The application of cubic trigonometric B-spline to the numerical solution of the hyperbolic problems

Muhammad Abbas a,b,⇑, Ahmad Abd. Majid b, Ahmad Izani Md. Ismail b, Abdur Rashid c

a Department of Mathematics, University of Sargodha, 40100 Sargodha, Pakistan
b School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia
c Department of Mathematics, Gomal University, 29050 Dera Ismail Khan, Pakistan

A R T I C L E   I N F O

Keywords:
One-dimensional wave equation
Non-local conservation constraints
Cubic trigonometric B-spline basis functions
Cubic trigonometric B-spline collocation method
Stability

A B S T R A C T

In this paper, a collocation finite difference scheme based on new cubic trigonometric B-spline is developed and analyzed for the numerical solution of a one-dimensional hyperbolic equation (wave equation) with non-local conservation condition. The usual finite difference scheme is used to discretize the time derivative while a cubic trigonometric B-spline is utilized as an interpolation function in the space dimension. The scheme is shown to be unconditionally stable using the von Neumann (Fourier) method. The accuracy of the proposed scheme is tested by using it for several test problems. The numerical results are found to be in good agreement with known exact solutions and with existing schemes in literature.

1. Introduction

There are quite a number of phenomena in science and engineering which can be modeled by the use of hyperbolic partial differential equations subject to non-local conservation condition instead of traditional boundary conditions [1] and these arise in the study of chemical heterogeneity [2,3], medical science, viscoelasticity, plasma physics [4] and thermoelasticity [5,6]. This type of problems also arises in non-local reactive transport in underground water flows in porous media, semi-conductor modeling, non-Newtonian fluid flows and radioactive nuclear decay in fluid flows [7]. The temperature distribution of air near the ground over time during calm clear nights is a good example of such models [8]. The analysis, development and implementation of numerical methods for the solution of such problems have received wide attention in the literature.

In this study, we discuss the numerical solution of the wave equation subject to non-local conservation condition, using cubic trigonometric B-spline collocation method (CuTBSM). Consider a vibrating elastic string of length $L$ which is located on the $x$-axis of the interval $[0, L]$. Then, the vertical displacement $u(x, t)$ of the elastic string at point $x$ units from the origin after a time $t$ elapsed is given by the one-dimensional wave equation

$$\frac{\partial^2 u}{\partial t^2}(x, t) - \alpha^2 \frac{\partial^2 u}{\partial x^2}(x, t) = q(x, t), \quad 0 \leq x \leq L, \quad 0 \leq t \leq T. \quad (1)$$
For the solution of the problem to exist and be unique, we can prescribe the following initial and non-local conservation boundary conditions

\[
\begin{align*}
\text{Initial displacement} : u(x, t = 0) &= e_1(x), \quad 0 \leq x \leq L, \\
\text{Initial velocity} : u_t(x, t = 0) &= e_2(x), \quad 0 \leq x \leq L,
\end{align*}
\] (2)

and boundary condition

\[ \beta u(x = 0, t) + \gamma u_t(x = 0, t) = f_1(t), \quad 0 < t \leq T, \]

(3)

and non-local conservation boundary condition

\[ \int_0^t u(x, t)dx = f_2(t), \quad 0 < t \leq T, \]

(4)

where \( q, g_1, g_2, f_1 \) and \( f_2 \) are the known functions and \( \beta, \gamma \) are constants. Beilin [9] discussed the existence and uniqueness of the solution of this problem.

A great deal of research on the numerical solution of partial differential equation has been done, especially on the one dimensional wave equation subject to non-local conservation condition. Ang [10] developed a numerical scheme based on the integro-differential equation with local interpolating functions for the solution of one-dimensional wave equation with non-local conservation condition and appropriately prescribed initial boundary conditions. Dehghan developed several finite difference methods for the one-dimensional wave equation with Neumann and integral conditions [1], method of lines (MOL I and II) [11], Thin plate splines-radial basis function (TPS-RBF) [12], Multiquadratics-radial basis function (MQ-RBF) [12], Compactly supported-radial basis function (CS-RBF) [12], Cubic B-spline scaling functions method (CBSFM) [13], Shifted Legendre tau technique (SLTM) [14], Bernstein Ritz-Galerkin method [15] for solution of the one-dimensional wave equation subject to an integral conservation condition. Feng and Li [16] conducted a study which was concerned with the boundary stabilization of a one-dimensional wave equation subject to boundary non-linear uncertainty. The non-linear uncertainty term was first calculated as an output and then canceled by its estimates.

The theory of B-spline functions has attracted attention in the literature [18–28] for the numerical solution of linear and non-linear boundary value problems [19–21] in science and engineering. This is because they have notable curve properties and features that make them suitable for shape analysis. Bickley [29] introduced the idea of using cubic splines rather than a global high-order approximation to improve accuracy for the computed solution of linear ordinary differential equation. Fyfe [30] used the method proposed by Bickley and conducted an error analysis. It was found that the spline method is better than the usual finite difference scheme because it has the flexibility to obtain the solution at any point in the domain with greater accuracy. The numerical solution of certain partial differential equations can be obtained using B-spline functions of various degrees. As examples, a cubic B-spline scaling function was used to solve the one-dimensional wave equation with non-local conservation condition [13], a combined finite difference and cubic B-spline approach was applied for the solution of the heat and wave equation [22–25], a collocation finite element method based on cubic spline was developed in [7] for the solution of a generalized wave equation with non-local conservation conditions and fourth and sixth degree B-spline functions were used for the solution of third and fifth order boundary value problems respectively [19–20]. Caglar et al. [31] used a non-polynomial cubic spline for solving the one-dimensional non-local hyperbolic equation subject to an integral conservation conditions.

Schoenberg introduced the trigonometric spline functions in 1964 and he proved the existence and local support of these functions, named as trigonometric B-spline functions which were constructed from trigonometric functions, as contrasted to polynomial functions in the case of B-spline. The derivation and geometric properties of functions were discussed in [32,33]. The trigonometric B-spline produced more accurate results of linear and non-linear initial boundary value problems as compared to traditional B-spline functions [34,35].

In our present study, a numerical collocation finite difference approach based on cubic trigonometric B-spline is presented for the solution of wave equation (1) with initial conditions in equation (2) and non-local conservation boundary conditions in Eqs. (3) and (4). Some researchers have utilized the traditional B-spline collocation methods to solve the wave equation subject to different types of boundary conditions but so far as we are aware not with cubic trigonometric B-spline collocation method. A usual finite difference scheme is used to discretize the time derivative. Cubic trigonometric B-spline is applied as an interpolation function in the space dimension. The unconditional stability property of the method is proved by von Neumann method. The feasibility of the method is shown by test problems and the approximated solutions are found to be in good agreement with the exact solutions. The proposed method is superior to CuBSM [24] due to smaller storage and CPU time.

This paper is organized as follows: A numerical method incorporating a finite difference approach with cubic trigonometric B-spline is presented in Section 2. In Section 3, the von Neumann approach is used to prove the stability of method. Numerical examples are considered in Section 4 to show the feasibility of the proposed method. Finally, in Section 5, the conclusion of this study is given.
2. Description of trigonometric B-spline method

In this section, we discuss the cubic trigonometric B-spline collocation method for solving numerically the one-dimensional hyperbolic equation (1). The solution domain \( a \leq x \leq b \) is equally divided by knots \( x_i \) into \( n \) subintervals \([x_i, x_{i+1}]\), \( i = 0, 1, 2, \ldots, n - 1 \) where \( a = x_0 < x_1 < \cdots < x_n = b \). Our approach for one-dimensional hyperbolic equation using collocation method with cubic trigonometric B-spline is to seek an approximate solution as [36,37]

\[
U_i(x, t) = \sum_{j=0}^{n-1} C_j(t) TB_j^i(x)
\]

where \( C_j(t) \) are to be determined for the approximated solutions \( U_i(x, t) \) to the exact solutions \( u(x, t) \) at the point \((x_i, t_j)\) whilst \( TB_j^i(x) \) are cubic trigonometric B-spline basis functions defined by [32–35]

\[
TB_j^i(x) = \frac{1}{\omega} \begin{cases} 
\frac{p^3(x_j)}{p^2(x_i)}, & x \in [x_i, x_{i+1}) \\
p(x_j)(p(x_i)q(x_{i+2}) + q(x_{i+1})p(x_{i+1})) + q(x_{i+1})p^3(x_{i+1}), & x \in [x_{i+1}, x_{i+2}) \\
q(x_i)(p(x_{i+1})q(x_{i+3}) + q(x_{i+2})p(x_{i+2})) + p(x_i)q^3(x_{i+3}), & x \in [x_{i+2}, x_{i+3}) \\
q^3(x_{i+4}), & x \in [x_{i+3}, x_{i+4})
\end{cases}
\]

where,

\[
p(x_i) = \sin \left( \frac{x_i - x_i}{2} \right), \quad q(x_i) = \sin \left( \frac{x_i - x_{i+1}}{2} \right), \quad \omega = \sin \left( \frac{h}{2} \right) \sin(h) \sin \left( \frac{3h}{2} \right)
\]

and where \( h = (b - a)/n \) and \( TB_j^i(x) \) is a piecewise cubic trigonometric function with some geometric properties like \( C^2 \) continuity, non-negativity and partition of unity [32–35]. The approximations \( U^i_j \) at the point \((x_i, t_j)\) over subinterval \([x_i, x_{i+1}]\) can be defined as

\[
U^i_j = \sum_{k=0}^{i-1} C_k^i TB_k^i(x)
\]

where \( i = 0, 1, 2, \ldots, n \). So as to obtain the approximations to the solutions, the values of \( TB_j^i(x) \) and its derivatives at nodal points are required and these derivatives are tabulated in Table 1 [35].

Using approximate functions (6) and (7), the values at the knots of \( U^i_j \) and their derivatives up to second order are

\[
\left\{ \begin{array}{l}
(U^i_j)_1 = a_1 C_{i-1,3}^i + a_2 C_{i-2,2}^i + a_1 C_{i-1,1}^i, \\
(U^i_j)_2 = a_1 C_{i-1,3}^i + a_1 C_{i-1,2}^i, \\
(U^i_j)_3 = a_1 C_{i-1,3}^i + a_2 C_{i-1,2}^i + a_1 C_{i-1,1}^i.
\end{array} \right.
\]

The approximations for the solutions of wave equation (1) at \( t_{j+1} \) th time level can be given as [24]

\[
(U^i_{j+1})_1 = \theta g^i_{j+1} + (1 - \theta) g^i_{j} + q(x_i, t_j), \quad \theta \in [0, 1]
\]

where \( g^i_j = x^2 \partial^2(U^i_j) \) and the subscripts \( j \) and \( j + 1 \) are successive time levels, \( j = 0, 1, 2, \ldots \). By using the central finite difference discretization of the time derivatives and rearranging the equations, we obtain

\[
\frac{U^i_{j+1} - 2U^i_j + U^i_{j-1}}{\Delta t^2} = (1 - \theta) g^i_{j+1} + \theta g^i_{j+1} + q(x_i, t_j),
\]

\[
U^i_{j+1} - k^2 \partial^2 g^i_{j+1} = 2U^i_j + k^2 (1 - \theta) g^i_j - U^i_{j-1} + k^2 q(x_i, t_j),
\]

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Values of ( TB_j^i(x) ) and its derivatives.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( TB_j^i )</td>
<td>( x_i )</td>
</tr>
<tr>
<td>( \frac{\partial}{\partial x}TB_j^i )</td>
<td>( a_1 )</td>
</tr>
<tr>
<td>( \frac{\partial^2}{\partial x^2}TB_j^i )</td>
<td>( a_1 )</td>
</tr>
</tbody>
</table>

where

\[
\begin{align*}
 a_1 &= \frac{\sin^2 \left( \frac{3}{2} \right)}{\sin(h) \sin \left( \frac{h}{2} \right)}, & a_2 &= \frac{2}{1 + 3 \cos(h)}, & a_3 &= \frac{3}{4 \sin \left( \frac{3}{2} \right)}, & a_4 &= \frac{3}{4 \sin \left( \frac{3}{2} \right)} \\
 a_5 &= \frac{3(1 + 3 \cos(h))}{16 \sin^2 \left( \frac{3}{2} \right) \cos \left( \frac{3}{2} \right) \cos \left( \frac{3}{2} \right)}, & a_6 &= \frac{3 \cos^2 \left( \frac{3}{2} \right)}{\sin^2 \left( \frac{3}{2} \right) \cos \left( \frac{3}{2} \right) \cos \left( \frac{3}{2} \right)}
\end{align*}
\]
where \( k = \Delta t \) is the time step. It is noted that the system becomes an explicit scheme when \( \theta = 0 \), a fully implicit scheme when \( \theta = 1 \), and a Crank-Nicolson scheme when \( \theta = 1/2 \) [24]. In this paper, we use the Crank-Nicolson approach. Hence, (10) becomes

\[
U_i^{j+1} - 0.5k^2g_i^{j+1} = 2U_i^j + 0.5k^2g_i^j + k^2q(x_i,t_j) - U_i^{j-1},
\]

The initial velocity condition (2) is substituted into last term of Eq. (11) for computing \( C_i^1 \).

By central difference approximation,

\[
U_i^1 = U_i^0 - 2ke_2(x),
\]

After that, the system thus obtained for \( j \geq 1 \) on simplifying (11) after using (8) consists of \( n + 1 \) linear equations in \( n + 3 \) unknowns \( C_i^{j+1} = (C_i^{j-1}, C_i^{j-2}, C_i^{j-3}, \ldots, C_i^{j-n}) \) at the time level \( t = t_j \). The Eq. (7) is applied to the boundary conditions (3) and (4) for two additional linear equations to obtain a unique solution of the resulting system. Thus, the system becomes a matrix system of dimension \((n + 3) \times (n + 3)\) which is a tri-diagonal system that can be solved by the Thomas Algorithm [38–42].

2.1. Initial state

After the initial vectors \( C^0 \) and \( C^1 \) have been computed from the initial displacement and velocity conditions (2) respectively, the approximate solutions \( U_i^{j+1} \) at a particular time level can be calculated repeatedly by solving the recurrence relation (11) [21].

\( C^0 \) can be obtained from the initial and boundary values of the derivatives of the initial condition as follows [25,43]:

\[
\begin{align*}
\left\{ \begin{array}{l}
U_0^0 = e_1(x_1), & i = 0, \\
U_i^0 = e_1(x_i), & i = 0, 1, 2, \ldots, n, \\
U_n^0 = e_1(x_n), & i = n.
\end{array} \right.
\]

(12)

Thus the Eq. (12) yield a \((n + 3) \times (n + 3)\) matrix system for which the solution can be computed by the use of the Thomas Algorithm.

3. Stability of proposed method

In this section, the von Neumann stability method is applied for investigating the stability of the proposed scheme. This approach has been used by many researchers [7,20,21,25,43,44]. Substituting the approximate solution \( U \), their derivatives at the knots with \( q(x, t) = 0 \), into Eq. (10) yields a difference equation with variables \( C_m \) given by:

\[
\begin{align*}
(a_1 - k^2 \theta x^2 a_2) C_{m-1}^{j+1} + (a_2 - k^2 \theta x^2 a_3) C_{m-2}^{j+1} + (a_1 - k^2 \theta x^2 a_5) C_{m-1}^{j+1} &= (2a_2 + k^2 (1 - \theta) x^2 a_6) C_{m-3}^{j+1} + (2a_2 + k^2 (1 - \theta) x^2 a_6) C_{m-2}^{j+1} + (2a_1 + k^2 (1 - \theta) x^2 a_5) C_{m-1}^{j+1} \\
- (a_1 C_{m-1}^{j-1} + a_1 C_{m-2}^{j-1} + a_1 C_{m-1}^{j-1})
\end{align*}
\]

(13)

Substituting the values of \( a_i \), \( i = 1, 2, 5, 6 \) in Eq. (13) we obtain

\[
\begin{align*}
(p_1 - 6k^2 \theta x^2 p_2) C_{m-1}^{j+1} + (p_3 + 6k^2 \theta x^2 p_4) C_{m-2}^{j+1} + (p_1 - 6k^2 \theta x^2 p_2) C_{m-1}^{j+1} &= (2p_1 + 6k^2 (1 - \theta) x^2 p_2) C_{m-3}^{j+1} + (2p_3 + 6k^2 (1 - \theta) x^2 p_4) C_{m-2}^{j+1} + (2p_1 + 6k^2 (1 - \theta) x^2 p_2) C_{m-1}^{j+1} \\
- (p_1 C_{m-1}^{j-1} + p_3 C_{m-2}^{j-1} + p_1 C_{m-1}^{j-1})
\end{align*}
\]

(14)

where,

\[
\begin{align*}
p_1 &= 16 \sin^2 \left( \frac{\theta}{2} \right)(2 \cos \left( \frac{\theta}{2} \right) + \cos \left( \frac{3\theta}{2} \right))(1 + 2 \cos \left( \frac{\theta}{2} \right)), \\
p_2 &= 0.5 \sin(h) \sin \left( \frac{\theta}{2} \right) (1 + 3 \cos \left( \frac{\theta}{2} \right)) \cos \left( \frac{\theta}{2} \right) (1 + 2 \cos \left( \frac{\theta}{2} \right)), \\
p_3 &= 16 \sin(h) \sin \left( \frac{\theta}{2} \right) (2 \cos \left( \frac{\theta}{2} \right) + \cos \left( \frac{3\theta}{2} \right)), \\
p_4 &= 4 \cot^2 \left( \frac{\theta}{2} \right) \sin(h) \sin \left( \frac{\theta}{2} \right) (2 \cos \left( \frac{\theta}{2} \right) + \cos \left( \frac{3\theta}{2} \right)).
\end{align*}
\]

(15)

Simplifying it leads to

\[
w_1 C_{m-3}^{j+1} + w_2 C_{m-2}^{j+1} + w_1 C_{m-1}^{j+1} = w_3 C_{m-3}^{j+1} + w_4 C_{m-2}^{j+1} + w_3 C_{m-1}^{j+1} - p_1 C_{m-3}^{j+1} - p_3 C_{m-2}^{j+1} - p_1 C_{m-1}^{j+1}.
\]

(16)
where
\[
\begin{align*}
  w_1 &= \left( p_1 - 6k^2 \partial_x^2 p_2 \right), \\
  w_2 &= \left( p_3 + 6k^2 \partial_x^2 p_4 \right), \\
  w_3 &= \left( 2p_1 + 6k^2 (1 - \theta) \partial_x^2 p_2 \right), \\
  w_4 &= \left( 2p_3 + 6k^2 (1 - \theta) \partial_x^2 p_4 \right).
\end{align*}
\] (17)

Now on inserting the trial solutions (one Fourier mode out of the full solution) at a given point \( x_n \) is \( C_m = \delta \exp(\text{i}m\eta h) \) into Eq. (16) and rearranging the equations, \( h \) is the element size and \( i = \sqrt{-1} \) we get
\[
\begin{align*}
  w_1 \delta^2 e^{\text{i}(m-3)\eta h} + w_2 \delta^2 e^{\text{i}(m-2)\eta h} + w_1 \delta^2 e^{\text{i}(m-1)\eta h} &= w_3 \delta^2 e^{\text{i}(m-3)\eta h} + w_4 \delta^2 e^{\text{i}(m-2)\eta h} + w_3 \delta^2 e^{\text{i}(m-1)\eta h} - p_1 \delta^2 e^{\text{i}(m-3)\eta h} \\
  - p_2 \delta^2 e^{\text{i}(m-2)\eta h} - p_3 \delta^2 e^{\text{i}(m-1)\eta h},
\end{align*}
\] (18)

Dividing Eq. (18) by \( \delta^2 e^{\text{i}(m-2)\eta h} \), rearranging the equation, we get
\[
\delta^2 (w_2 + w_1 \cos(\eta h)) - \delta (w_4 + w_3 \cos(\eta h)) + (p_3 + p_1 \cos(\eta h)) = 0
\] (19)

Let
\[
A = p_3 + p_1 \cos(\eta h), \quad B = k^2 \partial_x^2 (p_4 - p_2 \cos(\eta h))
\]

Therefore, Eq. (19) can be written as
\[
\delta^2 (A + 6\theta B) - \delta (2A - 6(1 - \theta)B) + A = 0
\] (20)

Hence,
\[
\delta = \frac{(2A - 6(1 - \theta)B) \pm \sqrt{-(2A - 6(1 - \theta)B)^2 - 4(A + \theta B)A}}{2(A + 6\theta B)}
\]

Simplifying it, we obtain
\[
\delta = \frac{(A - 3(1 - \theta)B) \pm \sqrt{6AB - 9(1 - \theta)^2 B^2}}{(A + 6\theta B)}
\]

and the amplification factor \(|\delta|\) can be calculated by
\[
|\delta| = \sqrt{\left(\frac{A - 3(1 - \theta)B}{(A + 6\theta B)}\right)^2 + \left(\frac{\sqrt{6AB - 9(1 - \theta)^2 B^2}}{(A + 6\theta B)}\right)^2}
\]

After simplification, since \( A > 0, B > 0 \) and \( 0 \leq \theta \leq 1 \), we always have
\[
|\delta| = \sqrt{\frac{A}{(A + 6\theta B)}} \leq 1
\] (21)

Thus, from (21), the proposed scheme for one-dimensional wave equation is unconditionally stable since the modulus of the eigen-values must be less than one.

4. Numerical examples

In this section, the cubic trigonometric B-spline collocation method is employed to obtain the numerical solutions for one-dimensional wave equation which is subject to non-local conservation conditions given in Eqs. (1)–(4). We carry out from equation (11) by the proposed method and Intel Core™2 Duo Processor E7500, 2.93 GHz CPU with 2 GB RAM with operating system (Windows 7). The numerical implementation is carried out in Mathematica 7. Several numerical examples are discussed in this section to demonstrate the capability, consistency and efficiency of the proposed trigonometric spline method with smaller storage and CPU time. Numerical results are compared with existing methods in literature and with the exact solution at the different nodal points \( x_i \) for some time levels \( t_j \) using some particular space step size \( h \) and time-step \( k \).

Example 1. Consider the following particular case of Eq. (1)
\[
\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 5,
\] (22)

subject to the following initial and boundary conditions
\[
\left\{
\begin{array}{ll}
  u(x, t = 0) = \cos(\pi x), & \frac{\partial u}{\partial x}(x, t = 0) = 0, \\
  u(x = 0, t) = \cos(\pi t), & \int_0^1 u(x, t) dx = 0.
\end{array}
\right.
\] (23)
The exact solution of this equation is \( u(x, t) = 0.5(\cos(\pi(x + t)) + \cos(\pi(x - t))) \) and this problem (22 and 23) has been taken from Dehghan [11–15], Goh et al. [24] and Caglar et al. [31]. We compare the numerical results obtained by cubic trigonometric B-spline method with finite difference methods developed in [1], method of lines (MOL) introduced in [11], TPS, MQ, CS-RBF [12], CBSFM [13], SLTM [14], Bernstein Ritz–Galerkin method [15], cubic B-spline method (CuBSM) introduced in [24] and non-polynomial spline method (NPSM) [31] which are tabulated in Tables 2–4. Due to comparison, firstly we calculate the absolute errors with \( h = 0.01 \) at time \( t = 0.25 \) because the final time \( T \) in [1, 11–15, 33] is 0.25 which are recorded in Tables 2 and 3. From these tables, we perceive that the numerical solution based on the CuTBSM is more precise as compared to both finite difference methods [1], MOL I [11], TPS, MQ, CS-TBF [12], CBSFM [13], SLTM [14] and NPSM [31]. Secondly, we compute the absolute errors with various space steps at final time \( T = 5.0 \) (the final time in [24] is 5.0) and there are recorded in Table 4. From this table, we see that the results of CuTBSM are more accurate as compared to CuBSM [24]. Thirdly, due to comparison, Fig. 1 is presented the space–time graphs of exact and approximate solutions at time \( t = 1.0 \) and it is similar to the quadratic Bernstein Ritz–Galerkin method [15]. Fig. 2 (a) and (b) also depict the greater accuracy of CuTBSM as compared to CuBSM in terms of absolute errors. It is worth noting that there is no effect in the accuracy of CuTBSM by increasing the final time \( T \) but for finite difference methods, there are usually limitations when the final time increases [11]. Fig. 3 illustrates the space–time graphs of exact and approximate solution for this example in time period \( t \in [0, 5] \). Fig. 4 depicts the space–time–error graph of approximate solution at each time step with \( h = 0.01 \).

### Table 2

Comparison results for Example 1 with \( h = 0.01 \) at \( t = 0.25 \).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.67249851196395</td>
<td>0.67249744833828</td>
<td>2.4E-03</td>
<td>5.9E-03</td>
<td>5.2E-05</td>
</tr>
<tr>
<td>0.2</td>
<td>0.57206140281768</td>
<td>0.57205781076955</td>
<td>2.5E-03</td>
<td>5.6E-03</td>
<td>5.1E-05</td>
</tr>
<tr>
<td>0.3</td>
<td>0.41562693777745</td>
<td>0.41562384683080</td>
<td>2.3E-03</td>
<td>5.9E-03</td>
<td>5.1E-05</td>
</tr>
<tr>
<td>0.4</td>
<td>0.21850801222441</td>
<td>0.2185036823198</td>
<td>2.5E-03</td>
<td>5.7E-03</td>
<td>5.3E-05</td>
</tr>
<tr>
<td>0.5</td>
<td>0.00000000000000</td>
<td>0.00000000002276</td>
<td>2.6E-03</td>
<td>6.0E-03</td>
<td>5.0E-05</td>
</tr>
<tr>
<td>0.6</td>
<td>0.21850801222441</td>
<td>0.2185036823198</td>
<td>2.4E-03</td>
<td>6.2E-03</td>
<td>5.2E-05</td>
</tr>
<tr>
<td>0.7</td>
<td>-0.41562693777745</td>
<td>-0.41562384683080</td>
<td>2.2E-03</td>
<td>5.8E-03</td>
<td>5.4E-05</td>
</tr>
<tr>
<td>0.8</td>
<td>-0.70710678118655</td>
<td>-0.70710678117266</td>
<td>2.0E-03</td>
<td>6.1E-03</td>
<td>5.4E-05</td>
</tr>
<tr>
<td>0.9</td>
<td>-0.67249851196395</td>
<td>-0.67249744832488</td>
<td>1.8E-03</td>
<td>5.9E-03</td>
<td>5.5E-05</td>
</tr>
<tr>
<td>1.0</td>
<td>-0.70710678118655</td>
<td>-0.70710678117266</td>
<td>1.7E-03</td>
<td>6.1E-03</td>
<td>5.4E-05</td>
</tr>
</tbody>
</table>

### Table 3

Absolute errors for Example 1 at \( t = 0.25 \) with \( h = 0.01 \).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>6.0E-04</td>
<td>6.1E-04</td>
<td>7.6E-04</td>
<td>3.8E-04</td>
<td>0.203</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>6.8E-05</td>
<td>7.2E-05</td>
<td>2.3E-04</td>
<td>1.6E-05</td>
<td>1.953</td>
<td></td>
</tr>
<tr>
<td>0.0001</td>
<td>2.6E-03</td>
<td>5.5E-05</td>
<td>4.55E-05</td>
<td>1.54E-05</td>
<td>2.25E-05</td>
<td>1.85E-04</td>
<td>3.82E-06</td>
<td>19.156</td>
<td></td>
</tr>
</tbody>
</table>
Table 4
Comparison results for Example 1 for various mesh sizes at $T = 5.0$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$h = 0.1$ [24]</th>
<th>Present method $h = 0.1$ CPU time (s) = 2.421</th>
<th>$h = 0.02$ [24]</th>
<th>Present method $h = 0.02$ CPU time (s) = 14.469</th>
<th>$h = 0.01$ [24]</th>
<th>Present method $h = 0.01$ CPU time (s) = 38.188</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.96E-03</td>
<td>1.94E-03</td>
<td>1.39E-04</td>
<td>8.87E-05</td>
<td>7.97E-05</td>
<td>2.27E-05</td>
</tr>
<tr>
<td>0.2</td>
<td>2.87E-03</td>
<td>2.94E-03</td>
<td>2.09E-04</td>
<td>1.30E-04</td>
<td>1.21E-04</td>
<td>3.13E-04</td>
</tr>
<tr>
<td>0.3</td>
<td>2.69E-03</td>
<td>2.87E-03</td>
<td>1.99E-04</td>
<td>1.20E-04</td>
<td>1.15E-04</td>
<td>3.09E-05</td>
</tr>
<tr>
<td>0.4</td>
<td>1.60E-03</td>
<td>1.77E-03</td>
<td>1.19E-04</td>
<td>7.14E-05</td>
<td>6.89E-05</td>
<td>1.82E-05</td>
</tr>
<tr>
<td>0.5</td>
<td>3.45E-13</td>
<td>5.09E-12</td>
<td>6.18E-14</td>
<td>5.51E-12</td>
<td>2.03E-13</td>
<td>8.63E-13</td>
</tr>
<tr>
<td>0.6</td>
<td>1.60E-03</td>
<td>1.77E-03</td>
<td>1.19E-04</td>
<td>7.14E-05</td>
<td>6.89E-05</td>
<td>1.82E-05</td>
</tr>
<tr>
<td>0.7</td>
<td>2.69E-03</td>
<td>2.87E-03</td>
<td>1.99E-04</td>
<td>1.20E-04</td>
<td>1.15E-04</td>
<td>3.09E-05</td>
</tr>
<tr>
<td>0.8</td>
<td>2.87E-03</td>
<td>2.94E-03</td>
<td>2.09E-04</td>
<td>1.30E-04</td>
<td>1.21E-04</td>
<td>3.33E-05</td>
</tr>
<tr>
<td>0.9</td>
<td>1.96E-03</td>
<td>1.94E-03</td>
<td>1.39E-04</td>
<td>8.87E-05</td>
<td>7.97E-05</td>
<td>2.27E-05</td>
</tr>
</tbody>
</table>

Fig. 1. Comparison of space–time graph of exact and approximate solutions for Example 1 with $h = 0.05$, $k = 0.01$.

Fig. 2. Absolute errors of CuBSM [24] and CuTBSM methods for Example 1 with $h = 0.01$ (a) at each time step over $t \in [0, 5]$, (b) at $T = 5.0$.

Fig. 3. Space–time graphs for Example 1 with $h = 0.01$ at $T = 5.0$. 
In order to calculate the order of convergence, $p$ of the proposed method numerically, we use the following Log ratio formula \cite{7, 43, 44}
\begin{equation}
p = \frac{\log(L_{\infty}(n_i)) - \log(L_{\infty}(n_{i+1}))}{\log(n_i) - \log(n_{i+1})},
\end{equation}
where $L_{\infty} = \max \limits_i |u_i^{\text{exact}} - u_i^{\text{num}}|$ is the absolute error and $L_{\infty}(n_i)$ and $L_{\infty}(n_{i+1})$ are the absolute errors at number of partitions $n_i$ and $n_{i+1}$ respectively and relative $L_2$ error norms are calculated by
\begin{equation}
L_2 = \sqrt{\frac{\sum_{i=1}^{n} |u_i^{\text{exact}} - u_i^{\text{num}}|^2}{\sum_{i=1}^{n} |u_i^{\text{exact}}|^2}},
\end{equation}
where $u_i^{\text{num}}$ is the numerical solution at $(x_i, T)$ using $n$ mesh intervals and $u_i^{\text{exact}}$ is the exact solution at $(x_i, T)$. The order of convergence, relative error and ratio of this example are calculated by given formula in Eqs. (24) and (25) and are tabulated in Table 5. From this table the order of convergence in this example is shown to be equal to two. Here, we only compare the CPU time, absolute error $L_{\infty}$ and relative error $L_2$ for the numerical solution of one-dimensional wave equation (1) with different space size and time step size and recorded in Tables 3 and 5. The proposed scheme requires a smaller storage and CPU time than traditional cubic B-spline collocation method therefore it is more efficient and accurate.

Example 2. Consider the Eqs. (1)–(4) with $L = 1$ and $T = 4.0$,
\begin{equation}
\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 4,
\end{equation}
subject to following conditions
\begin{equation}
\begin{align*}
u(x, t = 0) &= 0, & \frac{\partial u}{\partial t}(x, t = 0) &= \pi \cos(\pi x), \\
u(x = 0, t) &= \sin(\pi t), & \int_0^1 u(x, t)dx &= 0.
\end{align*}
\end{equation}
This problem \cite{26, 27} is from Dehghan \cite{11, 12, 14, 15} and Khuri and Sayfy \cite{7} and the exact solution of this equation is $u(x, t) = \cos(\pi x) \sin(\pi t)$. We compare the numerical results obtained by cubic trigonometric B-spline method with finite difference methods developed in \cite{1}, B-spline finite element method (BSFEM) introduced in \cite{7}, method of lines (MOL) introduced in \cite{11}, TPS, MQ, CS-RBF \cite{12}, SLTM \cite{14} and Bernstein Ritz-Galerkin method \cite{15} are tabulated in Tables 6–10.

For comparison with existing schemes, firstly we compute the absolute errors with $h = 0.01$ at time $t = 0.5$ (the final time $T$ in \cite{11, 12, 14, 15} is 0.5) at each point in space and these are given in Table 6. In Table 7 we report the absolute errors of proposed method for various values of the time step size with $h = 0.01$ with CPU time for each and compared with different existing methods. From these tables, we see that the numerical solution based on the CuTBSM is more accurate as compared to the other methods.

**Table 5**

<table>
<thead>
<tr>
<th>$t = 2.0$ $h$</th>
<th>$L_2$</th>
<th>$L_{\infty}$</th>
<th>CPU time (s)</th>
<th>Ratio</th>
<th>$t = 4.0$ $L_2$</th>
<th>$L_{\infty}$</th>
<th>CPU time (s)</th>
<th>Ratio</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>6.85E-03</td>
<td>8.39E-03</td>
<td>0.454</td>
<td></td>
<td>1.25E-02</td>
<td>1.53E-02</td>
<td>0.937</td>
<td></td>
<td>2.9688</td>
</tr>
<tr>
<td>1/8</td>
<td>1.02E-03</td>
<td>1.10E-03</td>
<td>0.797</td>
<td></td>
<td>1.80E-03</td>
<td>1.95E-03</td>
<td>1.563</td>
<td></td>
<td>2.2237</td>
</tr>
<tr>
<td>1/12</td>
<td>4.41E-04</td>
<td>4.84E-04</td>
<td>1.188</td>
<td></td>
<td>7.58E-04</td>
<td>7.93E-04</td>
<td>2.297</td>
<td></td>
<td>1.9699</td>
</tr>
<tr>
<td>1/16</td>
<td>2.49E-04</td>
<td>2.65E-04</td>
<td>1.515</td>
<td></td>
<td>4.27E-04</td>
<td>4.50E-04</td>
<td>3.062</td>
<td></td>
<td>2.0078</td>
</tr>
<tr>
<td>1/20</td>
<td>1.60E-04</td>
<td>1.67E-04</td>
<td>1.921</td>
<td></td>
<td>2.0218</td>
<td>2.75E-04</td>
<td>2.87 E-04</td>
<td></td>
<td>1.4490</td>
</tr>
<tr>
<td>1/24</td>
<td>1.11E-04</td>
<td>1.15E-04</td>
<td>2.359</td>
<td></td>
<td>1.3660</td>
<td>1.91E-04</td>
<td>4.719</td>
<td></td>
<td>2.0342</td>
</tr>
<tr>
<td>1/28</td>
<td>8.20E-05</td>
<td>8.45E-05</td>
<td>2.812</td>
<td></td>
<td>1.41E-04</td>
<td>1.45E-04</td>
<td>5.625</td>
<td></td>
<td>2.0296</td>
</tr>
</tbody>
</table>
Table 6
Comparison results for Example 2 with $h = 0.01$ at $t = 0.5$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.00000000</td>
<td>0.99999999</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>0.1</td>
<td>0.95105651</td>
<td>0.95102104</td>
<td>1.5E-03</td>
<td>5.0E-03</td>
<td>3.0E-05</td>
</tr>
<tr>
<td>0.2</td>
<td>0.89091699</td>
<td>0.88959945</td>
<td>1.4E-03</td>
<td>5.2E-03</td>
<td>3.0E-05</td>
</tr>
<tr>
<td>0.3</td>
<td>0.58778525</td>
<td>0.58772603</td>
<td>1.7E-03</td>
<td>5.2E-03</td>
<td>3.2E-05</td>
</tr>
<tr>
<td>0.4</td>
<td>0.30901699</td>
<td>0.30897726</td>
<td>1.6E-03</td>
<td>5.5E-03</td>
<td>3.0E-05</td>
</tr>
<tr>
<td>0.5</td>
<td>0.00000000</td>
<td>0.00000000</td>
<td>1.5E-03</td>
<td>5.3E-03</td>
<td>3.0E-05</td>
</tr>
<tr>
<td>0.6</td>
<td>–</td>
<td>–</td>
<td>1.5E-03</td>
<td>5.1E-03</td>
<td>3.4E-05</td>
</tr>
<tr>
<td>0.7</td>
<td>–</td>
<td>–</td>
<td>1.9E-03</td>
<td>5.6E-03</td>
<td>3.1E-05</td>
</tr>
<tr>
<td>0.8</td>
<td>–</td>
<td>–</td>
<td>1.8E-03</td>
<td>5.5E-03</td>
<td>3.2E-05</td>
</tr>
<tr>
<td>0.9</td>
<td>–</td>
<td>–</td>
<td>1.7E-03</td>
<td>5.5E-03</td>
<td>3.4E-05</td>
</tr>
<tr>
<td>1.0</td>
<td>–</td>
<td>–</td>
<td>1.6E-03</td>
<td>5.3E-03</td>
<td>3.2E-05</td>
</tr>
</tbody>
</table>

– – – – – 1.11E-16

Table 7
Absolute errors for Example 2 at $t = 0.5$ with $h = 0.01$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>2.12E-03</td>
<td>2.11E-03</td>
</tr>
<tr>
<td>0.001</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>2.15E-04</td>
<td>2.12E-04</td>
</tr>
<tr>
<td>0.0001</td>
<td>1.90E-03</td>
<td>3.40E-06</td>
<td>1.60E-04</td>
<td>2.22E-16</td>
<td>1.43E-05</td>
<td>2.11E-05</td>
<td>1.43E-05</td>
<td>1.11E-16</td>
<td>38.594</td>
</tr>
</tbody>
</table>

Table 8
Comparison results for Example 2 with $h = 0.05$ at $t = 0.5$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>Approximate solution</th>
<th>BSFEM [7]</th>
<th>Present Method (CPU time (s) = 4.860)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.00000000</td>
<td>0.000000</td>
<td>2.22E-16</td>
</tr>
<tr>
<td>0.2</td>
<td>0.80854806</td>
<td>6.5E-04</td>
<td>4.68E-04</td>
</tr>
<tr>
<td>0.4</td>
<td>0.30875306</td>
<td>3.7E-04</td>
<td>2.63E-04</td>
</tr>
<tr>
<td>0.6</td>
<td>-0.30875306</td>
<td>3.7E-04</td>
<td>2.63E-04</td>
</tr>
<tr>
<td>0.8</td>
<td>-0.80854806</td>
<td>6.5E-04</td>
<td>4.68E-04</td>
</tr>
<tr>
<td>1.0</td>
<td>-0.99999999</td>
<td>5.0E-07</td>
<td>1.34E-10</td>
</tr>
</tbody>
</table>

Table 9
Computational results for Example 2 with $h = 0.1$, $p = 0.01$ and $k = p^2$ [7].

<table>
<thead>
<tr>
<th>$x$</th>
<th>$t = 0.1$ BSFEM [7]</th>
<th>Present method CPU time (s) = 0.031</th>
<th>$t = 0.1$ BSFEM [7]</th>
<th>Present method CPU time (s) = 0.516</th>
<th>$t = 0.5$ BSFEM [7]</th>
<th>Present method CPU time (s) = 2.500</th>
<th>$t = 1.0$ BSFEM [7]</th>
<th>Present method CPU time (s) = 4.781</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>5.2E-06</td>
<td>6.93E-18</td>
<td>0.0000</td>
<td>5.55E-17</td>
<td>0.0000</td>
<td>2.22E-16</td>
<td>0.0000</td>
<td>6.56E-22</td>
</tr>
<tr>
<td>0.2</td>
<td>4.2E-06</td>
<td>8.08E-08</td>
<td>1.1E-03</td>
<td>3.16E-05</td>
<td>9.7E-03</td>
<td>1.76E-03</td>
<td>1.6E-02</td>
<td>1.77E-04</td>
</tr>
<tr>
<td>0.4</td>
<td>1.6E-06</td>
<td>3.33E-08</td>
<td>4.3E-04</td>
<td>1.22E-05</td>
<td>6.5E-03</td>
<td>9.83E-04</td>
<td>8.6E-03</td>
<td>7.45E-05</td>
</tr>
<tr>
<td>0.6</td>
<td>4.2E-06</td>
<td>3.33E-08</td>
<td>4.3E-04</td>
<td>1.22E-05</td>
<td>6.5E-03</td>
<td>9.83E-04</td>
<td>8.6E-03</td>
<td>7.45E-05</td>
</tr>
<tr>
<td>0.8</td>
<td>5.2E-06</td>
<td>8.08E-08</td>
<td>1.1E-03</td>
<td>3.16E-05</td>
<td>9.7E-03</td>
<td>1.76E-03</td>
<td>1.6E-02</td>
<td>1.77E-04</td>
</tr>
<tr>
<td>1.0</td>
<td>5.2E-06</td>
<td>3.44E-15</td>
<td>0.0000</td>
<td>1.58E-12</td>
<td>0.0000</td>
<td>4.35E-11</td>
<td>0.0000</td>
<td>5.12E-10</td>
</tr>
</tbody>
</table>
to finite difference methods [1], MOL I [11], TPS, MQ, CS-RBF [12] and SLTM [14]. Secondly, we work out the absolute errors with two time step (i) \( k = p^2 \) (ii) \( k = (e^p - 1)^2 \) where \( p = 0.01 \) and space step size \( h = 0.05 \) at time \( t = 0.5 \) (the final time \( T \) in [7] is 0.5) are given in Tables 8 and 9, respectively. It is clearly shown from these tables that the results of CuTBSM are more accurate as compared to BSFEM [7]. We also compare the absolute errors with BSFEM [7] when they are calculated at time \( t = 0.5 \), \( h = 0.05 \) and these are recorded in Table 10. It is worth noting that the results obtained using CuTBSM are slightly accurate as compared to BSFEM [7]. In Table 11 we report the CPU time, absolute error \( L_1 \) and relative error \( L_2 \) for the numerical solution of one-dimensional wave equation (1)-(4) with different space size. The order of convergence and ratio of this example are also calculated and they are tabulated in same Table 11. From this table the order of convergence is shown to be equal to two. Thirdly, for the purpose of comparison, Fig. 5 is presented the space–time graphs of exact and approximate

Table 10
Computational results for Example 2 with \( h = 0.1 \), \( p = 0.01 \) and \( k = (e^p - 1)^2 \) [7].

<table>
<thead>
<tr>
<th>( x )</th>
<th>( t = 0.01 )</th>
<th>Present method CPU time (s)</th>
<th>BSFEM [7]</th>
<th>( t = 0.1 )</th>
<th>Present method CPU time (s)</th>
<th>BSFEM [7]</th>
<th>( t = 0.5 )</th>
<th>Present method CPU time (s)</th>
<th>BSFEM [7]</th>
<th>( t = 1.0 )</th>
<th>Present method CPU time (s)</th>
<th>BSFEM [7]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>5.2E-06</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.11E-16</td>
<td>0.0000</td>
<td>2.73E-21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>4.2E-06</td>
<td>2.94E-08</td>
<td>1.1E-03</td>
<td>7.1E-03</td>
<td>1.72E-03</td>
<td>1.00E-04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>1.6E-06</td>
<td>1.21E-08</td>
<td>4.1E-04</td>
<td>5.0E-03</td>
<td>9.57E-04</td>
<td>3.35E-05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>4.2E-06</td>
<td>1.21E-08</td>
<td>4.1E-04</td>
<td>5.0E-03</td>
<td>9.57E-04</td>
<td>3.35E-05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>5.2E-06</td>
<td>2.94E-08</td>
<td>1.1E-03</td>
<td>7.1E-03</td>
<td>1.72E-03</td>
<td>1.00E-04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>5.2E-06</td>
<td>2.46E-13</td>
<td>0.0000</td>
<td>2.68E-09</td>
<td>0.0000</td>
<td>1.17E-08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11
Relative errors, absolute errors, ratio and order of convergence of the proposed scheme for \( U(x, t) \) at \( t = 0.5 \) for Example 2.

<table>
<thead>
<tr>
<th>( h )</th>
<th>( L_2 ) [7]</th>
<th>( p ) [7]</th>
<th>( L_2 )</th>
<th>( L_\infty )</th>
<th>Ratio</th>
<th>CPU time (s)</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>0.014885504</td>
<td>2.0</td>
<td>0.0113414</td>
<td>0.0092602</td>
<td>4.0</td>
<td>1.140</td>
<td>2.0</td>
</tr>
<tr>
<td>1/8</td>
<td>0.003671173</td>
<td>2.0</td>
<td>0.0027691</td>
<td>0.0025302</td>
<td>2.2</td>
<td>2.000</td>
<td>2.0</td>
</tr>
<tr>
<td>1/12</td>
<td>0.001651035</td>
<td>1.9</td>
<td>0.0012430</td>
<td>0.0011780</td>
<td>1.7</td>
<td>2.843</td>
<td>1.9</td>
</tr>
<tr>
<td>1/16</td>
<td>0.000952668</td>
<td>1.9</td>
<td>0.0007165</td>
<td>0.0006910</td>
<td>1.5</td>
<td>3.812</td>
<td>1.9</td>
</tr>
<tr>
<td>1/20</td>
<td>0.000630767</td>
<td>1.9</td>
<td>0.0004739</td>
<td>0.0004617</td>
<td>1.4</td>
<td>4.813</td>
<td>1.8</td>
</tr>
<tr>
<td>1/24</td>
<td>–</td>
<td>–</td>
<td>0.0003423</td>
<td>0.0003358</td>
<td>–</td>
<td>5.922</td>
<td>–</td>
</tr>
</tbody>
</table>

Fig. 5. Comparison of space–time graph of exact and approximate solutions for Example 2 with \( h = 0.05 \), \( k = 0.01 \).

Fig. 6. Absolute errors at each time step over \( t \in [0, 4] \) for Example 2 with \( h = 0.05 \).
solutions at time $t = 1.0$ and it is noticed that it is similar to the quadratic Bernstein Ritz-Galerkin method [15]. Fig. 6 depicts the absolute errors over the interval $[0, 4]$ with space step size $h = 0.05$ obtained by using CuTBSM. Fig. 7 demonstrates the space–time graphs of numerical solution and exact solution in time period $t \in [0, 4]$.

**Example 3.** Consider the following one-dimensional wave equation from [7,12–15]

\[
\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = \left( \pi^2 + \frac{1}{4} \right) e^{\frac{-1}{2} t} \sin(\pi x), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 3,
\]

subject to following initial and boundary constraints

\[
\begin{aligned}
&u(x, t = 0) = \sin(\pi x), \quad \frac{\partial u}{\partial t}(x, t = 0) = -\frac{1}{2} \sin(\pi x), \\
&u(x = 0, t) + \frac{\partial u}{\partial x}(x = 0, t) = \pi e^{\frac{-1}{2} t}, \quad \int_0^1 u(x, t) \, dx = \frac{\pi}{6} e^{\frac{-1}{2} t}.
\end{aligned}
\]

The exact solution of this problem is $u(x, t) = e^{\frac{-1}{2} t} \sin(\pi x)$. In Table 12 we show the absolute error obtained for this problem using CuTBSM at different nodes at time $t = 1.0$ with space step size $h = 0.01$. Absolute errors recorded in Table 13 corroborated that numerical results of the CuTBSM are in good agreement with given exact solutions and more accurate as compared to BSFEM [7], CBSFM [13] and SLTM [14]. For the purpose of comparison, in Table 14 we record the absolute errors using CuTBSM at $t = 0.5$ and $1.0$ with $h = 0.01$ and $k = 0.0001$ and found that they are in good agreement with given exact solutions and more accurate as compared to TPS, MQ, CS-RBF [12]. Numerical results in Table 15 confirm that the order of convergence

![Space–time graph of exact and approximate solution for Example 2 with $h = 0.05$.](image)

| Table 12 | Absolute errors at various nodes for Example 3 with $h = 0.01$ at $t = 1.0$. |
|---|---|---|---|
| $x$ | Exact solution | Approximate solution | Present method |
| 0.1 | 0.187428 | 0.186311 | 1.11E-03 |
| 0.2 | 0.356509 | 0.355484 | 1.02E-03 |
| 0.3 | 0.490693 | 0.489739 | 9.54E-04 |
| 0.4 | 0.576844 | 0.575945 | 8.99E-04 |
| 0.5 | 0.606530 | 0.605674 | 8.56E-04 |
| 0.6 | 0.574844 | 0.576028 | 8.16E-04 |
| 0.7 | 0.490693 | 0.489919 | 7.74E-04 |
| 0.8 | 0.366509 | 0.355791 | 7.18E-04 |
| 0.9 | 0.187428 | 0.186786 | 6.41E-04 |

| Table 13 | Absolute error for Example 3 with $h = 0.025$ at $t = 1.0$. |
|---|---|---|---|---|---|
| BSFEM [7] | CBSFM [13] ($M = 2$) | SLTM [14] ($m = n = 4$) | SLTM [14] ($m = n = 5$) | CPU time(s) | Present Method |
| 2.30E-02 | 1.50E-02 | 5.00E-03 | 3.60E-03 | 2.219 | 2.45E-03 |

| Table 14 | Absolute error for Example 3 with $h = 0.01$, $k = 0.0001$ at different time levels. |
|---|---|---|---|---|---|
| 0.5 | 3.7977E-03 | 1.3371E-03 | 2.7790E-02 | 38.937 | 3.340E-05 |
| 1.0 | 6.8404E-03 | 2.3794E-03 | 5.0746E-02 | 77.484 | 8.0628E-04 |
of present method is approximately equal to two. These tables allow that the present method requires a much smaller storage and CPU time to get a smaller error and therefore it is more efficient method. Fig. 8 illustrates the approximate solution of this problem at various time levels. Fig. 9 depicts the absolute errors and space–time error graph of approximate solution.

![Table 15](image)

**Table 15**
Relative errors, absolute errors, ratio and order of convergence of the proposed scheme for Example 3 at $t = 0.5$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$L_2$</th>
<th>$L_{\infty}$</th>
<th>Ratio</th>
<th>$p$</th>
<th>CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>0.05639480</td>
<td>0.03440552</td>
<td>3.8</td>
<td>1.94</td>
<td>0.109</td>
</tr>
<tr>
<td>1/8</td>
<td>0.01357585</td>
<td>0.00899331</td>
<td>2.2</td>
<td>1.91</td>
<td>0.203</td>
</tr>
<tr>
<td>1/12</td>
<td>0.00623721</td>
<td>0.00414931</td>
<td>1.7</td>
<td>1.85</td>
<td>0.296</td>
</tr>
<tr>
<td>1/16</td>
<td>0.00364373</td>
<td>0.00244299</td>
<td>1.5</td>
<td>1.76</td>
<td>0.391</td>
</tr>
<tr>
<td>1/20</td>
<td>0.00245192</td>
<td>0.00165117</td>
<td>1.4</td>
<td>1.66</td>
<td>0.500</td>
</tr>
<tr>
<td>1/24</td>
<td>0.00180727</td>
<td>0.00122046</td>
<td>–</td>
<td>–</td>
<td>0.610</td>
</tr>
</tbody>
</table>

![Fig. 8](image)

**Fig. 8.** Approximate solutions for Example 3 with $h = 0.025$ at various time levels.

![Fig. 9](image)

**Fig. 9.** Absolute errors of Example 3 at each time level with $h = 0.025$ over $t \in [0, 3]$.

![Fig. 10](image)

**Fig. 10.** Comparison of space–time graph of exact and approximate solutions for Example 3 at $t = 1.0$ with $h = 0.025$, $k = 0.001$. 


obtained by present method. They are more accurate as compared to CuBSM [24] as well as BSFEM [7]. For the purpose of comparison, Fig. 10 depicts the space–time graphs of exact and approximate solutions at time \( t = 1.0 \). From this figure, it is clear that it looks similar to the quadratic Bernstein Ritz-Galerkin method [15]. In Fig. 11 we show the space–time graph of exact and approximate solution with \( h = 0.025 \) for \( t \in [0, 3] \).

**Example 4.** Consider the following one-dimensional wave equation from [17]

\[
\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = x^2 e^t - 2e^t, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 0.5,
\]

subject to following initial and boundary constraints

\[
\begin{align*}
\{ & u(x, t = 0) = x^2, \quad \frac{\partial u}{\partial x}(x, t = 0) = x^2, \\
& u(x = 0, t) = 0, \quad \int_0^1 u(x, t) dx = \frac{1}{2} e^t.
\end{align*}
\]

**Table 16**

<table>
<thead>
<tr>
<th>( x )</th>
<th>Exact solution</th>
<th>Approximate solution [26]</th>
<th>CuBSM [24]</th>
<th>Approximate solution</th>
<th>Present method CPU time (s) = 3.906</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000000</td>
<td>0.000000</td>
<td>9.71E-18</td>
<td>0.000000</td>
<td>5.69E-17</td>
</tr>
<tr>
<td>0.1</td>
<td>0.016487</td>
<td>0.016435</td>
<td>5.19E-05</td>
<td>0.016472</td>
<td>1.44E-05</td>
</tr>
<tr>
<td>0.2</td>
<td>0.065948</td>
<td>0.065861</td>
<td>8.78E-05</td>
<td>0.065926</td>
<td>2.27E-05</td>
</tr>
<tr>
<td>0.3</td>
<td>0.148384</td>
<td>0.148277</td>
<td>1.08E-04</td>
<td>0.148360</td>
<td>2.45E-05</td>
</tr>
<tr>
<td>0.4</td>
<td>0.263795</td>
<td>0.263681</td>
<td>1.13E-04</td>
<td>0.263775</td>
<td>1.95E-05</td>
</tr>
<tr>
<td>0.5</td>
<td>0.412180</td>
<td>0.412074</td>
<td>1.05E-04</td>
<td>0.412172</td>
<td>8.05E-06</td>
</tr>
<tr>
<td>0.6</td>
<td>0.593539</td>
<td>0.593445</td>
<td>9.51E-05</td>
<td>0.593545</td>
<td>6.02E-06</td>
</tr>
<tr>
<td>0.7</td>
<td>0.807873</td>
<td>0.807779</td>
<td>9.43E-05</td>
<td>0.807890</td>
<td>1.73E-05</td>
</tr>
<tr>
<td>0.8</td>
<td>1.055181</td>
<td>1.055074</td>
<td>1.06E-04</td>
<td>1.055205</td>
<td>2.35E-05</td>
</tr>
<tr>
<td>0.9</td>
<td>1.335464</td>
<td>1.335328</td>
<td>1.35E-04</td>
<td>1.335486</td>
<td>2.21E-05</td>
</tr>
<tr>
<td>1.0</td>
<td>1.648721</td>
<td>1.648536</td>
<td>1.84E-04</td>
<td>1.648731</td>
<td>1.05E-05</td>
</tr>
</tbody>
</table>

\[
\text{Fig. 11.} \text{ Space–time graph of exact and approximate solution for Example 3 with } h = 0.025.
\]

\[
\text{Fig. 12.} \text{ Approximate solutions for Example 4 with } h = 0.05 \text{ at different time levels.}
\]
The exact solution of this problem is \( u(x, t) = x^2 e^t \). In Table 16 we report the absolute errors obtained for this problem using CuTBSM at different nodes at time \( t = 0.5 \) with space step size \( h = 0.01 \) and \( k = 0.001 \). This allows that the CuTBSM requires a much smaller storage and CPU time than CuBSM [24] to get a smaller error and therefore it is more efficient method. Fig. 12 depicts the approximate solution of this problem at various time levels. Fig. 13 also depicts the greater accuracy of CuTBSM as compared to CuBSM [24] in terms of absolute errors. In Fig. 14, we show the space–time graph of approximate solution and exact solution of this problem with \( h = 0.05 \) at \( t \in [0, 1] \).

5. Concluding remarks

In this paper, we have proposed a new cubic trigonometric B–spline collocation method for solving the one-dimensional wave equation with known initial condition and non-local conservation condition instead of classical boundary conditions. A finite difference approach was used to discretize the time derivatives and cubic trigonometric B-spline was used to interpolate the solutions at each time level. It was observed that sometimes the accuracy of solution can reduce as time increases due to the time truncation errors of time derivative term [44]. The numerical results shown in Tables 2–16 and Figs. 1–14 indicate the reliable results obtained. The obtained solution to the wave equation for various time levels has been compared with the exact solution by calculating \( L_\infty \) and \( L_2 \) errors. The order of convergence was shown to be approximately equal to two. In addition to its simple and straightforward application, an advantage of using the cubic trigonometric B-spline method outlined in this paper is that it can give accurate solutions at any intermediate point in the space direction if with exact solution is a trigonometric function. The CuTBSM has approximated the solution more accurate results as compared to some finite difference schemes such as 3 point explicit approach, the weighted finite difference schemes, the optimal explicit approach, method of lines (MOL I), SLTM [14] and some spline collocation methods like BSFEM [7], TPS, MQ, CS-RBF [12], CBSFM [13], Bernstein Ritz–Galerkin method [15], CuBSM [24] and NPSM [31]. The new CuTBSM was also found to be superior to CuBSM [24] with respect to CPU time and smaller storage. However, it was less accurate as compared to method of lines (MOL II) because the authors used fourth-order approximation in second order derivative rather than the second order approximation, used in proposed method in \( x \) and they solved the resulting ordinary differential equation (ODE) system by using Built-in Mat lab solver “ode45” which is based on the explicit Runge–Kutta (4, 5) formula, the Dormand–Prince pair.
Acknowledgments

This study was fully supported by FRGS Grant of No. 203/PMATHS/6711324 from the School of Mathematical Sciences, Universiti Sains Malaysia, Penang, Malaysia. The first author was supported by Post Doctorate Fellowship from School of Mathematical Sciences, Universiti Sains Malaysia, Penang, Malaysia, during a part of the time in which the research was carried out. The authors are also indebted to the anonymous reviewers for their helpful, valuable comments and suggestions in the improvement of this manuscript.

References