t), \quad u(b,y,t) = f_2(y,t), \quad c < y < d, \quad t > 0, \\ u(x,c,t) &= f_3(x,t), \quad u(x,d,t) = f_4(x,t), \quad a < x < b, \quad t > 0. \end{aligned} \tag{1.3}$$

*Corresponding author. *Email addresses:* `ssingh@svc.ac.in` (S. Singh), `ssuruchi2005@yahoo.co.in` (S. Singh), `rrajni19@gmail.com` (R. Arora)

If $\alpha > 0$ and $\beta > 0$, the Eq. (1.1) becomes the telegraph equation and it is damped wave equation if $\alpha > 0$ and $\beta = 0$.

This equation is used in diffusion processes [10], image processing [28], vapor phase chromatography [9], dispersal in biological systems [1], and stochastic processes [18, 19]. A considerable attention has been paid to the solution of one-, two- and three-dimensional second order hyperbolic equations. In particular, Mohanty *et al.* [15–17] developed unconditionally stable implicit three level methods for one-dimensional second order hyperbolic problems and unconditionally stable implicit alternating direction methods for two- and three- dimensional hyperbolic problems. For two-dimensional linear telegraph equations, Bülbül *et al.* [4] and Jiwari *et al.* [12] developed Taylor matrix based methods and a differential quadrature method, respectively. Considering two-dimensional second-order hyperbolic equations, Ding and Zhang [7] proposed a fourth-order compact difference scheme, Dehghan and Ghesmati [5] studied meshless local weak and strong form methods and Dehghan and Mohebbi [6] considered a collocation method. In addition, Rashidinia *et al.* [20] and Mittal *et al.* [14] used cubic B-splines in one- and two-dimensional equations, respectively.

Here, we deal with an approximation method based on exponential B-splines. It was shown by McCartin [13] that exponential splines have a number of advantages — i.e. in computational aerodynamics they do not produce false oscillations of interpolants that appear in cubic splines methods. Nevertheless, exponential splines are rarely used in approximate solution of partial differential equations. Thus Ersoy and Idris [8] provided an exponential B-spline based algorithm for the Korteweg-de Vries equation, Singh *et al.* [21] used exponential B-splines in collocation method for one dimensional second order hyperbolic equation. Note that these splines have been introduced by Späth [23], who also considered their two-dimensional generalisation [24]. In this work an exponential B-spline based collocation method is applied to the second order two-dimensional non-linear hyperbolic equation (1.1). Decomposing the Eq. (1.1) into two equations, we discretise them in spatial directions and convert into the systems of ordinary differential equations. The systems obtained, are solved by SSP-RK(2,2) method — cf. Ref. [25].

The paper is organised as follows. In Section 2, we discuss a two-dimensional exponential B-spline based collocation method, with more details being provided in Section 3. Section 4 is concerned with the stability analysis. Five numerical examples are considered in Section 5 and our concluding remarks are in Section 6.

## 2. Two-Dimensional Exponential B-Spline Collocation Method

We consider the partitions

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b, \tag{2.1}$$

$$c = y_0 < y_1 < \cdots < y_{M-1} < y_M = d \tag{2.2}$$

of the domain $\Omega = \{(x, y) : a \leq x \leq b, c \leq y \leq d\}$, where $h_x = x_l - x_{l-1} = (b-a)/N$, $l = 1, 2, \cdots, N$ and $h_y = y_m - y_{m-1} = (d-c)/M$, $m = 1, 2, \cdots, M$. Moreover, we use the partition $t_j = jk$, $j = 0,1,\cdots,J$, $k > 0$ for time variable. Let $M_l(x)$ and $N_m(y)$ be the exponential B-splines defined by the partitions (2.1) and (2.2) respectively. Then, the two-dimensional exponential B-spline is defined as

$$B_{l,m}(x,y) = M_l(x)N_m(y), \quad l = 0,1,\cdots,N, \quad m = 0,1,\cdots,M. \tag{2.3}$$

The functions $B_{l,m}(x,y)$ are called tensor product B-splines [3]. The exponential B-splines $M_l(x)$ [24] related to the above partitions with added nodes $x_{-1}$ and $x_{N+1}$ are defined by

$$M_l(x) = \begin{cases} a_1\Big((x_{l-2}-x) - \dfrac{1}{p}(\sinh(p(x_{l-2}-x)))\Big), & x \in [x_{l-2}, x_{l-1}), \\ b_1 + b_2(x_l - x) + b_3\exp(p(x_l - x)) + b_4\exp(-p(x_l - x)), & x \in [x_{l-1}, x_l), \\ b_1 + b_2(x - x_l) + b_3\exp(p(x - x_l)) + b_4\exp(-p(x - x_l)), & x \in [x_l, x_{l+1}), \\ a_1\Big((x - x_{l+2}) - \dfrac{1}{p}(\sinh(p(x - x_{l+2})))\Big), & x \in [x_{l+1}, x_{l+2}), \\ 0, & \text{otherwise}, \end{cases} \tag{2.4}$$

where

$$\begin{aligned} a_1 &= \frac{p}{2(ph_xc - s)}, \quad b_1 = \frac{ph_xc}{(ph_xc - s)}, \\ b_2 &= \frac{p}{2}\left[\frac{c(c-1)+s^2}{(ph_xc - s)(1-c)}\right], \\ b_3 &= \frac{1}{4}\left[\frac{\exp(-ph_x)(1-c) + s(\exp(-ph_x)-1)}{(ph_xc - s)(1-c)}\right], \\ b_4 &= \frac{1}{4}\left[\frac{\exp(ph_x)(c-1) + s(\exp(ph_x)-1)}{(ph_xc - s)(1-c)}\right], \\ s &= \sinh(ph_x), \quad c = \cosh(ph_x), \end{aligned}$$

and parameter $p$ defines the shape of the corresponding spline functions. The exponential B-splines $N_m(y)$ are defined analogously but $h_x$ shall be replaced by $h_y$. According to the exponential B-spline collocation method [25], an approximate solution $U(x,y,t)$ of the problem (1.1)-(1.3) is sought in the form

$$U(x,y,t) = \sum_{l=-1}^{N+1}\sum_{m=-1}^{M+1} c_{l,m}(t)B_{l,m}(x,y), \tag{2.5}$$

where $c_{l,m}(t)$ are time dependent coefficients to be determined from collocation equations. The values of $M_l(x)$ and its first and second derivatives are presented in Table 1 and can be also used for evaluating the function $N_m(y)$ and its derivatives. Using these representations, we can evaluate $U(x,y,t)$ and its partial derivatives in spatial directions as follows:

$$\begin{aligned} U(x_l,y_m,t_j) = {} & p_1\big(m_1c^j_{l-1,m-1} + c^j_{l,m-1} + m_1c^j_{l+1,m-1}\big) + \big(m_1c^j_{l-1,m} + c^j_{l,m} + m_1c^j_{l+1,m}\big) \\ & + p_1\big(m_1c^j_{l-1,m+1} + c^j_{l,m+1} + m_1c^j_{l+1,m+1}\big), \end{aligned} \tag{2.6}$$

Table 1: Values of exponential B-spline $M_l(x)$ and its derivatives.

| $x$ | $x_{l-2}$ | $x_{l-1}$ | $x_l$ | $x_{l+1}$ | $x_{l+2}$ |
|---|---|---|---|---|---|
| $M_l(x)$ | 0 | $\frac{s-ph_x}{2(ph_xc-s)}$ | 1 | $\frac{s-ph_x}{2(ph_xc-s)}$ | 0 |
| $M_{x_l}(x)$ | 0 | $\frac{p(c-1)}{2(ph_xc-s)}$ | 0 | $-\frac{p(c-1)}{2(ph_xc-s)}$ | 0 |
| $M_{xx_l}(x)$ | 0 | $\frac{p^2s}{2(ph_xc-s)}$ | $-\frac{p^2s}{(ph_xc-s)}$ | $\frac{p^2s}{2(ph_xc-s)}$ | 0 |

$$\begin{aligned} U_x(x_l,y_m,t_j) &= m_2\big(p_1c^j_{l+1,m-1}+c^j_{l+1,m}+p_1c^j_{l+1,m+1}\big)-m_2\big(p_1c^j_{l-1,m-1}+c^j_{l-1,m} \\ &\quad +p_1c^j_{l-1,m+1}\big), \end{aligned} \tag{2.7}$$

$$\begin{aligned} U_{xx}(x_l,y_m,t_j) &= m_3p_1\big(c^j_{l-1,m-1}-2c^j_{l,m-1}+c^j_{l+1,m-1}\big)+m_3\big(c^j_{l-1,m}-2c^j_{l,m}+c^j_{l+1,m}\big) \\ &\quad +m_3p_1\big(c^j_{l-1,m+1}-2c^j_{l,m+1}+c^j_{l+1,m+1}\big), \end{aligned} \tag{2.8}$$

$$\begin{aligned} U_y(x_l,y_m,t_j) &= p_2\big(m_1c^j_{l-1,m+1}+c^j_{l,m+1}+m_1c^j_{l+1,m+1}\big)-p_2\big(m_1c^j_{l-1,m-1}+c^j_{l,m-1} \\ &\quad +m_1c^j_{l+1,m-1}\big), \end{aligned} \tag{2.9}$$

$$\begin{aligned} U_{yy}(x_l,y_m,t_j) &= p_3\big(m_1c^j_{l-1,m-1}+c^j_{l,m-1}+m_1c^j_{l+1,m-1}\big)-2p_3\big(m_1c^j_{l-1,m}+c^j_{l,m} \\ &\quad +m_1c^j_{l+1,m}\big)+p_3\big(m_1c^j_{l-1,m+1}+c^j_{l,m+1}+m_1c^j_{l+1,m+1}\big), \end{aligned} \tag{2.10}$$

$$\begin{aligned} U_t(x_l,y_m,t_j) &= p_1\big(m_1\dot{c}^j_{l-1,m-1}+\dot{c}^j_{l,m-1}+m_1\dot{c}^j_{l+1,m-1}\big)+\big(m_1\dot{c}^j_{l-1,m}+\dot{c}^j_{l,m}+m_1\dot{c}^j_{l+1,m}\big) \\ &\quad +p_1\big(m_1\dot{c}^j_{l-1,m+1}+\dot{c}^j_{l,m+1}+m_1\dot{c}^j_{l+1,m+1}\big), \end{aligned} \tag{2.11}$$

where $c^j_{l,m}=c_{l,m}(t_j)$ and

$$\begin{aligned} m_1 &= \frac{s-ph_x}{2(ph_xc-s)}, \quad m_2=\frac{p(c-1)}{2(ph_xc-s)}, \quad m_3=\frac{p^2s}{2(ph_xc-s)}, \\ p_1 &= \frac{s-ph_y}{2(ph_yc-s)}, \quad p_2=\frac{p(c-1)}{2(ph_yc-s)}, \quad p_3=\frac{p^2s}{2(ph_yc-s)}. \end{aligned}$$

## 3. Description of the Numerical Method

Writing $u^j_{l,m}$ for $u(x_l,y_m)$ at the time level $t_j$ we represent Eq. (1.1) as

$$u^j_{ttl,m}=u^j_{xxl,m}+u^j_{yyl,m}-2\alpha u^j_{tl,m}-\beta^2u^j_{l,m}+g^j_{l,m}+f\big(u^j_{l,m}\big). \tag{3.1}$$

It can be split into the system of two equations – viz.

$$\begin{aligned} u^j_{tl,m} &= v^j_{l,m}, \\ v^j_{tl,m} &= u^j_{xxl,m}+u^j_{yyl,m}-2\alpha u^j_{tl,m}-\beta^2u^j_{l,m}+g^j_{l,m}+f\big(u^j_{l,m}\big). \end{aligned} \tag{3.2}$$

Substituting (2.6), (2.8), (2.10) and (2.11) in the Eq. (3.2) yields

$$\begin{aligned}
v_{l,m}^{j} &= p_1\big(m_1\dot{c}_{l-1,m-1}^{j}+\dot{c}_{l,m-1}^{j}+m_1\dot{c}_{l+1,m-1}^{j}\big)+\big(m_1\dot{c}_{l-1,m}^{j}+\dot{c}_{l,m}^{j}+m_1\dot{c}_{l+1,m}^{j}\big)\\
&\quad +p_1\big(m_1\dot{c}_{l-1,m+1}^{j}+\dot{c}_{l,m+1}^{j}+m_1\dot{c}_{l+1,m+1}^{j}\big),\\
\dot{v}_{l,m}^{j} &= m_3p_1\big(c_{l-1,m-1}^{j}-2c_{l,m-1}^{j}+c_{l+1,m-1}^{j}\big)+m_3\big(c_{l-1,m}^{j}-2c_{l,m}^{j}+c_{l+1,m}^{j}\big)\\
&\quad +m_3p_1\big(c_{l-1,m+1}^{j}-2c_{l,m+1}^{j}+c_{l+1,m+1}^{j}\big)+p_3\big(m_1c_{l-1,m-1}^{j}+c_{l,m-1}^{j}+m_1c_{l+1,m-1}^{j}\big)\\
&\quad -2p_3\big(m_1c_{l-1,m}^{j}+c_{l,m}^{j}+m_1c_{l+1,m}^{j}\big)+p_3\big(m_1c_{l-1,m+1}^{j}+c_{l,m+1}^{j}+m_1c_{l+1,m+1}^{j}\big)\\
&\quad -2\alpha v_{l,m}^{j}-\beta^2\Big(p_1\big(m_1c_{l-1,m-1}^{j}+c_{l,m-1}^{j}+m_1c_{l+1,m-1}^{j}\big)\\
&\quad +\big(m_1c_{l-1,m}^{j}+c_{l,m}^{j}+m_1c_{l+1,m}^{j}\big)+p_1\big(m_1c_{l-1,m+1}^{j}+c_{l,m+1}^{j}+m_1c_{l+1,m+1}^{j}\big)\Big)+g_{l,m}^{j}\\
&\quad +f\Big(p_1\big(m_1c_{l-1,m-1}^{j}+c_{l,m-1}^{j}+m_1c_{l+1,m-1}^{j}\big)+\big(m_1c_{l-1,m}^{j}+c_{l,m}^{j}+m_1c_{l+1,m}^{j}\big)\\
&\quad +p_1\big(m_1c_{l-1,m+1}^{j}+c_{l,m+1}^{j}+m_1c_{l+1,m+1}^{j}\big)\Big),\quad l=0,\cdots,N,\quad m=0,\cdots,M.
\end{aligned}\tag{3.3}$$

Note that the system (3.3) consists of $(N+1)(M+1)$ equations with $(N+3)(M+3)$ unknowns. In order to eliminate additional unknowns, we redefine two-dimensional exponential B-spline functions accommodating the boundary conditions — viz. we rewrite the approximate solution as

$$U(x,y,t)=\sum_{l=0}^{N}\sum_{m=0}^{M}c_{l,m}(t)\tilde{B}_{l,m}(x,y)=\sum_{l=0}^{N}\sum_{m=0}^{M}c_{l,m}(t)\tilde{M}_l(x)\tilde{N}_m(y),\tag{3.4}$$

where $\tilde{B}_{l,m}(x,y)=\tilde{M}_l(x)\tilde{N}_m(y)$ with $\tilde{M}_l(x)$ and $\tilde{N}_m(y)$ defined by

$$\tilde{M}_l(x)=\begin{cases} M_0(x)+2M_{-1}(x) & \text{for}\quad l=0,\\ M_1(x)-M_{-1}(x) & \text{for}\quad l=1,\\ M_l(x) & \text{for}\quad l=2,\cdots,N-2,\\ M_{N-1}(x)-M_{N+1}(x) & \text{for}\quad l=N-1,\\ M_N(x)+2M_{N+1}(x) & \text{for}\quad l=N.\end{cases}\tag{3.5}$$

$$\tilde{N}_m(y)=\begin{cases} N_0(y)+2N_{-1}(y) & \text{for}\quad m=0,\\ N_1(y)-N_{-1}(y) & \text{for}\quad m=1,\\ N_m(y) & \text{for}\quad m=2,\cdots,M-2,\\ N_{M-1}(y)-N_{M+1}(y) & \text{for}\quad m=M-1,\\ N_M(y)+2N_{M+1}(y) & \text{for}\quad m=M.\end{cases}\tag{3.6}$$

The matrices arising from representations (3.4) are tri-diagonal and can be easily handled. The set of functions $\{\tilde{B}_{l,m}: l=0,1,\cdots,N; m=0,1,\cdots,M\}$ forms a basis for the vector space of all exponential splines defined over the domain $\Omega$. It follows from the collocation method and the Eqs. (3.4)-(3.6) that the approximations $U^j_{tl,m}$ of $u^j_{tl,m}$ can be represented as

$$U^j_{tl,m} = \sum_{l=0}^{N}\sum_{m=0}^{M} \dot{c}^j_{l,m}\tilde{B}_{l,m}(x,y). \tag{3.7}$$

Finally, using the values of $M_l(x)$, $N_m(y)$ and their derivatives and substituting (3.4)-(3.7) into the Eq. (3.2) we arrive at the following systems of ordinary differential equations

$$v^j_{l,m} = \begin{cases} (1+2m_1)(1+2p_1)\dot{c}^j_{l,m}, & l=0,N, \quad m=0,M, \\ \big(1+2m_1\big)\big(p_1\dot{c}^j_{l,m-1}+\dot{c}^j_{l,m}+p_1\dot{c}^j_{l,m+1}\big), & l=0,N, \quad m=1,\cdots,M-1, \\ \big(1+2p_1\big)\big(m_1\dot{c}^j_{l-1,m}+\dot{c}^j_{l,m}+m_1\dot{c}^j_{l+1,m}\big), & l=1,\cdots,N-1, \quad m=0,M, \\ p_1\big(m_1\dot{c}^j_{l-1,m-1}+\dot{c}^j_{l,m-1}+m_1\dot{c}^j_{l+1,m-1}\big)+\big(m_1\dot{c}^j_{l-1,m}+\dot{c}^j_{l,m}+m_1\dot{c}^j_{l+1,m}\big) \\ \quad +p_1\left(m_1\dot{c}^j_{l-1,m+1}+\dot{c}^j_{l,m+1}+m_1\dot{c}^j_{l+1,m+1}\right), \\ \quad l=1,\cdots,N-1, \quad m=1,\cdots,M-1, \end{cases} \tag{3.8}$$

and

$$\dot{v}^j_{l,m} = \begin{cases} -2\alpha v^j_{l,m}-\beta^2(1+2m_1)(1+2p_1)c^j_{l,m}+g^j_{l,m} \\ \quad +f\big((1+2m_1)(1+2p_1)c^j_{l,m}\big), \quad l=0,N, \quad m=0,M, \\ -2\alpha v^j_{l,m}-\beta^2\big(1+2m_1\big)\big(p_1c^j_{l,m-1}+c^j_{l,m}+p_1c^j_{l,m+1}\big)+g^j_{l,m} \\ \quad +f\left(\big(1+2m_1\big)\big(p_1c^j_{l,m-1}+c^j_{l,m}+p_1c^j_{l,m+1}\big)\right), \\ \quad l=0,N, \quad m=1,\cdots,M-1, \\ -2\alpha v^j_{l,m}-\beta^2\big(1+2p_1\big)\big(m_1c^j_{l-1,m}+c^j_{l,m}+m_1c^j_{l+1,m}\big)+g^j_{l,m} \\ \quad +f\left(\big(1+2p_1\big)\big(m_1c^j_{l-1,m}+c^j_{l,m}+m_1c^j_{l+1,m}\big)\right), \\ \quad l=1,\cdots,N-1, \quad m=0,M, \\ m_3p_1\big(c^j_{l-1,m-1}-2c^j_{l,m-1}+c^j_{l+1,m-1}\big)+m_3\big(c^j_{l-1,m}-2c^j_{l,m}+c^j_{l+1,m}\big) \\ \quad +m_3p_1\big(c^j_{l-1,m+1}-2c^j_{l,m+1}+c^j_{l+1,m+1}\big)+p_3\big(m_1c^j_{l-1,m-1}+c^j_{l,m-1} \\ \quad +m_1c^j_{l+1,m-1}\big)-2p_3\big(m_1c^j_{l-1,m}+c^j_{l,m}+m_1c^j_{l+1,m}\big)+p_3\big(m_1c^j_{l-1,m+1} \\ \quad +c^j_{l,m+1}+m_1c^j_{l+1,m+1}\big)-2\alpha v^j_{l,m}-\beta^2\left(p_1\big(m_1c^j_{l-1,m-1}+c^j_{l,m-1}+m_1c^j_{l+1,m-1}\big)\right. \\ \quad \left.+\left(m_1c^j_{l-1,m}+c^j_{l,m}+m_1c^j_{l+1,m}\right)+p_1\big(m_1c^j_{l-1,m+1}+c^j_{l,m+1}+m_1c^j_{l+1,m+1}\big)\right) \\ \quad +g^j_{l,m}+f\left(p_1\big(m_1c^j_{l-1,m-1}+c^j_{l,m-1}+m_1c^j_{l+1,m-1}\big)+\big(m_1c^j_{l-1,m}+c^j_{l,m}\right. \\ \quad \left.+m_1c^j_{l+1,m}\big)+p_1\big(m_1c^j_{l-1,m+1}+c^j_{l,m+1}+m_1c^j_{l+1,m+1}\big)\right), \\ \quad l=1,\cdots,N-1, \quad m=1,\cdots,M-1. \end{cases} \tag{3.9}$$

The matrix form of the Eqs. (3.8) and (3.9) is

$$\mathbf{A}\dot{\mathbf{C}}^j = \mathbf{V}^j, \quad j = 1, 2, \cdots, \tag{3.10}$$

$$\dot{\mathbf{V}}^j = \mathbf{G}^j, \quad j = 1, 2, \cdots, \tag{3.11}$$

where

$$\mathbf{A} = \begin{bmatrix} (1+2m_1)\mathbf{X_1} & & & & \mathbf{0} \\ \mathbf{X_2} & \mathbf{X_3} & \mathbf{X_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{X_2} & \mathbf{X_3} & \mathbf{X_2} \\ \mathbf{0} & & & & (1+2m_1)\mathbf{X_1} \end{bmatrix},$$

$$\mathbf{X_1} = \begin{bmatrix} (1+2p_1) & & & & \\ p_1 & 1 & p_1 & & \\ & \ddots & \ddots & \ddots & \\ & & p_1 & 1 & p_1 \\ & & & & (1+2p_1) \end{bmatrix},$$

$$\mathbf{X_2} = \begin{bmatrix} m_1(1+2p_1) & & & & \\ m_1p_1 & p_1 & m_1p_1 & & \\ & \ddots & \ddots & \ddots & \\ & & m_1p_1 & p_1 & m_1p_1 \\ & & & & m_1(1+2p_1) \end{bmatrix},$$

$$\mathbf{X_3} = \begin{bmatrix} (1+2p_1) & & & & \\ m_1 & 1 & m_1 & & \\ & \ddots & \ddots & \ddots & \\ & & m_1 & 1 & m_1 \\ & & & & (1+2p_1) \end{bmatrix},$$

$$\mathbf{G}^j = \left[G^j_{0,0}, G^j_{0,1}, \cdots, G^j_{0,M}, G^j_{1,0}, G^j_{1,1}, \cdots, G^j_{1,M}, \cdots\cdots, G^j_{N,0}, G^j_{N,1}, \ldots, G^j_{N,M}\right]^{\mathsf{T}}$$

the components of $\mathbf{G}^j$ are the right-hand sides in the Eq. (3.9) and "$\mathsf{T}$" denotes the transposition operation. Analogously

$$\mathbf{V}^j = \left[v^j_{0,0}, v^j_{0,1}, \cdots, v^j_{0,M}, v^j_{1,0}, v^j_{1,1}, \cdots, v^j_{1,M}, \cdots\cdots, v^j_{N,0}, v^j_{N,1}, \cdots, v^j_{N,M}\right]^{\mathsf{T}},$$

$$\mathbf{C}^j = \left[c^j_{0,0}, c^j_{0,1}, \cdots, c^j_{0,M}, c^j_{1,0}, c^j_{1,1}, \cdots, c^j_{1,M}, \cdots\cdots, c^j_{N,0}, c^j_{N,1}, \cdots, c^j_{N,M}\right]^{\mathsf{T}}.$$

To start computations, we need to choose initial vectors $\mathbf{V}^0$ and $\mathbf{C}^0$. They can be obtained from the initial condition (1.2) — viz.

$$u^0_{tl,m} = v^0_{l,m} = \psi_{l,m}, \quad l = 0, \cdots, N, \quad m = 0, \cdots, M, \tag{3.12}$$

$$u^0_{l,m} = \phi_{l,m}, \quad l = 0, \cdots, N, \quad m = 0, \cdots, M. \tag{3.13}$$

Thus we derive the approximation $U^0_{l,m}=\phi_{l,m}$ of $u^0_{l,m}$ in the form

$$U^0_{l,m}=\begin{cases}(1+2m_1)(1+2p_1)c^0_{l,m}, & l=0,N, \quad m=0,M,\\ (1+2m_1)\big(p_1c^0_{l,m-1}+c^0_{l,m}+p_1c^0_{l,m+1}\big), & l=0,N, \quad m=1,\cdots,M-1,\\ (1+2p_1)\big(m_1c^0_{l-1,m}+c^0_{l,m}+m_1c^0_{l+1,m}\big), & l=1,\cdots,N-1, \quad m=0,M,\\ p_1\big(m_1c^0_{l-1,m-1}+c^0_{l,m-1}+m_1c^0_{l+1,m-1}\big)+\big(m_1c^0_{l-1,m}+c^0_{l,m}+m_1c^0_{l+1,m}\big) & \\ \quad +p_1\big(m_1c^0_{l-1,m+1}+c^0_{l,m+1}+m_1c^0_{l+1,m+1}\big), & \\ \quad l=1,\cdots,N-1, \quad m=1,\cdots,M-1. & \end{cases} \tag{3.14}$$

The system of Eqs. (3.14) has either tri-diagonal structure or block tri-diagonal structure and it can be solved either by using tri-diagonal solver or split into the following system of equations with tri-diagonal matrices:

$$p_1c^*_{l,m-1}+c^*_{l,m}+p_1c^*_{l,m-1}=U^0_{l,m}, \quad l=1,\cdots,N-1, \quad m=1,\cdots,M-1, \tag{3.15}$$
$$c^*_{l,m}=m_1c^0_{l-1,m}+c^0_{l,m}+m_1c^0_{l+1,m}, \quad l=1,\cdots,N-1, \quad m=1,\cdots,M-1, \tag{3.16}$$

where $c^*_{l,m}$ are intermediate values and the corresponding boundary conditions can be derived from (3.16). We first solve the Eq. (3.15) for $c^*_{l,m}$ and then (3.16) for $c^0_{l,m}$.

The same splitting can be used to derive $\dot{\mathbf{C}}^j$ from the Eq. (3.10) for $j=1,2,\cdots$. Finally, $\mathbf{V}^j$ and $\mathbf{C}^j$ are found by two-step strong stability-preserving Runge-Kutta method of the second order (SSP-RK(2,2)).

## 4. Stability Analysis

To verify the stability of the method, we apply it to the following equations

$$\begin{aligned}&u^j_{ttl,m}=u^j_{xxl,m}+u^j_{yyl,m}-2\alpha u^j_{tl,m}-\beta^2u^j_{l,m}+g^j_{l,m},\\ &l,m=0,1,\cdots,N, \quad \beta>0\end{aligned}$$

with the initial and Dirichlet boundary conditions (1.2), (1.3). As the result, we obtain the Eqs. (3.10), (3.11) with $f\equiv 0$. Setting $h_x=h_y=h$ leads to $m_1=p_1, m_2=p_2, m_3=p_3$ and combining (3.10) and (3.11) yields

$$\mathscr{A}\dot{\mathscr{C}}^j=\mathscr{B}\mathscr{C}^j+\mathscr{F}, \tag{4.1}$$

where

$$\begin{aligned}&\mathscr{A}=\begin{bmatrix}\mathbf{A_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}, \quad \mathscr{B}=\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{P} & -2\alpha\mathbf{I}\end{bmatrix},\\ &\mathscr{C}^j=\begin{bmatrix}\mathbf{C}^j\\ \mathbf{V}^j\end{bmatrix}, \qquad \mathscr{F}^j=\begin{bmatrix}\mathbf{0}\\ \mathbf{g}^j\end{bmatrix},\\ &\mathbf{P}=-\beta^2\mathbf{A_1}+m_3\mathbf{A_2}+m_3\mathbf{A}_3,\end{aligned}$$

$$\mathbf{A_1} = \begin{bmatrix} (1+2m_1)\mathbf{X_1} & & & & \\ m_1\mathbf{X_1} & \mathbf{X_1} & m_1\mathbf{X_1} & & \\ & \ddots & \ddots & \ddots & \\ & & m_1\mathbf{X_1} & \mathbf{X_1} & m_1\mathbf{X_1} \\ & & & & (1+2m_1)\mathbf{X_1} \end{bmatrix},$$

$$\mathbf{A_2} = \begin{bmatrix} \mathbf{0} & & & \\ m_1\mathbf{X_2} & \mathbf{X_2} & m_1\mathbf{X_2} & \\ & \ddots & \ddots & \ddots \\ & & m_1\mathbf{X_2} & \mathbf{X_2} & m_1\mathbf{X_2} \\ & & & & \mathbf{0} \end{bmatrix},$$

$$\mathbf{A_3} = \begin{bmatrix} \mathbf{0} & & & \\ \mathbf{X_3} & -2\mathbf{X_3} & \mathbf{X_3} & \\ & \ddots & \ddots & \ddots \\ & & \mathbf{X_3} & -2\mathbf{X_3} & \mathbf{X_3} \\ & & & & \mathbf{0} \end{bmatrix},$$

$$\mathbf{X_1} = \begin{bmatrix} (1+2m_1) & & & & \\ m_1 & 1 & m_1 & & \\ & \ddots & \ddots & \ddots & \\ & & m_1 & 1 & m_1 \\ & & & & (1+2m_1) \end{bmatrix},$$

$$\mathbf{X_2} = \begin{bmatrix} 0 & & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & & 0 \end{bmatrix}, \quad \mathbf{X_3} = \begin{bmatrix} 0 & & & & \\ m_1 & 1 & m_1 & & \\ & \ddots & \ddots & \ddots & \\ & & m_1 & 1 & m_1 \\ & & & & 0 \end{bmatrix}.$$

The Taylor series expansions of $\sinh(ph)$ and $\cosh(ph)$ show that $0 < m_1 < 1/2$, $m_2 > 0$ and $m_3 > 0$ for all $p, h > 0$. Moreover, one can prove by mathematical induction that all principal minors of the real-valued symmetric matrix $\mathbf{X_1}$ are positive, so that all its eigenvalues are also positive. Moreover, $\mathbf{X_2}$ and $\mathbf{X_3}$ are, respectively, real negative and real positive semi-definite matrices. Therefore, they correspondingly have non-positive and non negative eigenvalues. Now, we refer to a result in [22, pp. 107]. The components of matrix $\mathbf{A_1}$ have a common set of $N+1$ linearly independent eigenvectors and if $\lambda$ is an eigenvalue of $\mathbf{X_1}$, then the eigenvalues of matrix $\mathbf{A_1}$ are given by the eigenvalues of the matrix

$$\begin{bmatrix} (1+2m_1)\lambda & & & & \\ m_1\lambda & \lambda & m_1\lambda & & \\ & \ddots & \ddots & \ddots & \\ & & m_1\lambda & \lambda & m_1\lambda \\ & & & & (1+2m_1)\lambda \end{bmatrix}. \tag{4.2}$$

Clearly, eigenvalues of the matrix (4.2) are positive and hence the eigenvalues of $\mathbf{A_1}$ are positive. Analogous considerations show that the eigenvalues of $\mathbf{A_2}$ and $\mathbf{A_3}$ are non-positive. It follows that the eigenvalues of matrix $\mathbf{P}$ are negative. Further, the matrix $\mathscr{A}$ is a strictly diagonally dominant, so it is invertible. Multiplying Eq. (4.1) by $\mathscr{A}^{-1}$, we obtain

$$\dot{\mathscr{C}}^j = \mathscr{A}^{-1}\mathscr{B}\mathscr{C}^j + \mathscr{A}^{-1}\mathscr{F}, \tag{4.3}$$

where

$$\mathscr{A}^{-1}\mathscr{B} = \begin{bmatrix} \mathbf{0} & \mathbf{A}_1^{-1} \\ \mathbf{P} & -2\alpha\mathbf{I} \end{bmatrix}.$$

In order to establish the stability of the method, we show that the eigenvalues of coefficient matrix $\mathscr{A}^{-1}\mathscr{B}$ have negative real part. If $\mathbf{\Lambda} = p + iq$ is an eigenvalue of $\mathscr{A}^{-1}\mathscr{B}$ and $[\mathbf{X}\,\mathbf{Y}]^{\mathsf{T}}$ is the corresponding eigenvector, then

$$\begin{bmatrix} \mathbf{0} & \mathbf{A}_1^{-1} \\ \mathbf{P} & -2\alpha\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \mathbf{\Lambda}\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}, \tag{4.4}$$

and, consequently,

$$\mathbf{P}\mathbf{A}_1^{-1}\mathbf{Y} = \mathbf{\Lambda}(\mathbf{\Lambda} + 2\alpha)\mathbf{Y}. \tag{4.5}$$

Hence, $\mathbf{\Lambda}(\mathbf{\Lambda} + 2\alpha)$ is an eigenvalue of $\mathbf{P}\mathbf{A}_1^{-1}$. Taking into account that $\mathbf{P}\mathbf{A}_1^{-1}$ has only real negative eigenvalues, we obtain that

$$q(p+\alpha) = 0 \quad \text{and} \quad p(p+2\alpha) - q^2 < 0.$$

This system has two sets of solutions — viz.

1. $q$ is an arbitrary real number and $p = -\alpha$;
2. $q = 0$ and $(p+\alpha)^2 < \alpha^2$.

Due to the assumption $\alpha > 0$, in either case we get $p < 0$. Hence, the real part of $\Lambda$ is negative. By Routh-Hurwitz criteria [11] we conclude that the proposed method is unconditionally stable.

## 5. Numerical Experiments

Now we want to test the accuracy and efficiency of the method above. In what follows, a discrete $l_2$- and the maximum-norm are used to evaluate the errors of the method — viz.

$$\mathrm{ER}(l_2) = \sqrt{h_x h_y \sum_{l=0}^{N}\sum_{m=0}^{M}(U_{l,m} - u_{l,m})^2}, \quad \mathrm{ER}(M) = \max_{l,m}\left|U_{l,m} - u_{l,m}\right|.$$

We also provide the graphs of numerical and exact solutions and compare the results with previous studies. Computations are carried out on a laptop with Intel Pentium processor, 2.0 GHz CPU and 2 GB RAM using in MATLAB 13 software.

**Example 5.1.** Consider the following problem

$$u_{tt} = u_{xx} + u_{yy} - 2u_t - u + 2(\cos(t) - \sin(t))\sin(x)\sin(y), \quad 0 \le x, y \le 1$$

subject to initial and boundary conditions

$$\begin{aligned}
&u(x,y,0) = \sin(x)\sin(y), \quad u_t(x,y,0) = 0, && 0 \le x, y \le 1,\\
&u(0,y,t) = 0, \quad u(1,y,t) = \cos(t)\sin(1)\sin(y), && 0 \le y \le 1, \quad t \ge 0,\\
&u(x,0,t) = 0, \quad u(x,1,t) = \cos(t)\sin(x)\sin(1), && 0 \le x \le 1, \quad t \ge 0.
\end{aligned}$$

The analytic solution of this problem is $u(x,y,t) = \cos(t)\sin(x)\sin(y)$.

Tables 2 and 3 show the errors $\mathrm{ER}(l_2)$ and $\mathrm{ER}(M)$ at different time levels and these results are better than those obtained in [14]. Fig. 1 demonstrates the numerical and analytic solutions at the time $t = 3$. It is clear that the numerical and analytic solutions are in good agreement with each other.

**Example 5.2.** Consider the following problem

$$u_{tt} = u_{xx} + u_{yy} - 2\alpha u_t - \beta^2 u + (-2\alpha + \beta^2 - 1)e^{-t}\sinh(x)\sinh(y), \quad 0 \le x, y \le 1$$

with initial and boundary conditions

$$\begin{aligned}
&u(x,y,0) = \sinh(x)\sinh(y), \quad u_t(x,y,0) = -\sinh(x)\sinh(y), && 0 \le x, y \le 1,\\
&u(0,y,t) = 0, \quad u(1,y,t) = e^{-t}\sinh(1)\sinh(y), && 0 \le y \le 1, \quad t \ge 0,\\
&u(x,0,t) = 0, \quad u(x,1,t) = e^{-t}\sinh(x)\sinh(1), && 0 \le x \le 1, \quad t \ge 0.
\end{aligned}$$

The analytic solution of this problem is

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

Table 4 shows the errors $\mathrm{ER}(l_2)$ and $\mathrm{ER}(M)$ at different time levels and the results are better than the corresponding results of Mittal and Bhatia [14]. We also compute the errors $\mathrm{ER}(l_2)$ and $\mathrm{ER}(M)$ at different time levels for the parameters $\alpha = 5, \beta = 5, h_x = 0.1, h_y = 0.05, k = 0.01, p = 0.1$ and display them in Table 5. Fig. 2 demonstrates numerical and analytic solutions at the time $t = 2$.

**Example 5.3.** Consider the following problem

$$\begin{aligned}
u_{tt} = u_{xx} + u_{yy} - 2\alpha u_t - \beta^2 u + (-3\cos(t) - 2\alpha\sin(t + \beta^2\cos(t))\\
\times \sinh(x)\sinh(y), \quad 0 \le x, y \le 1
\end{aligned}$$

with initial and boundary conditions

$$\begin{aligned}
&u(x,y,0) = \sinh(x)\sinh(y), \quad u_t(x,y,0) = 0, && 0 \le x, y \le 1,\\
&u(0,y,t) = 0, \quad u(1,y,t) = \cos(t)\sinh(1)\sinh(y), && 0 \le y \le 1, \quad t \ge 0,\\
&u(x,0,t) = 0, \quad u(x,1,t) = \cos(t)\sinh(x)\sinh(1), && 0 \le x \le 1, \quad t \ge 0.
\end{aligned}$$

The analytic solution of this problem is $u(x,y,t) = \cos(t)\sinh(x)\sinh(y)$.

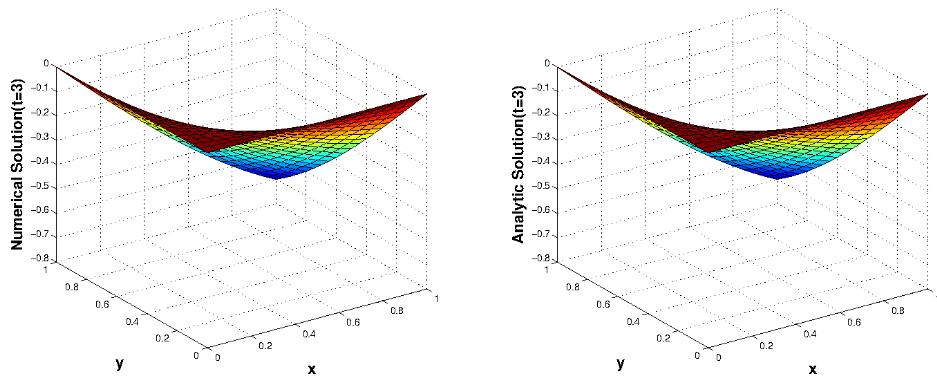


Figure 1: Example 5.1: Numerical (left) and analytic (right) solution at $t = 3$; $h_x = h_y = 0.05$, $k = 0.001$, $p = 0.1$.

Table 2: Errors in Example 5.1, $h_x = 0.1, h_y = 0.1, k = 0.01$.

| $t$(sec) | Proposed Method | | | Method [14] | |
|---|---|---|---|---|---|
| | ER($l_2$) | ER($M$) | CPU time (in sec) | ER($l_2$) | ER($M$) |
| 1 | 2.4712e-04 | 5.7512e-04 | 0.2 | 9.9722e-04 | 2.2746e-03 |
| 2 | 1.1357e-04 | 3.9570e-04 | 0.3 | 1.0926e-03 | 2.8706e-03 |
| 5 | 9.3523e-05 | 2.7283e-04 | 0.7 | 1.1562e-03 | 2.9942e-03 |
| 7 | 3.4398e-04 | 7.9583e-04 | 1.0 | 7.2867e-04 | 1.8781e-03 |
| 10 | 3.7713e-04 | 8.8167e-04 | 1.4 | 5.8889e-04 | 1.5158e-03 |

Table 3: Errors in Example 5.1, $h_x = 0.05, h_y = 0.05, k = 0.001$.

| $t$(sec) | Proposed Method | | | Method [14] | |
|---|---|---|---|---|---|
| | ER($l_2$) | ER($M$) | CPU time (in sec) | ER($l_2$) | ER($M$) |
| 1 | 6.4842e-05 | 1.4964e-04 | 20.3 | 9.8870e-05 | 2.4964e-04 |
| 2 | 3.3664e-05 | 1.1061e-04 | 39.4 | 1.2148e-04 | 3.2296e-04 |
| 5 | 2.3932e-05 | 7.6386e-05 | 97.9 | 1.2762e-04 | 3.3205e-04 |

Table 4: Errors in Example 5.2, $h_x = 0.1, h_y = 0.1, k = 0.01$.

| $t$(sec) | Proposed Method | | | Method [14] | |
|---|---|---|---|---|---|
| | ER($l_2$) | ER($M$) | CPU time (in sec) | ER($l_2$) | ER($M$) |
| .5 | 2.9367e-04 | 1.0000e-03 | 0.4 | 8.3931e-04 | 3.3019e-03 |
| 1 | 2.1044e-04 | 6.2936e-04 | 0.8 | 6.0254e-04 | 2.0597e-03 |
| 2 | 8.4709e-05 | 2.3382e-04 | 1.2 | 2.4167e-04 | 7.6531e-04 |
| 3 | 3.1721e-05 | 8.6166e-05 | 1.7 | 8.9534e-05 | 2.7920e-04 |
| 5 | 4.3121e-06 | 1.1666e-05 | 2.8 | 1.2168e-05 | 3.7800e-05 |

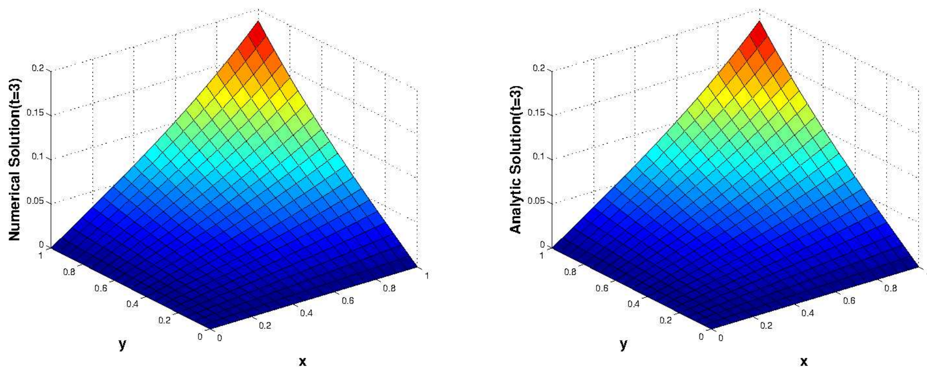


Figure 2: Example 5.2: Numerical (left) and analytic (right) solution at $t = 2$; $\alpha = 10, \beta = 5, h_x = h_y = 0.05, k = 0.001, p = 0.5$.

Table 5: Errors in Example 5.2, $h_x = 0.1, h_y = 0.05, k = 0.01$.

| $t$ (sec) | ER($l_2$) | ER($M$) | CPU time (in sec) |
|---|---|---|---|
| .5 | 2.2064e-04 | 6.9912e-04 | 0.7 |
| 1 | 1.2546e-04 | 4.2288e-04 | 1.3 |
| 2 | 4.6257e-05 | 1.5561e-04 | 2.7 |
| 3 | 6.2593e-06 | 2.1058e-05 | 5.3 |
| 5 | 2.3027e-06 | 7.7469e-06 | 6.7 |

Table 6: Errors in Example 5.3, $h_x = 0.05, h_y = 0.05, k = 0.001$.

| $t$ (sec) | Proposed Method | | | Method [14] | |
|---|---|---|---|---|---|
| | ER($l_2$) | ER($M$) | CPU time | ER($l_2$) | ER($M$) |
| | | $\alpha = 10, \beta = 5$ | | | |
| 1 | 7.1987e-05 | 2.7433e-04 | 19.5 | 1.7174e-04 | 5.6395e-04 |
| 2 | 3.0963e-05 | 1.8481e-04 | 38.6 | 1.6468e-04 | 5.1298e-04 |
| 5 | 1.9137e-05 | 1.2015e-04 | 101.7 | 1.7737e-04 | 5.5627e-04 |
| 7 | 9.3055e-05 | 3.7430e-04 | 136.7 | 1.4200e-04 | 4.7231e-04 |
| 10 | 9.9908e-05 | 4.1338e-04 | 195.5 | 1.2241e-04 | 4.1222e-04 |
| | | $\alpha = 50, \beta = 5$ | | | |
| 1 | 6.6613e-05 | 3.0759e-04 | 19.5 | 1.6766e-04 | 5.6874e-04 |
| 2 | 2.4542e-05 | 1.3383e-04 | 38.5 | 1.7109e-04 | 5.2572e-04 |
| 5 | 3.5198e-05 | 1.2266e-05 | 101.6 | 1.8420e-04 | 5.6940e-04 |
| 7 | 8.7977e-05 | 4.0534e-04 | 136.4 | 1.3760e-04 | 4.7587e-04 |
| 10 | 9.0059e-05 | 4.3761e-04 | 195.1 | 1.1691e-04 | 4.1396e-04 |

We set $\alpha = 10, \beta = 5$ and $\alpha = 50, \beta = 5$ and compute the errors ER($l_2$) and ER($M$) at different time levels. The results presented in Table 6 are clearly better than the ones in [14].

**Example 5.4.** Consider the following problem

$$u_{tt} = u_{xx} + u_{yy} - 2u_t - u - 2e^{x+y-t}, \quad 0 \le x, y \le 1$$

with initial and boundary conditions

$$\begin{aligned} &u(x,y,0) = e^{x+y}, \quad u_t(x,y,0) = -e^{x+y}, && 0 \le x, y \le 1, \\ &u(0,y,t) = e^{y-t}, \quad u(1,y,t) = e^{1+y-t}, && 0 \le y \le 1, \quad t \ge 0, \\ &u(x,0,t) = e^{x-t}, \quad u(x,1,t) = e^{x+1-t}, && 0 \le x \le 1, \quad t \ge 0. \end{aligned}$$

The analytic solution of this problem is $u(x,y,t) = e^{x+y-t}$.

Tables 7 and 8 show the errors $\mathrm{ER}(l_2)$ and $\mathrm{ER}(M)$ at different time levels. Note that the results are better than in [14].

Table 7: Errors in Example 5.4, $h_x = 0.1, h_y = 0.1, k = 0.01$.

| $t$(sec) | Proposed Method | | | Method [14] | |
|---|---|---|---|---|---|
| | $\mathrm{ER}(l_2)$ | $\mathrm{ER}(M)$ | CPU time (in sec) | $\mathrm{ER}(l_2)$ | $\mathrm{ER}(M)$ |
| 1 | 1.4000e-03 | 3.5000e-03 | 0.5 | 1.4441e-02 | 2.9996e-02 |
| 2 | 1.3000e-03 | 2.0000e-03 | 1.2 | 1.3898e-03 | 3.9711e-03 |
| 3 | 2.4578e-04 | 5.9770e-04 | 1.6 | 1.3018e-03 | 2.2178e-03 |
| 5 | 4.9201e-05 | 9.5647e-05 | 2.6 | 1.1112e-04 | 2.0618e-05 |
| 7 | 3.9652e-06 | 1.0328e-05 | 3.6 | 1.3695e-05 | 3.0052e-05 |
| 10 | 2.2358e-07 | 5.1364e-07 | 5.2 | 1.4408e-06 | 2.5354e-06 |

Table 8: Errors in Example 5.4, $h_x = 0.05, h_y = 0.05, k = 0.001$.

| $t$(sec) | Proposed Method | | | Method [14] | |
|---|---|---|---|---|---|
| | $\mathrm{ER}(l_2)$ | $\mathrm{ER}(M)$ | CPU time (in sec) | $\mathrm{ER}(l_2)$ | $\mathrm{ER}(M)$ |
| 1 | 6.1984e-04 | 1.0000e-03 | 19.0 | 3.2351e-03 | 7.4749e-03 |
| 2 | 3.4117e-04 | 6.8395e-04 | 37.7 | 2.8518e-04 | 1.0361e-03 |
| 3 | 5.4693e-05 | 1.4320e-04 | 56.0 | 3.1028e-04 | 5.7859e-04 |
| 4 | 2.7132e-05 | 6.8738e-05 | 74.7 | 9.0898e-05 | 2.7645e-04 |
| 5 | 1.6631e-05 | 3.3196e-05 | 94.0 | 2.4495e-05 | 6.7234e-05 |
| 7 | 6.2659e-07 | 2.3688e-06 | 130.9 | 2.5376e-06 | 8.2203e-06 |
| 10 | 2.9384e-08 | 1.0065e-07 | 187.6 | 3.6505e-06 | 8.5897e-06 |

**Example 5.5.** Consider the non-linear problem

$$\begin{aligned} u_{tt} = u_{xx} + u_{yy} - 2\alpha u_t - \beta^2 u + (1+\beta^2)x^2y^2\sinh(t) - 2x^2 + y^2\sinh(t) \\ + 2\alpha x^2 y^2 \cosh(t) - \sin(u), \quad 0 \le x, y \le 1 \end{aligned}$$

with initial and boundary conditions

$$
\begin{aligned}
u(x,y,0)&=0, \quad u_t(x,y,0)=x^2y^2, && 0\le x,y\le 1,\\
u(0,y,t)&=0, \quad u(1,y,t)=y^2\sinh(t), && 0\le y\le 1, \quad t\ge 0,\\
u(x,0,t)&=0, \quad u(x,1,t)=x^2\sinh(t), && 0\le x\le 1, \quad t\ge 0.
\end{aligned}
$$

The analytic solution of this problem is $u(x,y,t)=x^2y^2\sinh(t)$.

Choose $\alpha = 10$, $\beta = 50$. Table 9 shows the errors at different time levels. Another advantage of the method is the low computational time needed to obtain accurate results. Fig. 3 demonstrates numerical and analytic solutions at the time $t = 1$ and confirms the accuracy of the method.

Table 9: Errors in Example 5.5, $h_x = 0.025, h_y = 0.05, k = 0.001$.

| $t$ | ER($l_2$) | ER($M$) | CPU time(sec) |
|---|---|---|---|
| 0.5 | 3.8513e-05 | 1.8915e-04 | 3.4 |
| 1 | 8.4138e-05 | 3.9704e-04 | 6.5 |
| 2 | 1.8463e-04 | 6.0858e-04 | 12.4 |
| 3 | 2.7815e-04 | 1.7000e-03 | 20.6 |
| 4 | 5.7082e-04 | 4.0000e-03 | 27.4 |
| 5 | 1.5000e-03 | 1.1100e-02 | 34.1 |

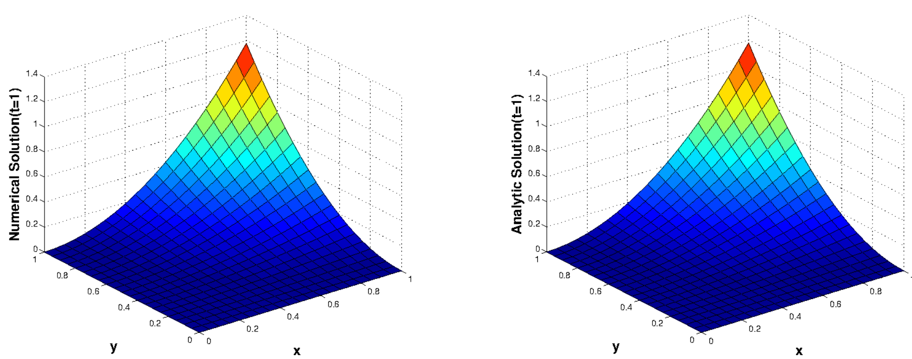


Figure 3: Example 5.5: Numerical (left) and analytic (right) solution at $t = 1$.

## 6. Concluding Remarks

We studied a collocation method for two-dimensional second-order non-linear hyperbolic equations. The method is based on exponential B-splines and, to the best of authors' knowledge, has not been exploited before. We split the initial second order two-dimensional non-linear hyperbolic equation into a system of coupled equations and then transform them into systems of ordinary differential equations. The corresponding equations are solved by SSP-RK(2,2) method. We also showed that this method is unconditionally stable. Numerical examples demonstrate the efficiency and accuracy of the method.

## Acknowledgments

The authors would like to thank the referees for the helpful suggestions which greatly improved the quality of the paper.

## References

[1] E. Ahmed, H.A. Abdusalam and E.S. Fahmy, *On telegraph reaction diffusion and coupled map lattice in some biological systems*, Int. J. Mod. Phys. C. **12**, 717–726 (2001).

[2] T. M. Apostol, *Calculus*, Vol. 2, Waltham Massachusetts: Blaisdell (1969).

[3] C. Boor, *A practical guide to splines*, Appl. Math. Sci. **27**, Springer-Verlag, (1978)

[4] B. Bülbül and M. Sezer, *A Taylor matrix method for the solution of a two-dimensional linear hyperbolic equation*, Appl. Math. Lett. **17**, 101–105 (2004).

[5] M. Dehghan and A. Ghesmati, *Combination of meshless local weak and strong (MLWS) forms to solve the two dimensional hyperbolic telegraph equation*, Eng. Anal. Bound. Elem. **34**, 324–336 (2010).

[6] M. Dehghan and A. Mohebbi, *High order implicit collocation method for the solution of two-dimensional linear hyperbolic equation*, Numer. Meth. Part. D. E. **25**, 232–243 (2009).

[7] H. Ding and Y. Zhang, *A new fourth-order compact difference scheme for the two-dimensional second-order hyperbolic equation*, J. Comput. Appl. Math. **230**, 626–632 (2009).

[8] O. Ersoy and D. Idris, *The exponential cubic B-spline algorithm for Korteweg-de Vries equation*, Adv. Numer. Anal. **367056**, 1–8 (2015).

[9] M.J.E. Golay, *Vapor phase chromatography and telegrapher's equation*, Anal. Chem. **29(6)**, 928–932 (1957).

[10] S. Goldstein, *On diffusion by discontinuous movements, and on the telegraph equation*, Quart. J. Mech. Appl. Math. **4**, 129–156 (1951).

[11] M.K. Jain, *Numerical solution of differential equations*, John Wiley and Sons Ltd (1984).

[12] R. Jiwari, S. Pandit and R.C. Mittal, *A differential quadrature algorithm to solve the two dimensional linear hyperbolic telegraph equation with Dirichlet and Neumann boundary conditions*, Appl. Math. Comput. **218**, 7279–7294 (2012).

[13] B.J. McCartin, *Theory of Exponential Splines*, J. Approx. Theory **66**, 1–23 (1991).

[14] R.C. Mittal and R. Bhatia, *A numerical study of two-dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method*, Appl. Math. Comput. **244**, 976–997 (2014).

[15] R.K. Mohanty and S. Singh, *A new high order approximation for the solution of two-space dimensional quasi-linear hyperbolic equations*, Adv. Math. Phys. **2011**, (2011).

[16] R.K. Mohanty, *New unconditionally stable difference schemes for the solution of multi-dimensional telegraphic equations*, Int. J. Comput. Math. **86(12)**, 2061–2071 (2009).

[17] R.K. Mohanty, *New high accuracy super stable alternating direction implicit methods for two and three dimensional hyperbolic damped wave equations*, Results Phys. **4**, 156–163 (2014).

[18] E. Orsingher, *A planar random motion governed by the two-dimensional telegraph equation*, J. Appl. Prob. **23(2)**, 385–397 (1986).

[19] H.G. Othmer, S.R. Dunbar, W. Alt, *Models of dispersal in biological systems*, J. Math. Biol. **26(3)**, 263–298 (1988).

[20] J. Rashidinia, S. Jamalzadeh and F. Esfahani, *Numerical solution of one-dimensional telegraph equation using cubic B-spline collocation method*, J. Inter. Approx. Scie. Comput. **2014** ,1–8 (2014).

[21] S. Singh, S. Singh and R. Arora, *Numerical solution of second order one dimensional hyperbolic equation by exponential B-spline collocation method*, Numer. Anal. Appl. **10(2)**, 164–176 (2017).

[22] G.D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford University Press (1978).

[23] H. Späth, *Exponential spline interpolation*, in: Computing **4**, pp. 225–233, Springer (1969).

[24] H. Späth, *Two dimensional exponential splines*, Computing **7**, pp. 364–369, Springer (1971).

[25] R. Spiteri and S. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM J. Numer. Anal. **40**, 469–491 (2002).

[26] G. Strang, *Linear Algebra and its applications*, Harcourt Brace Jovanovich, San Diego (1976).

[27] A. Tripathi and R.C. Mittal, *Numerical solutions of two-dimensional unsteady convection-diffusion problems using modified bi-cubic B-spline finite elements*, Int. J. Comput. Math. **94**, 1–21 (2017).

[28] Y. Yang and C. Zhang, *Kernel Based Telegraph-Diffusion Equation for Image Noise Removal*, Math. Probl. Eng. **283751**, 1–10 (2014).