Hyperbolic B-Spline Function-Based Differential Quadrature Method for the Approximation of 3D Wave Equations

Mohammad Tamsir *, Mutum Zico Meetei ° and Ahmed H. Msmali

Department of Mathematics, College of Science, Jazan University, Jazan 45142, Saudi Arabia
* Correspondence: mtamsir@jazanu.edu.sa

Abstract: We propose a differential quadrature method (DQM) based on cubic hyperbolic B-spline basis functions for computing 3D wave equations. This method converts the problem into a system of ODEs. We use an optimum five-stage and order four SSP Runge-Kutta (SSPRK-(5,4)) scheme to solve the obtained system of ODEs. The matrix stability analysis is also investigated. The accuracy and efficiency of the proposed method are demonstrated via three numerical examples. It has been found that the proposed method gives more accurate results than the existing methods. The main purpose of this work is to present an accurate, economically easy-to-implement, and stable technique for solving hyperbolic partial differential equations.

Keywords: 3D wave equations; DQM; hyperbolic B-spline functions; SSPRK-(5,4); stability analysis

MSC: 65Nxx; 35-xx

1. Introduction

Some of the existing numerical approaches for solving wave equations as well as fractional wave equations involve the temporal extrapolation method, finite difference method (FDM), finite volume method (FVM), finite element method (FEM), and boundary element method (BEM) [1–5]. It has been noticed that the most popularly used numerical approaches for solving 3D wave equations are based on the FDM [6,7]. In short, the FDM is utilized to handle the time derivative, and the space derivatives are discretized by other numerical techniques. In particular, the radial basis and B-spline basis functions-based collocation methods are extensively applied for solving 3D wave equations. Ranocha et al. [8] set up fully discrete conservative techniques for various dissemiative wave equations. Recently, Wang et al. [9] presented radial basis function-based single-step mesh free technique for 2D variable coefficients wave equation. Bakushinsky and Leonov [10] presented a fast Fourier transform-based algorithm to solve the 3D wave inverse problem in a cylindrical system.

In recent decades, wave equations have been approximated by numerous researchers. Dehghan [11] approximated the solution of 1D hyperbolic PDEs with nonlocal boundary specifications, while in [12], the author used ADI, fully implicit, fully explicit FD methods, and the Barakat and Clark type explicit formulae to approximate the 2D Schrodinger equation. Mohanty and Gopal [13] presented an off-step discretization-based technique for the approximation of 3D wave equations. Titarev and Toro [14] implemented fourth-order accurate ADER schemes for 3D hyperbolic systems. Zhang et al. [15] proposed an improved element-free Galerkin (EFG) method, while EFG method and Meshless local Petrov-Galerkin (MLPG) method have been proposed by Shivanian [16]. Recently, Shukla et al. [17] proposed an Expo-MCBDQM to approximate the aforementioned equations.

Bellman et al. [18] introduced DQM. DQM based on various basis functions has been presented to solve several PDEs such as sinc DQM [19], Fourier expansion based DQM [20], harmonic DQM [21], quintic B-spline DQM [22] and many more. The authors of [23–26] proposed a cubic B-spline (CB-spline) based DQM for the Burgers’
Burgers’, Sine-Gordon, and advection-diffusion equations, respectively while the authors of [27] proposed an exponential CB-spline based DQM for the Burgers’ equation. The authors of [28] proposed a polynomial differential quadrature method to solve the two-dimensional Sine-Gordon equation. Korkmaz and Dag [29] proposed CB-Spline based DQMs to simulate the boundary forced RLW equation, while in [30], they proposed Quartic as well as quintic B-spline based differential quadrature methods for the advection-diffusion equation. Jiwari [31] proposed Lagrange interpolation as well as CB-spline based DQMs to solve the hyperbolic PDEs. The authors in [32] presented a new cubic B-spline-based semi-analytical method for solving 3D anisotropic convention-diffusion-reaction problems. Ali et al. [33] considered nonlinear spin dynamics in Heisenberg ferromagnetic spin chain (2+1)-dimensional Broer-Kaup-Kupershmit and Drinfel’d-Sokolow-Wilson equations [34].

The rest of the paper is organized as follows. In Section 2, the procedure of the cubic hyperbolic B-spline DQM is described. Section 3 examines the stability analysis of the proposed method. Section 4 demonstrates the computational results. Finally, Section 5 presents the conclusion of our study.

2. The Hyperbolic B-Spline Differential Quadrature Method

In this section, we consider the problem (1)–(3) when \(a_1, a_2, \delta, f(u), p\) are known and \(u\) is to be evaluated. We apply the cubic hyperbolic B-spline DQM to approximate 3D wave Equations (1)–(3). First, we split \(\Omega = \{(x, y, z) : 0 \leq x, y, z \leq 1\}\) into equal length mesh...
where \( u_{ijk} = u(x_i, y_j, z_k, t) \), and \( a^{(p)}_{ir}, b^{(p)}_{jr}, c^{(p)}_{kr} \) are the weighting coefficients corresponding to \( \frac{\partial^p u_{ijk}}{\partial x^p} \), \( \frac{\partial^p u_{ijk}}{\partial y^p} \) and \( \frac{\partial^p u_{ijk}}{\partial z^p} \), respectively, at time \( t \).

The cubic hyperbolic B-spline functions are given as \([39]\):

\[
H_f_i(x) = \begin{cases} 
\frac{(S_i - 2)^3}{\sinh(3h) \sinh(2h) \sinh(h)}, & [x_{i-2}, x_{i-1}], \\
\frac{-S_i(S_i - 2)^3 - S_{i+1}S_{i+2}S_i - S_{i-1}^2(S_i - 1)^3}{\sinh(3h) \sinh(2h) \sinh(h)}, & [x_{i-1}, x_i], \\
\frac{S_i(S_i - 2)^3 + S_{i+1}S_{i+2}S_i + S_{i-1}^2(S_i - 1)^3}{\sinh(3h) \sinh(2h) \sinh(h)}, & [x_i, x_{i+1}], \\
\frac{-S_{i+1}^3}{\sinh(3h) \sinh(2h) \sinh(h)}, & [x_{i+1}, x_{i+2}], \\
0, & \text{otherwise},
\end{cases}
\]

where \( S_i = \sinh(x - x_i) \) and \( h = x_{i+1} - x_i \).

The cubic hyperbolic B-spline functions \( \{H_f_0, H_f_1, \ldots, H_f_M, H_f_{M+1}\} \) form a basis over \( \Omega \). Table 1 shows the values of cubic hyperbolic B-spline functions with derivatives at the knots, where

\[
\begin{align*}
\gamma_1 &= \gamma_3 = \frac{(\sinh(h))^3}{\sinh(3h) \sinh(2h) \sinh(h)}, \\
\gamma_2 &= \frac{2 \sinh(2h) \sinh(h)^2}{\sinh(3h) \sinh(2h) \sinh(h)}, \\
\gamma_4 &= \frac{2 \sinh(3h)}{3}, \\
\gamma_5 &= \frac{-2 \sinh(3h)}{3}.
\end{align*}
\]

### Table 1. The values of cubic hyperbolic B-spline functions at the knots.

<table>
<thead>
<tr>
<th></th>
<th>( x_{i-2} )</th>
<th>( x_{i-1} )</th>
<th>( x_i )</th>
<th>( x_{i+1} )</th>
<th>( x_{i+2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H_f_i(x) )</td>
<td>0</td>
<td>( \gamma_1 )</td>
<td>( \gamma_2 )</td>
<td>( \gamma_3 )</td>
<td>0</td>
</tr>
<tr>
<td>( H_f'_i(x) )</td>
<td>0</td>
<td>( \gamma_4 )</td>
<td>0</td>
<td>( \gamma_5 )</td>
<td>0</td>
</tr>
</tbody>
</table>

Preserving the matrix system remains diagonally dominant, we modify the hyperbolic B-spline functions as:

\[
\begin{align*}
\dot{H}_f_1(x) &= H_f_1(x) + 2H_f_0(x) \\
\dot{H}_f_2(x) &= H_f_2(x) - H_f_0(x) \\
\dot{H}_f_m(x) &= H_f_m(x) \text{ for } m = 3, \ldots, M - 2 \\
\dot{H}_f_{M-1}(x) &= H_f_{M-1}(x) - H_f_{M+1}(x) \\
\dot{H}_f_M(x) &= H_f_M(x) + 2H_f_{M+1}(x)
\end{align*}
\]
where the modified cubic hyperbolic B-spline functions \( \{ \hat{H}_f, \hat{H}_f' \ldots, \hat{H}_f_{M_x} \} \) form basis over \( \Omega \). Next, in \( \frac{\partial u_{ijk}}{\partial x} \), we use the modified basis functions \( \hat{H}_f_r(x) \), \( r = 1, 2, \ldots, M_x \) in (4) to evaluate \( a_{ij}^{(1)} \). Then, Equation (4) yields

\[
\hat{H}_f'(x_i) = \sum_{q=1}^{M_x} a_{ij}^{(1)} \hat{H}_f_r(x_i), \quad i = 1, 2, \ldots, M_x,
\]

which can be written in the form of tridiagonal system as follows:

\[
A \vec{x}[i] = \vec{B}[i], \quad \text{for } i = 1, 2, \ldots, M_x,
\]

where \( A = [\hat{H}_f_{ij}] \) is the \( M_x \times M_x \) matrix given by

\[
A = \begin{bmatrix}
\gamma_2 + 2\gamma_1 & \gamma_3 & 0 & 0 & \cdots & 0 & 0 \\
\gamma_1 - \gamma_3 & \gamma_2 & \gamma_3 & 0 & \cdots & 0 & 0 \\
0 & \gamma_1 - \gamma_3 & \gamma_2 & \gamma_3 & \cdots & 0 & 0 \\
& \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & \cdots & \gamma_1 & \gamma_2 & \gamma_3 - \gamma_1 & 0 \\
0 & 0 & \cdots & 0 & \gamma_1 & \gamma_2 & \gamma_3 - \gamma_1 \\
0 & 0 & \cdots & 0 & 0 & \gamma_1 & \gamma_1 + 2\gamma_1 \\
\end{bmatrix},
\]

\[
\vec{x}[i] = [a_{i1}^{(1)}, a_{i2}^{(1)}, \ldots, a_{iM_x}^{(1)}]^T
\]

is the unknown vector and

\[
\vec{B}[1] = \begin{bmatrix}
2\gamma_5 \\
\gamma_4 - \gamma_5 \\
0 \\
0 \\
0 \\
\vdots \\
0 \\
0 \\
\end{bmatrix}, \quad \vec{B}[2] = \begin{bmatrix}
\gamma_5 \\
0 \\
\gamma_4 \\
0 \\
0 \\
\vdots \\
0 \\
0 \\
\end{bmatrix}, \quad \ldots, \quad \vec{B}[M_x - 1] = \begin{bmatrix}
0 \\
0 \\
\gamma_5 \\
0 \\
0 \\
\vdots \\
0 \\
\gamma_5 - \gamma_4 \\
2\gamma_4 \\
\end{bmatrix}.
\]

We solve the system (10) to find the weighting coefficient vector \( a_{ij}^{(1)} \). Similarly, by fixing \( x \) and \( z \) in \( \frac{\partial u_{ijk}}{\partial y} \) and using the modified cubic hyperbolic B-spline functions \( \hat{H}_f_r(y) \), \( r = 1, 2, \ldots, M_x \) in Equation (5), and fixing \( x \) and \( y \) in \( \frac{\partial u_{ijk}}{\partial z} \) and using the modified cubic hyperbolic B-spline functions \( \hat{H}_f_r(z) \), \( r = 1, 2, \ldots, M_x \) in Equation (6), we can compute \( b_{ij}^{(1)} \) and \( c_{ij}^{(1)} \).

Next, for computing \( a_{ij}^{(p)} \), \( b_{ij}^{(p)} \) and \( c_{ij}^{(p)} \), we use the Shu’s [40] recurrence relations

\[
a_{ij}^{(p)} = p \left( a_{ij}^{(1)} a_{ij}^{(p-1)} - a_{ij}^{(p-1)} \right) / x_i - x_j), \quad \text{if } j \neq i; \quad a_{ii}^{(p)} = - \sum_{j=1, j \neq i}^{M_x} a_{ij}^{(p)} \text{, for } i, j = 1, 2, \ldots, M_x,
\]

\[
b_{ij}^{(p)} = p \left( b_{ij}^{(1)} b_{ij}^{(p-1)} - b_{ij}^{(p-1)} \right) / y_i - y_j), \quad \text{if } j \neq i; \quad b_{ii}^{(p)} = - \sum_{j=1, j \neq i}^{M_x} b_{ij}^{(p)} \text{, for } i, j = 1, 2, \ldots, M_y,
\]

\[
c_{ij}^{(p)} = p \left( c_{ij}^{(1)} c_{ij}^{(p-1)} - c_{ij}^{(p-1)} \right) / z_i - z_j), \quad \text{if } j \neq i; \quad c_{ii}^{(p)} = - \sum_{j=1, j \neq i}^{M_z} c_{ij}^{(p)} \text{, for } i, j = 1, 2, \ldots, M_z.
\]
Now, using \( u_t = w, u_{tt} = w_t \) and substituting the approximated \( u_{xx}, u_{yy} \) and \( u_{zz} \) by the cubic hyperbolic B-spline DQM in problem (1) and (2), we have

\[
d rac{u_{ijk}}{dt} = w_{ijk},
\]

(14)

and

\[
d rac{w_{ijk}}{dt} = \sum_{r=1}^{M_x} a_{i}^{(2)} u_{r,jk} + \sum_{r=1}^{M_y} b_{j}^{(2)} u_{ir,k} + \sum_{r=1}^{M_z} c_{k}^{(2)} u_{ijk} - (a_1 - \delta f(u_{ijk})) w_{ijk} - a_2 u_{ijk} + \bar{P}_{ijk},
\]

(15)

where \( i = 0, 1, \cdots, M_x, \ j = 0, 1, \cdots, M_y, \ k = 0, 1, \cdots, M_z \). Using Equation (3) in the above equation, we have

\[
d rac{u_{ijk}}{dt} = w_{ijk},
\]

(16)

and

\[
d rac{w_{ijk}}{dt} = \sum_{r=2}^{M_x-1} a_{i}^{(2)} u_{r,jk} + \sum_{r=2}^{M_y-1} b_{j}^{(2)} u_{ir,k} + \sum_{r=2}^{M_z-1} c_{k}^{(2)} u_{ijk} - (a_1 - \delta f(u_{ijk})) w_{ijk} - a_2 u_{ijk} + \bar{P}_{ijk},
\]

(17)

where,

\[
\bar{P}_{ijk} = a_{i}^{(2)} u_{ijk} + a_{2i,M_x}^{(2)} u_{M_x,jk} + b_{j}^{(2)} u_{i1,k} + b_{j,M_y}^{(2)} u_{iM_y,k} + c_{k}^{(2)} u_{i1j} + c_{k,M_z}^{(2)} u_{ijM_z} + \tilde{p}_{ijk}.
\]

(18)

Finally, we apply SSPRK-(5,4) scheme [41] to solve the above systems.

3. Stability Analysis

For stability, we rewrite Equations (16) and (17) by choosing \( \alpha_1, \alpha_2 \geq 0 \) and \( \alpha_1 > \delta f \) as follows:

\[
d \frac{\vec{X}}{dt} = \vec{A} \vec{X} + \vec{\bar{X}},
\]

(19)

where, \( \vec{X} = [u \ w]^{tr}, \vec{\bar{X}} = [N \ P]^{tr} \), \( \vec{\bar{X}} = \left[ \begin{array}{c} N \ \bar{F} \ 
\end{array} \right] \), \( N \) and \( I \) are null and identity matrices, respectively. We have \( \bar{F} = \bar{F}_x + \bar{F}_y + \bar{F}_z - a_2 I \) where \( \bar{F}_x, \bar{F}_y \) and \( \bar{F}_z \) are \((M_x-2)(M_y-2)(M_z-2)\) order matrices for \( a_{i}^{(2)} \), \( b_{j}^{(2)} \) and \( c_{k}^{(2)} \), respectively, and given as follows:

\[
\vec{F}_x = \left[ \begin{array}{ccc}
\begin{array}{ccc}
a_{i+1}^{(2)} I_x & a_{i}^{(2)} I_x & \cdots \\
\vdots & \vdots & \ddots \\
a_{i,M_x}^{(2)} I_x & a_{i}^{(2)} I_x & \cdots \\
\end{array}
\end{array} \right],
\]

(20)

\[
\vec{F}_y = \left[ \begin{array}{ccc}
\begin{array}{ccc}
b_{j+1}^{(2)} I_x & b_{j}^{(2)} I_x & \cdots \\
\vdots & \vdots & \ddots \\
b_{j,M_y}^{(2)} I_x & b_{j}^{(2)} I_x & \cdots \\
\end{array}
\end{array} \right],
\]

(21)

\[
\vec{F}_z = \left[ \begin{array}{ccc}
\begin{array}{ccc}
c_{k+1}^{(2)} I_x & c_{k}^{(2)} I_x & \cdots \\
\vdots & \vdots & \ddots \\
c_{k,M_z}^{(2)} I_x & c_{k}^{(2)} I_x & \cdots \\
\end{array}
\end{array} \right],
\]

(22)
The order of null matrices \( N_y \) and \( N_z \) are \((M_y - 2)(M_z - 2)\) and \((M_z - 2)\), respectively which is same as the order of identity matrices \( I_x \) and \( I_z \).

Now, we suppose that \( \lambda_A \) be an eigenvalue of \( A \) associated with the eigenvector \((X_1, X_2)^T\), where the order of each component vector is \((M_x - 2)(M_y - 2)(M_z - 2)\). Then, we have

\[
\begin{bmatrix}
O & I \\
F & (-\alpha_1 + \delta f)I
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2
\end{bmatrix}
= \lambda_A
\begin{bmatrix}
X_1 \\
X_2
\end{bmatrix},
\]

which implies that \( IX_2 = \lambda_A X_1 \) and \( FX_1 - (\alpha_1 - \delta f)X_2 = \lambda_A X_2 \).

Thus, we have

\[
FX_1 = \lambda_A(\lambda_A + \alpha_1 - \delta f)X_1.
\]

This illustrates that \( \lambda_A(\lambda_A + \alpha_1 - \delta f) \) is the eigenvalue of \( F \) as follows:

\[
F = -\alpha_2 I + F_1.
\]

The eigenvalues of \( F_1 \) with \( h_x = h_y = h_z = 0.2, 0.1 \) and 0.05 are represented in Figure 1, where real and negative eigenvalues are observed. Equation (25) implies that all eigenvalues of \( B \) are real and negative.

![Figure 1. Eigenvalues of the matrix \( F_1 \).](image)

Now, let \( \lambda_A = x + iy \), then we have \((\lambda_A + \alpha_1 - \delta f)\lambda_A = (x^2 + (\alpha_1 - \delta f)x - y^2) + i(2x + (\alpha_1 - \delta f))y \) is negative and real, that is,

\[
x^2 + (\alpha_1 - \delta f)x - y^2 < 0 \quad \text{and} \quad 2xy + (\alpha_1 - \delta f)y = 0.
\]

From Equation (26), we conclude that \( x = -0.5(\alpha_1 - \delta f) \) if \( y \neq 0 \) and \( x < - (\alpha_1 - \delta f) \) if \( y = 0 \). Since \( \alpha_1 > \delta f \) and so \( x < 0 \), we conclude that the real part of \( \lambda_A \) will be negative. Therefore, the proposed method is stable for the 3D wave equations discretized system.

4. Computational Results

Now, we consider three examples of the 3D wave Equation (1) to check the accuracy and efficiency of the proposed method.

**Example 1.** We consider Equation (1) for \( \delta = 0 \) and \( \alpha_i = 2, \ i = 1, 2 \) with the analytical solution \( u(x, y, z, t) = e^{-2t}\sinhx \sinhy \sinhz \).
The $\beta(x,y,z,t)$ is described appropriately. We choose $h = 0.1, 0.05$ and $\Delta t = 0.01$. Table 2 shows the comparison between the proposed method and the existing ones in terms of RMS error norms. It can be noted that the present solutions are more accurate than the solutions presented in [16] by EFG and MLPG methods, and Expo-MCBDQM [17]. Figure 2 illustrates the absolute error norms for fixed $z = 0.5$ with $h = 0.05$ at $t = 1$ while Figure 3 shows the behavior of the solutions. From Figures 2 and 3, one can notice that the absolute error norms are very small, and analytical and numerical solutions are very close each other which shows the accuracy of the proposed method.

**Table 2.** Comparison between the present method and existing methods with $h = 0.1$ and $\Delta t = 0.01$ at different values of $t$ for Example 1.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>$9.131 \times 10^{-7}$</td>
<td>$1.361376 \times 10^{-1}$</td>
<td>$6.389040 \times 10^{-4}$</td>
<td>$1.013 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.2</td>
<td>$9.126 \times 10^{-7}$</td>
<td>$1.108673 \times 10^{-1}$</td>
<td>$1.621007 \times 10^{-3}$</td>
<td>$1.666 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.3</td>
<td>$9.751 \times 10^{-7}$</td>
<td>$9.031794 \times 10^{-2}$</td>
<td>$2.069397 \times 10^{-3}$</td>
<td>$1.725 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.4</td>
<td>$9.357 \times 10^{-7}$</td>
<td>$7.555177 \times 10^{-2}$</td>
<td>$1.851491 \times 10^{-3}$</td>
<td>$1.498 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.5</td>
<td>$9.263 \times 10^{-7}$</td>
<td>$6.113317 \times 10^{-2}$</td>
<td>$1.406413 \times 10^{-3}$</td>
<td>$1.196 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.6</td>
<td>$8.105 \times 10^{-7}$</td>
<td>$5.076050 \times 10^{-2}$</td>
<td>$1.120239 \times 10^{-3}$</td>
<td>$9.059 \times 10^{-7}$</td>
</tr>
<tr>
<td>0.7</td>
<td>$6.102 \times 10^{-7}$</td>
<td>$4.276296 \times 10^{-2}$</td>
<td>$8.762877 \times 10^{-4}$</td>
<td>$7.061 \times 10^{-7}$</td>
</tr>
<tr>
<td>0.8</td>
<td>$4.458 \times 10^{-7}$</td>
<td>$3.416178 \times 10^{-2}$</td>
<td>$5.762842 \times 10^{-4}$</td>
<td>$5.566 \times 10^{-7}$</td>
</tr>
<tr>
<td>0.9</td>
<td>$3.614 \times 10^{-7}$</td>
<td>$3.072394 \times 10^{-2}$</td>
<td>$7.779958 \times 10^{-4}$</td>
<td>$4.758 \times 10^{-7}$</td>
</tr>
<tr>
<td>1.0</td>
<td>$3.326 \times 10^{-7}$</td>
<td>$2.562088 \times 10^{-2}$</td>
<td>$8.638229 \times 10^{-4}$</td>
<td>$4.417 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

**Figure 2.** The absolute error norms with $h = 0.05$ and $\Delta t = 0.01$ for $z = 0.5$ at $t = 1$.

**Figure 3.** The analytical (a) and numerical (b) solutions with $h = 0.05$ and $\Delta t = 0.01$ for $z = 0.5$ at $t = 1$. 
Example 2. Next, we consider Equation (1) for $a_1 = \delta = \kappa, a_2 = 0$ and $f(u) = u^2$ with the analytical solution $u(x,y,z,t) = \sin \pi x \sin \pi y \sin \pi z e^{-\kappa t}$.

The function $\hat{p}(x,y,z,t)$ is described appropriately and we choose the parameters $h = 0.05, 0.1, \Delta t = 0.01, \kappa = 3$ for the numerical approximation of this example. The proposed method is again compared with EFG [16] and MLPG [16] methods, and Expo-MCBDQM [17] in terms of RMS error norms and shown in Table 3. It is noticed that the proposed method shows better solutions than the solutions presented in [16,17]. Figure 4 illustrates the absolute error norms for $z = 1, h = 0.05$ at $t = 1$ while Figure 5 demonstrates the comparison of the analytical and numerical solutions, where a close agreement is noticed between analytical and numerical solutions. From Figure 4, one can notice that the absolute error norms are very small in $10^{-19}$, which shows that the proposed method provides very accurate results.

Example 3. Finally, we consider Equation (1) for $a_1 = a_2 = 0, \delta = -2$ and $f(u) = u$ with the analytical solution $u(x,y,z,t) = \sin \pi x \sin \pi y \sin \pi z \sin t$.

The function $\hat{p}(x,y,z,t)$ is chosen appropriately. We choose the parameters $h = 0.05$ and $0.1, \Delta t = 0.01$. The present solutions are compared with the solutions obtained by EFG [16] and MLPG [16] methods, and Expo-MCBDQM [17] in terms of RMS error norms and shown in Table 4. Again, it is noticed that the proposed method provides better solutions than the existing methods. Figure 6 illustrates the absolute error norms for $z = 0.5, h = 0.05$ at $t = 1$. Figure 7 shows the comparison of the analytical and numerical solutions.

Table 3. Comparison between the present method and existing methods with $h = 0.1$ and $\Delta t = 0.01$ at different values of $t$ for Example 2.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>$4.472 \times 10^{-6}$</td>
<td>$1.653265 \times 10^0$</td>
<td>$2.777931 \times 10^{-3}$</td>
<td>$5.667 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.2</td>
<td>$8.511 \times 10^{-6}$</td>
<td>$1.005632 \times 10^0$</td>
<td>$8.477482 \times 10^{-3}$</td>
<td>$9.701 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.3</td>
<td>$2.23 \times 10^{-6}$</td>
<td>$9.786343 \times 10^{-1}$</td>
<td>$1.352534 \times 10^{-2}$</td>
<td>$1.231 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.4</td>
<td>$5.02 \times 10^{-6}$</td>
<td>$7.456237 \times 10^{-1}$</td>
<td>$1.583307 \times 10^{-2}$</td>
<td>$1.512 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.5</td>
<td>$1.143 \times 10^{-5}$</td>
<td>$6.213675 \times 10^{-1}$</td>
<td>$1.550351 \times 10^{-2}$</td>
<td>$1.824 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.6</td>
<td>$1.062 \times 10^{-5}$</td>
<td>$4.354421 \times 10^{-1}$</td>
<td>$1.367202 \times 10^{-2}$</td>
<td>$2.222 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.7</td>
<td>$1.231 \times 10^{-5}$</td>
<td>$1.345213 \times 10^{-1}$</td>
<td>$1.052578 \times 10^{-2}$</td>
<td>$2.570 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.8</td>
<td>$1.324 \times 10^{-5}$</td>
<td>$9.973233 \times 10^{-2}$</td>
<td>$6.216680 \times 10^{-3}$</td>
<td>$2.866 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.9</td>
<td>$2.014 \times 10^{-5}$</td>
<td>$7.132423 \times 10^{-2}$</td>
<td>$5.280951 \times 10^{-3}$</td>
<td>$3.117 \times 10^{-5}$</td>
</tr>
<tr>
<td>1.0</td>
<td>$2.1025 \times 10^{-5}$</td>
<td>$6.124572 \times 10^{-2}$</td>
<td>$2.276681 \times 10^{-3}$</td>
<td>$3.329 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Figure 4. The absolute error norms with $h = 0.05$ and $\Delta t = 0.01$ for $z = 1$ at $t = 1$. 
Axioms 2022, 11, x FOR PEER REVIEW 9 of 13

Figure 5. The analytical (a) and numerical (b) solutions with $h = 0.05$ and $\Delta t = 0.01$ for $z = 1$ at $t = 1$.

Table 4. Comparison between the present method and existing methods with $h = 0.1$ and $\Delta t = 0.01$ at different values of $t$ for Example 3.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>$1.673 \times 10^{-7}$</td>
<td>$1.435666 \times 10^{-3}$</td>
<td>$8.903029 \times 10^{-5}$</td>
<td>$2.887 \times 10^{-7}$</td>
</tr>
<tr>
<td>0.2</td>
<td>$2.481 \times 10^{-7}$</td>
<td>$3.867576 \times 10^{-3}$</td>
<td>$9.910264 \times 10^{-5}$</td>
<td>$1.257 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.3</td>
<td>$1.522 \times 10^{-6}$</td>
<td>$5.033494 \times 10^{-3}$</td>
<td>$1.590388 \times 10^{-4}$</td>
<td>$2.944 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.4</td>
<td>$4.137 \times 10^{-6}$</td>
<td>$7.655177 \times 10^{-3}$</td>
<td>$3.77687 \times 10^{-4}$</td>
<td>$5.348 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.5</td>
<td>$7.465 \times 10^{-6}$</td>
<td>$9.119769 \times 10^{-3}$</td>
<td>$4.781290 \times 10^{-4}$</td>
<td>$8.787 \times 10^{-6}$</td>
</tr>
<tr>
<td>0.6</td>
<td>$3.304 \times 10^{-6}$</td>
<td>$1.034540 \times 10^{-2}$</td>
<td>$6.416380 \times 10^{-4}$</td>
<td>$1.361 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.7</td>
<td>$1.007 \times 10^{-5}$</td>
<td>$3.279875 \times 10^{-2}$</td>
<td>$8.809498 \times 10^{-4}$</td>
<td>$2.029 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.8</td>
<td>$1.716 \times 10^{-5}$</td>
<td>$5.233178 \times 10^{-2}$</td>
<td>$9.279331 \times 10^{-4}$</td>
<td>$2.918 \times 10^{-5}$</td>
</tr>
<tr>
<td>0.9</td>
<td>$3.014 \times 10^{-5}$</td>
<td>$6.072234 \times 10^{-2}$</td>
<td>$1.059260 \times 10^{-4}$</td>
<td>$4.049 \times 10^{-5}$</td>
</tr>
<tr>
<td>1.0</td>
<td>$4.221 \times 10^{-5}$</td>
<td>$7.545088 \times 10^{-2}$</td>
<td>$1.529316 \times 10^{-3}$</td>
<td>$5.432 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Figure 6. The absolute error norms with $h = 0.05$ and $\Delta t = 0.01$ for $z = 0.5$ at $t = 1$. 
subtractions, 3

\[ M + 1 \] simple arithmetic operations.

\[ M \] multiplications, and 2

Axioms 2022, 11, x FOR PEER REVIEW 11 of 13

Figure 7. The numerical (a) and analytical (b) solutions with \( h = 0.05 \) and \( \Delta t = 0.01 \) for \( z = 0.5 \) at 
\( t = 1 \).

Computational Complexity

The system of equations, where the coefficient matrix is tridiagonal, is solved by
using the Thomas algorithm which takes \( 3M \) subtractions, \( 3M \) multiplications, and \( 2M + 1 \)
divisions. Therefore, the algorithm needs a total of \( 8M + 1 \) simple arithmetic operations.
Therefore, the Thomas algorithm requires \( O(n) \) operations. The computational cost of the
SSPRK-(5,4) technique is same as the cost of traditional ODE solvers. Hence, the proposed
technique is not too complex from the computational point of view.

5. Conclusions

This work proposed a differential quadrature method based on cubic hyperbolic
B-spline functions together with SSPRK-(5,4) scheme to solve 3D wave equations. The
numerical examples show that the proposed method provides more accurate solutions
than those discussed in [16,17]. The matrix stability analysis is also investigated, and
we found that the proposed method is stable. Additionally, the method is economically
easy-to-implement for solving hyperbolic partial differential equations. Moreover, the
computational complexity shows that the proposed technique is not too complex from the
computational point of view.

Author Contributions: Formal analysis, M.T.; investigation, M.T.; methodology, M.T.; validation,
M.Z.M.; writing—original draft, M.T. and M.Z.M.; writing—review and editing, A.H.M. All authors
have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

Access 2019, 99, 1–11. [CrossRef]
2017, 80, 58–71. [CrossRef]
3. Baccouch, M.; Temimi, H. A high-order space-time ultra-weak discontinuous Galerkin method for the second-order wave


