

# Non-polynomial Cubic Spline Method for the Solution of Second-order Linear Hyperbolic Equation

NAZAN ÇAĞLAR

Faculty of Economic and Administrative Science  
Istanbul Kultur University, Faculty of Economic and Administrative Science, 34156 Atakoy Istanbul  
TURKEY

**Abstract:** Second-order linear hyperbolic equations are solved by using a new three level method based on non-polynomial spline in the space direction and Taylor expansion in the time direction. Numerical results reveal that three level method based on non-polynomial spline is implemented and effective.

**Keywords:** Second-order linear hyperbolic equation , Non-polynomial cubic spline

Received: May 13, 2021. Revised: October 7, 2021. Accepted: October 21, 2021. Published: November 9, 2021.

## 1. Introduction

We consider the second-order linear hyperbolic equation:

$$u_{tt}(x, t) + 2\alpha u_t(x, t) + \beta^2 u(x, t) = u_{xx}(x, t) + f(x, t), \quad x \in (a, b), \quad t > 0 \quad (1)$$

with initial conditions

$$u(x, 0) = \Phi(x), \quad u_t(x, 0) = \Psi(x)$$

and boundary conditions

$$u(a, t) = g_1(t), \quad u(b, t) = g_2(t)$$

where  $\alpha$  and  $\beta$  are constants.

Above one can find representations of the damped wave and telegraph equations respectively. See [1] for the existence and approximations of the solutions investigated.

There have been many prominent work regarding the development and implementation of the high resolution methods for the numerical solution of the second – order linear hyperbolic equation in (1), see [1 – 3]. Mohanty and Jain [4 – 6] developed three-level implicit schemes for linear hyperbolic equations.

Also, Huan-Wen Liu and Li-Bin Liu solved [8] linear hyperbolic equation, where their solution based on quartic spline interpolation our solution based on non-polynomial spline method. In this paper, we propose a spline difference scheme to solve the linear hyperbolic equation (1).

We proceed this paper as follows; Section 2 briefly describes the non-polynomial spline function. Section 3 describes the methods used to solve and analyze the solution of problem (1). Section 4 contains the numerical results and illustrations obtained by MATLAB 6.5 before the overall conclusion in Section 5.

## 2. Spline Method

We divide the interval  $[a, b]$  into  $n$  equal subintervals using the grid points

$$x_i = a + ih, \quad i = 0, 1, 2, \dots, n,$$

with

$$x_0 = a, \quad x_n = b, \quad h = (b - a)/n$$

where  $n$  is defined as an arbitrary positive integer.

Let  $u(x)$  be the exact solution and  $u_i$  approximation of  $u(x_i)$  which is obtained by the non-polynomial cubic  $S_i(x)$  defined as passing through the points  $(x_i, u_i)$  and  $(x_{i+1}, u_{i+1})$ . Here we do not only require that  $S_i(x)$  satisfies interpolatory conditions at

$x_i$  and  $x_{i+1}$ , but also the continuity of first derivative at the common nodes  $(x_i, u_i)$  are fulfilled. We write  $S_i(x)$  in the form:

$$S_i(x) = a_i + b_i(x - x_i) + c_i \sin \tau(x - x_i) + d_i \cos \tau(x - x_i), i = 0, 1, \dots, n - 1 \quad (2)$$

where  $a_i, b_i, c_i$  and  $d_i$  are constants and  $\tau$  is a free parameter.

The non-polynomial function  $S(x)$  which belongs to the class  $C^2[a, b]$  interpolates  $u(x)$  at the grid points  $x_i$ , where  $i = 0, 1, 2, \dots, n$ , and reduces to an ordinary cubic spline  $S(x)$  in  $[a, b]$  depending on a parameter  $\tau$  when  $\tau \rightarrow 0$ .

To derive expression for the coefficients of Eq. (2) in term of  $u_i, u_{i+1}, M_i$  and  $M_{i+1}$ , we first define:

$$S_i(x_i) = u_i, S_i(x_{i+1}) = u_{i+1}, S''(x_i) = M_i, S''(x_{i+1}) = M_{i+1}. \quad (3)$$

After manipulating through algebra we get the following expression:

$$\begin{aligned} a_i &= u_i + \frac{M_i}{\tau^2}, \\ b_i &= \frac{u_{i+1} - u_i}{h} + \frac{M_{i+1} - M_i}{\tau \theta}, \\ c_i &= \frac{M_i \cos \theta - M_{i+1}}{\tau^2 \sin \theta}, \\ d_i &= -\frac{M_i}{\tau^2}, \end{aligned} \quad (4)$$

where  $\theta = \tau h$  and  $i = 0, 1, 2, \dots, n - 1$ .

Using the continuity of the first derivative at  $(x_i, u_i)$ , that is  $S'_{i-1}(x_i) = S'_i(x_i)$  we obtain the following relations for  $i = 1, \dots, n - 1$ .

$$\begin{aligned} aM_{i+1} + bM_i + aM_{i-1} &= \\ (1/h^2)(u_{i+1} - 2u_i + u_{i-1}), \end{aligned} \quad (5)$$

where  $a = (-1/\theta^2 + 1/\theta \sin \theta)$ ,  $b = (1/\theta^2 - \cos \theta/\theta \sin \theta)$  and  $\theta = \tau h$ . If  $b = 5/12$  and  $a = 1/12$  the method is fourth-order convergent [9].

### 3. The Spline Difference Scheme

By using the Taylor expansion in the time direction for every  $x_i$  where  $i = 1, 2, \dots, n - 1$ , we have the following difference schemes

$$u(x_i, t_j) = \frac{u(x_i, t_{j+1}) + 2u(x_i, t) + u(x_i, t_{j-1}))}{4} + O(k^2), \quad (6)$$

$$u_{xx}(x_i, t_j) = \frac{u_{xx}(x_i, t_{j+1}) + u_{xx}(x_i, t_{j-1}))}{2} + O(k^2), \quad (7)$$

$$u_{xx}(x_i, t_j) = \frac{u(x_i, t_{j+1}) - u(x_i, t_{j-1}))}{2k} + O(k^2), \quad (8)$$

$$u_{xx}(x_i, t_j) = \frac{u(x_i, t_{j+1}) - 2u(x_i, t_j) + u(x_i, t_{j-1}))}{k^2} + O(k^2). \quad (9)$$

The given eq.(1) can be discretized as

$$\begin{aligned} &\frac{u(x_i, t_{j+1}) - 2u(x_i, t_j) + u(x_i, t_{j-1}))}{k^2} + \\ &2\alpha \frac{u(x_i, t_{j+1}) - u(x_i, t_{j-1}))}{2k} + \\ &\beta^2 \frac{u(x_i, t_{j+1}) + 2u(x_i, t) + u(x_i, t_{j-1}))}{4} + \\ &= \frac{u_{xx}(x_i, t_{j+1}) + u_{xx}(x_i, t_{j-1}))}{2} + \\ &f(x_i, t_j) + O(k^2), \end{aligned} \quad (10)$$

We can rewrite (5) in a new form:

$$\begin{aligned} (1 + \frac{1}{12}\delta_x^2)M(x_i, t_j) = \\ \frac{1}{h^2}\delta_x^2 u(x_i, t_j), i = 1, \dots, n - 1 \end{aligned} \quad (11)$$

where

$$\begin{aligned} \delta_x M(x_i, t_j) &= M(x_{i+\frac{1}{2}}, t_j) - M(x_{i-\frac{1}{2}}, t_j), \\ \delta_x^2 M(x_i, t_j) &= \delta_x(\delta_x M(x_i, t_j)) = M(x_{i+1}, t_j) - \\ &2M(x_i, t_j) + M(x_{i-1}, t_j), \\ \text{for } i &= 1, \dots, n - 1. \text{ Putting (6) and (11), it follows that;} \end{aligned}$$

$$\begin{aligned} (1 + \frac{1}{12}\delta_x^2)[u_{xx}(x_i, t_{j+1}) + u_{xx}(x_i, t_{j-1})] = \\ (1 + \frac{1}{12}\delta_x^2)[M(x_i, t_{j+1}) + M(x_i, t_{j-1}) + O(h^4)] \\ = \frac{1}{h^2}\delta_x^2[u(x_i, t_{j+1}) + u(x_i, t_{j-1})] + O(h^4) \end{aligned} \quad (12)$$

Applying the operator  $(1 + \frac{1}{12}\delta_x^2)$  to two sides of Eq. (10) and using Eq. (12), then it is obtained as follows

$$\begin{aligned} \frac{1}{k^2}(1 + \frac{1}{12}\delta_x^2)u(x_i, t_j) + \frac{\alpha}{k}(1 + \frac{1}{12}\delta_x^2)\delta_t u(x_i, t_j) + \\ \frac{\beta^2}{4}(1 + \frac{1}{12}\delta_x^2)[u(x_i, t_{j+1}) + 2u(x_i, t_j) + u(x_i, t_{j-1})] = \\ (1 + \frac{1}{12}\delta_x^2)f(x_i, t_j) + O(k^2 + h^4) \\ i = 1(1)n - 1, j = 1, 2, \dots \end{aligned} \quad (13)$$

The proposed scheme (13) is an implicit three level scheme. Before starting on any computation it is found necessary to know the value of  $u(x,t)$  at the nodal points of the first time level expressed as when  $t=k$ . Following the work in [2], a Taylor series expansion at [2], a Taylor series expansion at  $t=k$  may be written as

$$\begin{aligned} u(x, k) = u(x, 0) + ku_t(x, 0) + \frac{k^2}{2}u_{tt}(x, 0) + \\ \frac{k^3}{6}u_{ttt}(x, 0) + O(k^4) \end{aligned} \quad (14)$$

Using the initial values, from (1) we can calculate

$$\begin{aligned} u_{tt}(x, 0) = \phi_{xx}(x, 0) + \\ f(x, 0) - 2\alpha u_t(x, 0) - \beta^2 u(x, 0), \end{aligned} \quad (15)$$

$$\begin{aligned} u_{ttt}(x, 0) = \psi_{xx}(x, 0) + \\ f_t(x, 0) - 2\alpha u_{tt}(x, 0) - \beta^2 u_t(x, 0). \end{aligned} \quad (16)$$

We can obtain the numerical solution of  $u$  by using initial values in (15) and (16) for  $t=k$ .

## 4. Numerical Examples

Here we provide the evidence of illustrations by MATLAB 6.5 of our method to two second-order linear hyperbolic equations.

### Example 1.

Consider the following equation

$$u_{tt}(x, t) + 4u_t(x, t) + 2u(x, t) = u_{xx}(x, t), \quad x \in (a, b), \quad t > 0$$

with initial conditions

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

and boundary conditions

$$u(0, t) = 0, \quad u(\pi, t) = 0.$$

The exact solution of the above problem is  $u(x, t) = e^{-t} \sin x$  which could be found by solving the equation using the scheme (13) provided in this paper. The absolute errors given of the scheme (16) in [7], (23) in [8] and by the present scheme in (13) are listed in Tables 1-8, respectively. It can be seen from the tables that when  $h = \frac{\pi}{300}$  and  $k = 0.1$ , the accuracy of solutions obtained by using scheme (16) in [7] provides more accurate results than the present scheme (13). The reason is that the error orders of the scheme (16) in [7] is approximately  $O(k^5)$  in addition to the step length “ $h$ ” being quite small. As  $k$  respectively decreases to  $k = 0.1$  and  $k = 0.01$ , since  $k$  is now quite small in comparison with  $h$ , the errors of numerical solutions mainly come from the approximation in the space direction, therefore the absolute errors obtained from the present scheme (13) is much better than those by using the scheme (16)

in [7]. Finally, we believe it is crucial to mention that the absolute errors of scheme (23) in [8] are similar to those by using the present scheme (13) where scheme (23) in [8] using quartic spline functions, we use non-polynomial spline functions. The numerical results are illustrated in Fig 1.

**Example 2.** We consider the following equations

$$u_{tt}(x, t) + 2\alpha u_t(x, t) + \beta^2 u(x, t) = u_{xx}(x, t) + (4 - 4\alpha + \beta^2 + h^2)e^{-2t} \sinh x,$$

$$\alpha > \beta \geq 0$$

$$, x \in (a, b), \quad t > 0$$

with initial conditions

$$u(x, 0) = \sinh x, \quad u_t(x, 0) = -2\sinh x$$

and boundary conditions

$$u(0, t) = 0, \quad u(1, 0) = e^{-2t} \sinh$$

The exact solution of the above problem is  $u(x, t) = e^{-2t} \sinh x$ . The absolute errors given by the scheme (16) in [7], by the scheme (23) in [8] and by present scheme (13) are listed in Tables 9-14, respectively. Similar discussion to example 1 is valid for example 2. Only difference that here we use the term  $\lambda = \frac{k}{h}$  to simplify the calculation. The numerical results are illustrated in Fig. 2.

## 5. Conclusion

This paper presents a new non-polynomial spline method to solve the linear hyperbolic equation. The distinctness of this method as against the previous study in [10] is it using the Taylor expansion in time direction. Using the method described in this study gives acceptable results. We have concluded that numerical results converge to the exact solution when  $k$  goes to zero and for smaller  $h$  we have seen that the maximum absolute error decreases. Finally, we believe it is crucial to mention that the new proposed method for solving linear hyperbolic equation gives better numerical results than those produced by a finite difference method [11].

## 6. References

- [1] E.H. Twizel, An explicit difference method for the wave equation with extended stability range, BIT Numerical Mathematics 19(3):378-383(1979).
- [2] R.K. Mohanty, M.K. Jain, K. George, On the use of high order difference methods for the system of one space second order non-linear hyperbolic equations with variable coefficients, Journal of Computational and Applied Mathematics 72(2):421-431(1996).
- [3] M. Climent, S.H. Leventhal, A note on the operator compact implicit method for the wave equation, Mathematics of Computation 32(1):143-147(1978).
- [4] R.K. Mahanty, M.K. Jain, An unconditionally stable alternating direction implicit scheme for the two space dimensional linear hyperbolic equation, Numerical Methods for Partial Differential Equations 17(6):684-688(2001).
- [5] R.K. Mahanty, M.K. Jain, U Arora, An unconditionally stable ADI method for the linear hyperbolic equation in three space dimensional, International Journal of Computer Mathematics 79(1):133-142(2002).
- [6] R.K. Mahanty, An unconditionally stable difference scheme for the one-space dimensional linear hyperbolic equation, Applied Mathematics Letters 17(1):101-105(2004).
- [7] F. Gao, C.M. Chi, Unconditionally stable difference schemes for a one-space-dimensional linear hyperbolic equation, Applied Mathematics and Computation 187(2):1272-1276(2007).
- [8] Huan-Wen Liu, Li-Bin Liu, An unconditionally stable spline difference schemes of  $O(k^2 + h^4)$  for solving the second-order 1D linear hyperbolic equation, Mathematical and Computer Modelling, in press.
- [9] J. Rashidinia, R. Mohammadi, Non-polynomial cubic spline methods for the solution of parabolic equations, International Journal of Computer Mathematics 85(5):843-850(2008).
- [10] J. Rashidinia, R. Jalilian, V. Kazemi, Spline methods for the solutions of hyperbolic equations, Applied Mathematics and Computation 190:882-886(2007).
- [11] K. George, E.H. Twizell, Stable second order-finite difference methods for linear initial boundary-value problems, Applied Mathematics Letters 19:146-154(2006).

Table 1: Absolute errors of the scheme (16) in [7] ( $h = \frac{\pi}{300}, k=0.1$ ).

t	$x = \frac{\pi}{30}$	$x = \frac{8\pi}{30}$	$x = \frac{15\pi}{30}$	$x = \frac{22\pi}{30}$	$x = \frac{29\pi}{30}$
0.5	0.0577e-06	0.4102e-06	0.5519e-06	0.4102e-06	0.0577e-06
1.0	0.0105e-05	0.0747e-05	0.1005e-05	0.0747e-05	0.0105e-05
1.5	0.0114e-05	0.0811e-05	0.1091e-05	0.0811e-05	0.0114e-05
2.0	0.1015e-06	0.7215e-06	0.9709e-06	0.7215e-06	0.1015e-06

Table 2: Absolute errors of scheme (23) in [8] ( $h = \frac{\pi}{300}, k=0.1$ ).

t	$x = \frac{\pi}{30}$	$x = \frac{8\pi}{30}$	$x = \frac{15\pi}{30}$	$x = \frac{22\pi}{30}$	$x = \frac{29\pi}{30}$
0.5	0.0181e-03	0.1418e-03	0.1926e-03	0.1445e-03	0.0221e-03
1.0	0.0379e-03	0.2969e-03	0.4033e-03	0.3026e-03	0.0463e-03
1.5	0.0429e-03	0.3355e-03	0.4558e-03	0.3419e-03	0.0523e-03
2.0	0.0389e-03	0.3039e-03	0.4128e-03	0.3096e-03	0.0475e-03

Table 3: Absolute errors of the present scheme (13)( $h = \frac{\pi}{300}, k=0.1$ ).

t	$x = \frac{\pi}{30}$	$x = \frac{8\pi}{30}$	$x = \frac{15\pi}{30}$	$x = \frac{22\pi}{30}$	$x = \frac{29\pi}{30}$
0.5	0.0211e-03	0.1502e-03	0.2021e-03	0.1502e-03	0.0211e-03
1.0	0.0429e-03	0.3056e-03	0.4112e-03	0.3056e-03	0.0429e-03
1.5	0.0482e-03	0.3427e-03	0.4611e-03	0.3427e-03	0.0482e-03
2.0	0.0435e-03	0.3092e-03	0.4161e-03	0.3092e-03	0.0435e-03

Table 4: Absolute errors of the scheme (16) in [7] ( $h = \frac{\pi}{30}, k=0.1$ ).

t	$x = \frac{\pi}{30}$	$x = \frac{8\pi}{30}$	$x = \frac{15\pi}{30}$	$x = \frac{22\pi}{30}$	$x = \frac{29\pi}{30}$
0.5	0.0483e-04	0.3462e-04	0.4771e-04	0.3777e-04	0.1430e-04
1.0	0.0904e-04	0.6479e-04	0.8928e-04	0.7069e-04	0.0904e-04
1.5	0.0990e-04	0.7095e-04	0.9776e-04	0.7740e-04	0.0990e-04
2.0	0.0884e-04	0.6337e-04	0.8731e-04	0.6913e-04	0.0884e-04

Table 5: Absolute errors of the scheme (23) in [8] ( $h = \frac{\pi}{30}, k=0.1$ ).

t	$x = \frac{\pi}{30}$	$x = \frac{8\pi}{30}$	$x = \frac{15\pi}{30}$	$x = \frac{22\pi}{30}$	$x = \frac{29\pi}{30}$
0.5	0.0101e-03	0.1032e-03	0.1826e-03	0.1032e-03	0.0301e-03
1.0	0.0321e-03	0.6995e-03	0.2033e-03	0.6995e-03	0.0412e-03
1.5	0.0375e-03	0.4386e-03	0.5356e-03	0.4386e-03	0.0475e-03
2.0	0.0532e-03	0.5065e-03	0.3128e-03	0.5065e-03	0.0331e-03

Table 6: Absolute errors of the present scheme(13) ( $h = \frac{\pi}{30}, k=0.1$ ).

t	$x = \frac{\pi}{30}$	$x = \frac{8\pi}{30}$	$x = \frac{15\pi}{30}$	$x = \frac{22\pi}{30}$	$x = \frac{29\pi}{30}$
0.5	0.0211e-03	0.1502e-03	0.2021e-03	0.1502e-03	0.0420e-03
1.0	0.0298e-03	0.3055e-03	0.4111e-03	0.3055e-03	0.0429e-03
1.5	0.0819e-03	0.3426e-03	0.4610e-03	0.3426e-03	0.0481e-03
2.0	0.0349e-03	0.3092e-03	0.4161e-03	0.3092e-03	0.0865e-03

Table 7: Absolute errors of the present scheme (13),scheme(16) in [7] and scheme (23) in [8] ( $h = \frac{\pi}{30}, k=0.01$ ).

	$t$	$x = \frac{\pi}{10}$	$x = \frac{3\pi}{10}$	$x = \frac{5\pi}{10}$	$x = \frac{7\pi}{10}$	$x = \frac{9\pi}{10}$
The present s.	1.0	0.13203e-05	0.34565e-04	0.42725e-05	0.34565e-05	0.13203e-05
[7]	1.0	0.29477e-04	0.77174e-04	0.95392e-04	0.77174e-04	0.29477e-04
[8]	1.0	0.08869e-05	0.31701e-05	0.42424e-05	0.31701e-05	0.08869e-05
The present s.	2.0	0.12957e-05	0.31161e-05	0.41931e-05	0.33923e-05	0.12957e-05
[7]	2.0	0.28834e-04	0.75489e-04	0.93309e-04	0.75489e-04	0.28834e-04
[8]	2.0	0.08712e-05	0.31140e-05	0.41673e-05	0.31140e-05	0.08712e-05

Table 8: Absolute errors of the present scheme(13),scheme(16) in[7] and scheme (23) in [8]( $h = \frac{\pi}{30}, k=0.001$ ).

	$t$	$x = \frac{\pi}{10}$	$x = \frac{3\pi}{10}$	$x = \frac{5\pi}{10}$	$x = \frac{7\pi}{10}$	$x = \frac{9\pi}{10}$
The present s.	1.0	0.10965e-08	0.71347e-08	0.88195e-08	0.71347e-08	0.10965e-08
[7]	1.0	0.29477e-04	0.77174e-04	0.95392e-04	0.77174e-04	0.29477e-04
[8]	1.0	0.18390e-08	0.65744e-08	0.87995e-08	0.65744e-08	0.18390e-08
The present s.	2.0	0.19920e-08	0.70295e-08	0.83985e-08	0.70295e-08	0.19920e-08
[7]	2.0	0.28834e-04	0.75489e-04	0.93309e-04	0.75489e-04	0.28834e-04
[8]	2.0	0.17996e-08	0.64330e-08	0.86095e-08	0.64330e-08	0.17996e-08

Table 9:RMS errors of schemes in [6] when  $\lambda=3.2$ .

$h$	$\alpha=50, \beta=5, \sigma=0.25, \gamma=0.75$		$\alpha=50, \beta=2, \sigma=10, \gamma=5$	
	$t = 1.0$	$t = 2.0$	$t = 1.0$	$t = 2.0$
$\frac{1}{16}$	0.6386e-02	0.5937e-02	0.8998e-02	0.8827e-02
$\frac{1}{32}$	0.2229e-02	0.1800e-02	0.2850e-02	0.2652e-02
$\frac{1}{64}$	0.6002e-03	0.4826e-03	0.7676e-03	0.7276e-03

Table 10:RMS errors of schemes in [8] when  $\lambda=3.2$ .

$h$	$\alpha=50, \beta=5$		$\alpha=50, \beta=2$	
	$t = 1.0$	$t = 2.0$	$t = 1.0$	$t = 2.0$
$\frac{1}{16}$	0.1368e-02	0.1005e-02	0.0853e-02	0.1076e-02
$\frac{1}{32}$	0.2159e-03	0.1368e-03	0.3015e-03	0.3915e-03
$\frac{1}{64}$	0.1507e-04	0.4015e-04	0.6059e-04	0.5252e-04

Table 11:RMS errors of present schemes when  $\lambda =3.2$ .

$h$	$\alpha =50, \beta =5$		$\alpha =50, \beta =2$	
	$t = 1.0$	$t = 2.0$	$t = 1.0$	$t = 2.0$
$\frac{1}{16}$	0.0253e-02	0.2643e-02	0.0258e-02	0.2839e-02
$\frac{1}{32}$	0.1659e-03	0.3804e-03	0.3872e-03	0.3648e-02
$\frac{1}{64}$	0.1002e-04	0.4984e-04	0.6754e-04	0.4563e-03

Table 12: RMS errors of schemes in [6] when  $\lambda =1.6$ .

$h$	$\alpha =10, \beta =5, \sigma =0.5, \gamma =1.0$		$\alpha =20, \beta =10, \sigma =\gamma =1.0$	
	$t = 1.0$	$t = 2.0$	$t = 1.0$	$t = 2.0$
$\frac{1}{16}$	0.6752e-03	0.1938e-03	0.4496e-02	0.7960e-03
$\frac{1}{32}$	0.2644e-03	0.7548e-04	0.1406e-03	0.2606e-04
$\frac{1}{64}$	0.7236e-04	0.2054e-04	0.2478e-04	0.4585e-05

Table 13: RMS errors of schemes [8] when  $\lambda=1.6$ .

$h$	$\alpha=10, \beta=5$		$\alpha=20, \beta=10$	
	$t = 1.0$	$t = 2.0$	$t = 1.0$	$t = 2.0$
$\frac{1}{16}$	0.1651e-07	0.2648e-08	0.3382e-05	0.1285e-05
$\frac{1}{32}$	0.5269e-08	0.1508e-08	0.2567e-06	0.2618e-07
$\frac{1}{64}$	0.9686e-09	0.1756e-09	0.3447e-07	0.4172e-08

Table 14: RMS errors of present schemes when  $\lambda=1.6$ .

$h$	$\alpha=10, \beta=5$		$\alpha=20, \beta=10$	
	$t = 1.0$	$t = 2.0$	$t = 1.0$	$t = 2.0$
$\frac{1}{16}$	0.2361e-07	0.1243e-08	0.4365e-05	0.8268e-06
$\frac{1}{32}$	0.1243e-08	0.8776e-09	0.3553e-06	0.7981e-08
$\frac{1}{64}$	0.8810e-09	0.0756e-09	0.2021e-07	0.0738e-08

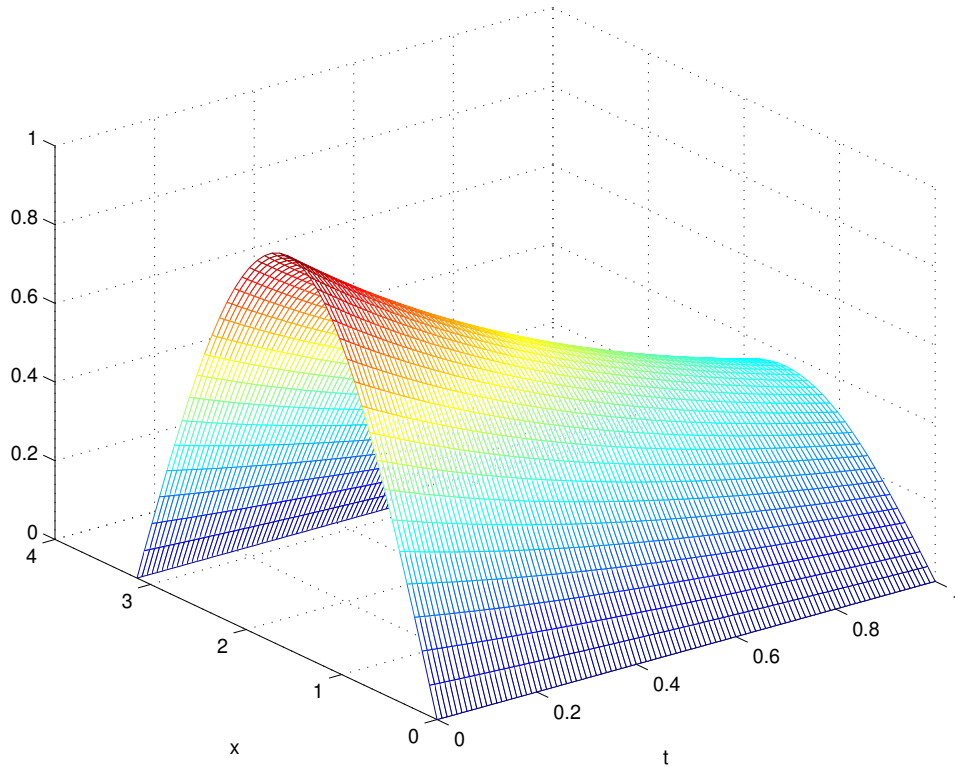
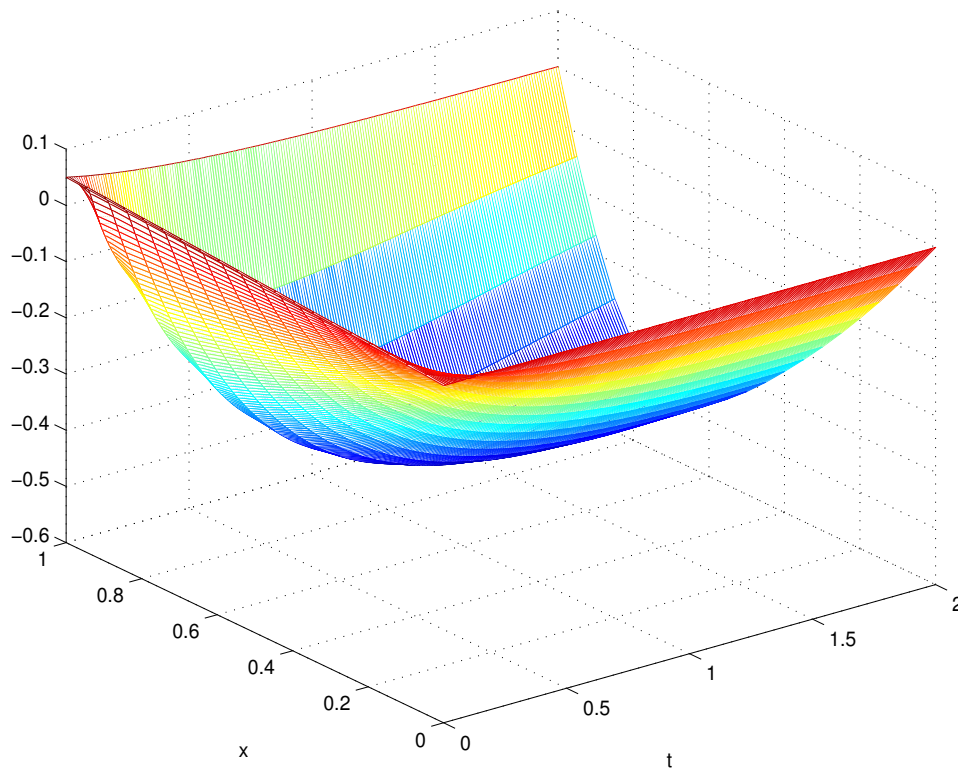


Fig. 1: Results for first example with  $h = \frac{\pi}{40}$  and  $k = 0.01$



**Fig. 2:** Results for second example with  $h = \frac{1}{40}, k = 0.01, \alpha = 50$  and  $\beta = 2$

**Creative Commons Attribution License 4.0  
(Attribution 4.0 International, CC BY 4.0)**

This article is published under the terms of the Creative Commons Attribution License 4.0  
[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)