Hindawi
Advances in Mathematical Physics
Volume 2017, Article ID 5691452, 15 pages
https://doi.org/10.1155/2017/5691452

Research Article

Bernoulli Collocation Method for Solving Linear Multidimensional Diffusion and Wave Equations with Dirichlet Boundary Conditions

Bashar Zogheib,[1] Emran Tohidi,[2] and Stanford Shateyi[3]

[1]*Department of Mathematics and Natural Sciences, American University of Kuwait, Salmiya, Kuwait*
[2]*Department of Mathematics, Kosar University of Bojnord, P.O. Box 9415615458, Bojnord, Iran*
[3]*Department of Mathematics and Applied Mathematics, University of Venda, Thohoyandou 0950, South Africa*

Correspondence should be addressed to Stanford Shateyi; stanford.shateyi@univen.ac.za

Received 5 November 2016; Accepted 30 January 2017; Published 22 February 2017

Academic Editor: Ricardo Weder

A numerical approach is proposed for solving multidimensional parabolic diffusion and hyperbolic wave equations subject to the appropriate initial and boundary conditions. The considered numerical solutions of the these equations are considered as linear combinations of the shifted Bernoulli polynomials with unknown coefficients. By collocating the main equations together with the initial and boundary conditions at some special points (i.e., CGL collocation points), equations will be transformed into the associated systems of linear algebraic equations which can be solved by robust Krylov subspace iterative methods such as GMRES. Operational matrices of differentiation are implemented for speeding up the operations. In both of the one-dimensional and two-dimensional diffusion and wave equations, the geometrical distributions of the collocation points are depicted for clarity of presentation. Several numerical examples are provided to show the efficiency and spectral (exponential) accuracy of the proposed method.

1. Introduction

Many of the physical models and chemical reactions can be formulated in terms of parabolic and hyperbolic partial differential equations (PDEs) [1, 2]. Since analyzing these models and reactions has considerable importance in applied science and engineering, we should investigate the solutions of the associated governed PDEs by modern tools [3, 4]. Analytical approaches for solving PDEs have received the researchers attention for solving a large number of PDEs, but numerical methods are more favorable with respect to these approaches. One of the basic advantages of the numerical methods, which made them more popular with respect to the analytical approaches, is based on the implementation of the operational matrices of differentiation and high accurate Gauss quadrature rules instead of direct differentiation and integration. By using these operational matrices and quadratures, computations time in numerical algorithms may be decreased significantly. Therefore, in most of the numerical methods, one of the above-mentioned tools may be applied for speeding up the operations in solving PDEs [5].

In a numerical point of view, the methods that are based on the operational matrices of differentiation may be divided into the collocation and Tau matrix methods. In Tau matrix methods, all the known and unknown functions should be approximated in terms of a specific complete basis (such as orthogonal polynomials or trigonometric functions). Since the mentioned basis is complete, one can factorize this basis and simplify the considered PDEs into the associated system of algebraic equations. On the other hand, in every considered PDE one may have some boundary conditions and we should impose these conditions in the procedure of the Tau matrix method. Our numerical implementation experiences show that if the boundary conditions are first imposed to the main equations [5] (via transforming the basic PDEs into the associated integro-PDEs), then we may have

more stable numerical solutions. However, in some research works (e.g., Taylor matrix method [6, 7], Euler matrix method [8], and Chebyshev Tau method [9]) boundary conditions are imposed after discretizing the main equations. Such ideas may have similar numerical results with [5] but produce more algebraic equations and this may take more computations time and therefore make these methods deficient. Another disadvantage of Tau matrix method is that these methods apply the operational matrices of product for solving PDEs with variable coefficients [10]. Such idea again reduces the accuracy of the Tau matrix method. Moreover, Tau matrix schemes in general have difficulties in the treatment of nonlinear multidimensional PDEs, since no straightforward algebraic system of equations can be efficiently achieved for such problems [11]. However, the collocation schemes for PDEs are rarely ready to implement and they can overcome the aforementioned challenges.

In recent years, some research works have been focused on the application of collocation methods for solving linear one-dimensional parabolic and hyperbolic (specially Telegraph equations) PDEs such as Chebyshev collocation method [12] and Bessel collocation method [13, 14]. But application of this scheme for solving multidimensional PDEs has had few results. This partially motivates us to propose a new collocation scheme, which is based on the Bernoulli polynomials, for solving linear multidimensional diffusion and wave PDEs. It should be noted that the proposed collocation method can be generalized for solving other linear multidimensional parabolic and hyperbolic PDEs. In this paper we consider the following linear one-dimensional diffusion equation

$$\frac{\partial \psi}{\partial \tau} = \frac{\partial^2 \psi}{\partial \eta^2} + \omega(\eta, \tau), \quad (\eta, \tau) \in [a_\eta, b_\eta] \times [a_\tau, b_\tau], \tag{1}$$

with the initial condition

$$\psi(\eta, a_\tau) = f(\eta), \quad \eta \in [a_\eta, b_\eta], \tag{2}$$

together with the Dirichlet boundary conditions

$$\begin{aligned} &\psi(a_\eta, \tau) = g_1(\tau), \\ &\psi(b_\eta, \tau) = g_2(\tau), \\ &\qquad \tau \in [a_\tau, b_\tau]. \end{aligned} \tag{3}$$

Also, we will consider the following linear one-dimensional wave equation

$$\frac{\partial^2 \psi}{\partial \tau^2} = \frac{\partial^2 \psi}{\partial \eta^2} + \omega(\eta, \tau), \quad (\eta, \tau) \in [a_\eta, b_\eta] \times [a_\tau, b_\tau], \tag{4}$$

with the initial conditions

$$\begin{aligned} &\psi(\eta, a_\tau) = f(\eta), \\ &\frac{\partial \psi}{\partial \tau}(\eta, a_\tau) = k(\eta), \\ &\qquad \eta \in [a_\eta, b_\eta], \end{aligned} \tag{5}$$

together with the Dirichlet boundary conditions

$$\begin{aligned} &\psi(a_\eta, \tau) = g_1(\tau), \\ &\psi(b_\eta, \tau) = g_2(\tau), \\ &\qquad \tau \in [a_\tau, b_\tau]. \end{aligned} \tag{6}$$

It should be noted that all the known functions in above-mentioned PDEs such as $f(\eta)$, $k(\eta)$, $g_1(\tau)$, $g_2(\tau)$, and $\omega(\eta, \tau)$ are some smooth functions. Numerical solutions of the above-mentioned equations are considered in the literature through some classical and modern techniques such as finite difference methods (FDMs) [15–18], finite element methods (FEMs) [19], finite volume methods (FVMs) [20], dual reciprocity boundary integral equation method [21], interpolating scaling function method [22], and Haar wavelet scheme [23]. In this paper, we treat the above equations (together with their 2D generalization forms) by a numerical method which is based on Bernoulli polynomials as the basis and the well-known Chebyshev Gauss Lobatto (CGL) collocation points. In this regard, the considered equations are collocated and then transformed into the associated systems of linear algebraic equations which can be solved through some iterative methods such as GMRES.

The structure of the remainder of this paper is as follows. In the next section, we will review some preliminaries regarding the shifted Bernoulli polynomials and shifted operational matrices which play important roles throughout the paper. In Section 3, implementation of the Bernoulli collocation method (BCM) for solving both of the 1D equations (1)–(3) and (4)–(6) will be provided and in Section 4, generalization of BCM for solving 2D diffusion and wave equations will be implemented. In Section 5, several numerical examples are given to illustrate the spectral accuracy of the proposed method. Finally, some conclusions regarding the suggested BCM are conveyed in Section 6.

2. Preliminaries

Bernoulli polynomials form a complete basis in the interval $[0, 1]$. Since in our considered PDEs we have arbitrary intervals, we should introduce the shifted Bernoulli polynomials, which keep the property of completeness, and the shifted operational matrices of differentiation. The shifted Bernoulli polynomials can be constructed via the relation $\beta_i(\eta) = B_i((\eta - a_\eta)/(b_\eta - a_\eta))$ for any arbitrary positive integer index i in the interval $[a_\eta, b_\eta]$. It should be recalled that the standard Bernoulli polynomials, in the interval $[0, 1]$, have the following properties [24]:

$$\begin{aligned} &B_i'(x) = iB_{i-1}(x), \quad \forall i \geq 1, \\ &\int_0^1 B_i(x)\,dx = 0, \quad \forall i \geq 1, \\ &B_0(x) = 1. \end{aligned} \tag{7}$$

Therefore the shifted Bernoulli polynomials have the following properties in the interval $[a_\eta, b_\eta]$:

$$\begin{gathered}\beta_i'(\eta) = \frac{i}{b_\eta - a_\eta}\beta_{i-1}(\eta), \quad \forall i \geq 1,\\ \int_{a_\eta}^{b_\eta} \beta_i(\eta)\, d\eta = 0, \quad \forall i \geq 1,\\ \beta_0(\eta) = 1.\end{gathered} \tag{8}$$

Now we assume that $\beta(\eta) = [\beta_0(\eta)\ \beta_1(\eta)\ \cdots\ \beta_N(\eta)]_{1\times(N+1)}$. The shifted operational matrix of differentiation associated with the shifted Bernoulli polynomials can be derived from the first relation in (8):

$$\begin{aligned}&\begin{bmatrix}\beta_0(\eta)\\ \beta_1(\eta)\\ \vdots\\ \beta_{N-1}(\eta)\\ \beta_N(\eta)\end{bmatrix}'\\ &= \overbrace{\frac{1}{b_\eta - a_\eta}\begin{bmatrix}0 & 0 & 0 & \cdots & 0\\ 1 & 0 & 0 & \cdots & 0\\ 0 & 2 & 0 & \cdots & 0\\ \vdots & \vdots & \ddots & \ddots & \vdots\\ 0 & 0 & \cdots & N & 0\end{bmatrix}}^{M_\eta}\begin{bmatrix}\beta_0(\eta)\\ \beta_1(\eta)\\ \vdots\\ \beta_{N-1}(\eta)\\ \beta_N(\eta)\end{bmatrix},\end{aligned} \tag{9}$$

where M_η is the shifted operational matrix of differentiation in the interval $[a_\eta, b_\eta]$. Because of the existence of two-variable functions in (1)–(3) and (4)–(6), we must extend the mentioned matrix through the Kronecker multiplication. For this purpose, assume that

$$\beta(\eta,\tau) = [\beta_0(\eta,\tau)\ \ \beta_1(\eta,\tau)\ \ \cdots\ \ \beta_N(\eta,\tau)]_{1\times(N+1)^2}, \tag{10}$$

where $\beta_i(\eta,\tau) = [\beta_0(\eta,\tau)\ \beta_1(\eta,\tau)\ \cdots\ \beta_N(\eta,\tau)]$ for all $i = 0, 1, \ldots, N$ and $\beta_{m,n}(\eta,\tau) = \beta_m(\eta)\beta_n(\tau)$ for all $m, n = 0, 1, \ldots, N$. Evidently, $\beta(\eta,\tau) = \beta(\eta)\otimes\beta(\tau)$, where $\beta(\tau) = [\beta_0(\tau)\ \beta_1(\tau)\ \cdots\ \beta_N(\tau)]_{1\times(N+1)}$.

In the following lemma, operational matrices of differentiation associated with the two-variable functions will be introduced.

Lemma 1. *Suppose that $\widehat{M} = I_{N+1}\otimes M_\tau^T$ and $\widetilde{M} = M_\eta^T\otimes I_{N+1}$, where I_{N+1} denotes the identity matrix of dimension $(N+1)$ and $M_\tau = ((b_\eta - a_\eta)/(b_\tau - a_\tau))M_\eta$. One can conclude that*

(i)

$$\begin{gathered}\frac{\partial\beta(\eta,\tau)}{\partial\tau} = \beta_\tau(\eta,\tau) = \beta(\eta,\tau)\widehat{M},\\ \frac{\partial^2\beta(\eta,\tau)}{\partial\tau^2} = \beta_{\tau\tau}(\eta,\tau) = \beta(\eta,\tau)\left(\widehat{M}\right)^2,\end{gathered} \tag{11}$$

(ii)

$$\begin{gathered}\frac{\partial\beta(\eta,\tau)}{\partial\eta} = \beta_\eta(\eta,\tau) = \beta(\eta,\tau)\widetilde{M},\\ \frac{\partial^2\beta(\eta,\tau)}{\partial\eta^2} = \beta_{\eta\eta}(\eta,\tau) = \beta(\eta,\tau)\left(\widetilde{M}\right)^2.\end{gathered} \tag{12}$$

Proof. See [24]. □

Since in 2D diffusion and wave equations, which will be considered in Section 4, we have three-variable functions, again one can develop the subject of operational matrices via the Tensor product. For this goal, we should suppose that

$$\beta(\eta,\xi,\tau) = \left(\beta(\eta)\otimes\beta(\xi)\otimes\beta(\tau)\right)_{1\times(N+1)^3}, \tag{13}$$

where $\beta(\xi) = [\beta_0(\xi)\ \beta_1(\xi)\ \cdots\ \beta_N(\xi)]_{1\times(N+1)}$. Similar to the previous lemma, operational matrices of differentiation corresponding to the three-variable functions will be introduced.

Lemma 2. *Assume that $\widehat{\widehat{M}} = I_{(N+1)^2}\otimes M_\tau^T$, $\widetilde{\widetilde{M}} = M_\eta^T\otimes I_{(N+1)^2}$, and $\dot{\dot{M}} = I_{N+1}\otimes M_\xi^T\otimes I_{N+1}$, where $M_\xi = ((b_\eta - a_\eta)/(b_\xi - a_\xi))M_\eta$. The following relations hold.*

(i)

$$\begin{gathered}\frac{\partial\beta(\eta,\xi,\tau)}{\partial\tau} = \beta_\tau(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\widehat{\widehat{M}},\\ \frac{\partial^2\beta(\eta,\xi,\tau)}{\partial\tau^2} = \beta_{\tau\tau}(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\left(\widehat{\widehat{M}}\right)^2.\end{gathered} \tag{14}$$

(ii)

$$\begin{gathered}\frac{\partial\beta(\eta,\xi,\tau)}{\partial\eta} = \beta_\eta(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\widetilde{\widetilde{M}},\\ \frac{\partial^2\beta(\eta,\xi,\tau)}{\partial\eta^2} = \beta_{\eta\eta}(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\left(\widetilde{\widetilde{M}}\right)^2.\end{gathered} \tag{15}$$

(iii)

$$\begin{gathered}\frac{\partial\beta(\eta,\xi,\tau)}{\partial\xi} = \beta_\xi(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\dot{\dot{M}},\\ \frac{\partial^2\beta(\eta,\xi,\tau)}{\partial\xi^2} = \beta_{\xi\xi}(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\left(\dot{\dot{M}}\right)^2.\end{gathered} \tag{16}$$

Proof. The proof of this lemma is similar to the previous lemma. □

3. Bernoulli Collocation Method for 1D Equations

This section is devoted to implement BCM for solving 1D diffusion equations (1)–(3) and also 1D wave equations (4)–(6). In this regard, we first discretize 1D diffusion equation

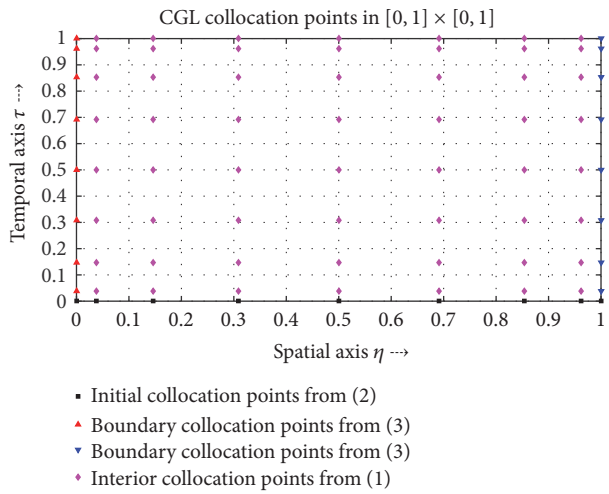

Figure 1: CGL collocation points in the unit square for $N = 8$ for 1D diffusion equation.

and then localize 1D wave equation in two separate subsections via a geometrical point of view. In both cases, we should assemble the associated algebraic equations to form a final linear system $AU = b$ which can be solved by some efficient iterative solvers such as GMRES method. It should be noted that BCM has received considerable attentions during recent years. The interested readers can refer to [25] and the references therein for application of BCM in solving complex ODEs, nonlinear two-point BVPs, delay Pantograph ODEs, and other types of applied mathematics problems.

3.1. 1D Diffusion Equations. In this subsection, the aim is to reduce (1)–(3) into the associated system of linear algebraic equations in the form of $AU = b$ by implementing the BCM. This process has three steps.

(1) Collocating the main equation (1).

(2) Collocating the boundary conditions (3).

(3) Collocating the initial condition (2).

At first we assume that

$$\begin{aligned}\psi(\eta,\tau) \approx \psi_N(\eta,\tau) &= \sum_{m=0}^{N}\sum_{n=0}^{N} u_{mn}\beta_m(\eta)\beta_n(\tau) \\ &= \beta(\eta,\tau)U,\end{aligned} \tag{17}$$

where $\beta(\eta,\tau)$ is defined in the previous section and

$$U = [u_{00}\ \ u_{01}\ \ \cdots\ \ u_{0N}\ \ u_{10}\ \ u_{11}\ \ \cdots\ \ u_{1N}\ \ \cdots\ \ u_{N0}\ \ u_{N1}\ \ \cdots\ \ u_{NN}]^T_{(N+1)^2\times 1}. \tag{18}$$

It should be noted that our aim is to find the unknown vector U. From the previous section we have

$$\begin{aligned}\psi_\tau(\eta,\tau) &\approx \psi_{N,\tau}(\eta,\tau) = \beta(\eta,\tau)\widehat{M}U, \\ \psi_{\eta\eta}(\eta,\tau) &\approx \psi_{N,\eta\eta}(\eta,\tau) = \beta(\eta,\tau)\left(\widetilde{M}\right)^2 U.\end{aligned} \tag{19}$$

Also, we should define some suitable collocation points. In this paper, we choose the CGL points in the following form:

$$\begin{aligned}\eta_i &= \frac{b_\eta - a_\eta}{2}\left(1-\cos\left(\frac{(i-1)\pi}{N}\right)\right) + a_\eta, \\ &\qquad i = 1,2,\ldots,N+1, \\ \tau_j &= \frac{b_\tau - a_\tau}{2}\left(1-\cos\left(\frac{(j-1)\pi}{N}\right)\right) + a_\tau, \\ &\qquad j = 1,2,\ldots,N+1.\end{aligned} \tag{20}$$

Figure 1 plots such collocation points for $N = 8$ in the unit square (i.e., $[a_\eta, b_\eta]\times[a_\tau, b_\tau] = [0,1]\times[0,1]$) for the diffusion equation. At the first step, we collocate the main equation via the interior collocation points.

Step 1 (interior collocation points: main equation (1)). In this part of the paper, we have $2 \le i \le N$ and $2 \le j \le N+1$ (see the purple rhombics in Figure 1). So we should define A_{main} with $N(N-1)$ rows and $(N+1)^2$ columns. Moreover, we should define a column vector b_{main} with $N(N-1)$ elements. Therefore, one can refer to the first step of Algorithm 1 for this purpose.

Step 2 (boundary conditions collocation (3)). In this part of the paper, we have $i = 1$ for the left boundary condition and $i = N+1$ for the right boundary condition. In both cases, we have $2 \le j \le N+1$. Therefore, we should define A_{bcl} (left boundary condition: see the red triangles in Figure 1) and A_{bcr} (right boundary condition: see the blue down triangles in Figure 1) with N rows and $(N+1)^2$ columns. Moreover, we should define column vectors b_{bcl} (left boundary condition) and b_{bcr} (right boundary condition) with N elements. Thus, one can refer to the second step of Algorithm 1 for this goal.

Step 3 (initial condition collocation (2)). In this part of the paper, we have $j = 1$ and also $1 \le i \le N+1$ (see the black squares in Figure 1). In this case we should define A_{in} matrix with $(N+1)$ rows and $(N+1)^2$ columns. Moreover, we should define b_{in} column vector with $(N+1)$ elements. Therefore, one can refer to the third step of Algorithm 1 for this aim.

Step 4 (assembling the matrices and column vectors). Finally, by assembling the coefficient matrices and right hand side column vectors, we will reach to the system of linear algebraic equations in the form of $AU = b$, which can be solved by some appropriate iterative methods such as GMRES algorithm.

```
Step 1: Collocating the main equation (1):
for i = 2 : N
for j = 2 : N + 1
A_main((i − 2) ∗ N + (j − 1), :) = β(η_i, τ_j) ∗ (M̃ − (M̃)^2);
b_main((i − 2) ∗ N + (j − 1), 1) = ω(η_i, τ_j);
end
end
Step 2: Collocating the boundary conditions (3):
for j = 2 : N + 1
A_bcl(j − 1, :) = β(a_η, τ_j);
A_bcr(j − 1, :) = β(b_η, τ_j);
b_bcl(j − 1, 1) = g_1(τ_j);
b_bcr(j − 1, 1) = g_2(τ_j);
end
Step 3: Collocating the initial condition (2):
for i = 1 : N + 1
A_in(i, :) = β(η_i, a_τ);
b_in (i, 1) = f (η_i)
end
Step 4: Assembling the coefficient matrices and right hand side column vectors:
A = [A_main; A_bcl; A_bcr; A_in];
b = [b_main; b_bcl; b_bcr; b_in];
maxit = size(A, 1);
tol = 10^−(N−2);
U = gmres (A, b, [], tol, maxit)
```

Algorithm 1: Implementation of BCM for solving (1)–(3).

Algorithm 1 can describe BCM for solving 1D diffusion equations (1)–(3).

3.2. 1D Wave Equations. This subsection is similar to the previous subsection and we have just an extra initial condition $(\partial\psi/\partial\tau)(\eta, a_\tau) = k(\eta)$. Also, the aim is to reduce (4)–(6) into the associated system of linear equations in the form of $AU = b$ by implementing the BCM. Similar to the previous subsection, this process has three steps.

(1) Collocating the main equation (4).

(2) Collocating the boundary conditions (6).

(3) Collocating the initial conditions (5).

Again, we assume a similar approximate solution in the form of (17) and consider the collocation points in the rectangle $[a_\eta, b_\eta] \times [a_\tau, b_\tau]$ that was previously introduced in (20). Similar to the previous subsection, one can write

$$\psi_{\tau\tau}(\eta, \tau) \approx \psi_{N,\tau\tau}(\eta, \tau) = \beta(\eta, \tau)\left(\widehat{M}\right)^2 U. \tag{21}$$

It should be noted that Figure 2 plots such collocation points for $N = 8$ in the unit square for the wave equation. The readers can see the difference between this figure and the previous figure in distribution of collocation points. At the first step, we collocate the main equation via the interior collocation points.

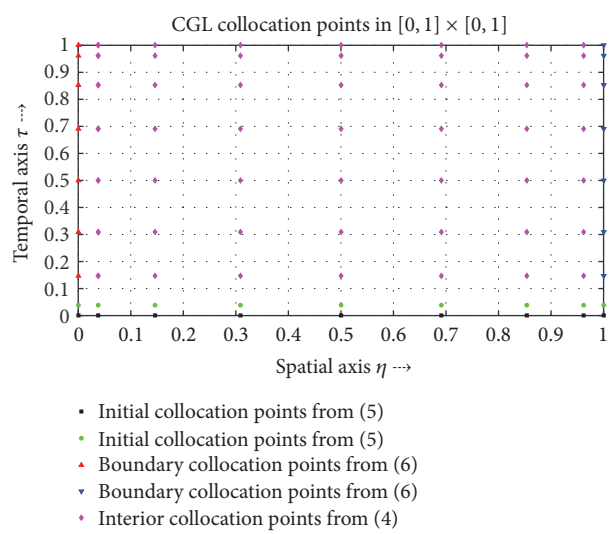

Figure 2: CGL collocation points in the unit square for $N = 8$ for 1D wave equation.

Step 1 (interior collocation points: main equation (4)). In this part of the paper, we have $2 \le i \le N$ and $3 \le j \le N + 1$ (see the purple rhombics in Figure 2). So we should define A_{main} with $(N-1)^2$ rows and $(N+1)^2$ columns. Moreover, we should define a column vector b_{main} with $(N-1)^2$ elements. Therefore, one can refer to the first step of Algorithm 2 for this purpose.

Step 2 (boundary conditions collocation (6)). In this stage, we have $i = 1$ for the left boundary condition and $i = N + 1$

```
Step 1: Collocating the main equation (4):
for i = 2 : N
for j = 3 : N + 1
A_main((i − 2) ∗ N + (j − 2), :) = β(η_i, τ_j) ∗ ((M̃)^2 − (M̃)^2);
b_main((i − 2) ∗ N + (j − 2), 1) = ω(η_i, τ_j);
end
end
Step 2: Collocating the boundary conditions (6):
for j = 3 : N + 1
A_bcl(j − 2, :) = β(a_η, τ_j);
A_bcr(j − 2, :) = β(b_η, τ_j);
b_bcl(j − 2, 1) = g_1(τ_j);
b_bcr(j − 2, 1) = g_2(τ_j);
End
Step 3: Collocating the initial conditions (5):
for i = 1 : N + 1
A_in(i, :) = β(η_i, a_τ);
A_in1(i, :) = β(η_i, a_τ)M̃;
b_in(i, 1) = f(η_i);
b_in1(i, 1) = k(η_i);
end
Step 4: Assembling the coefficient matrices and right hand side column vectors:
A = [A_main; A_bcl; A_bcr; A_in; A_in1];
b = [b_main; b_bcl; b_bcr; b_in; b_in1];
maxit = size(A, 1);
tol = 10^−(N−2);
U = gmres(A, b, [], tol, maxit)
```

Algorithm 2: Implementation of BCM for solving (4)–(6).

for the right boundary condition. In both cases, we have $3 \leq j \leq N+1$. Therefore, we should define A_{bcl} (left boundary condition: see the red triangles in Figure 2) and A_{bcr} (right boundary condition: see the blue down triangles in Figure 2) with $(N-1)$ rows and $(N+1)^2$ columns. Moreover, we should define column vectors b_{bcl} (left boundary condition) and b_{bcr} (right boundary condition) with $(N-1)$ elements. Therefore, one can refer to the second step of Algorithm 2 for this goal.

Step 3 (initial conditions collocation (5)). For the initial condition $\psi(\eta, a_\tau) = f(\eta)$, we have $j = 1$ and also $1 \leq i \leq N+1$ (see the black squares in Figure 2) and for the initial condition $(\partial\psi/\partial\tau)(\eta, a_\tau) = k(\eta)$ we have $j = 2$ and also $1 \leq i \leq N+1$ (see the green circles in Figure 2). In these cases, we should define A_{in} and A_{in1} matrices with $(N+1)$ rows and $(N+1)^2$ columns. Moreover, we should define b_{in} and b_{in1} column vectors with $(N+1)$ elements. Therefore, one can refer to the third step of Algorithm 2 for this aim.

In this case, assembling the coefficient matrices and right hand side column vectors is similar to the previous subsection. Algorithm 2 can describe BCM for solving 1D wave equations (4)–(6).

4. Bernoulli Collocation Method for 2D Equations

In this section, we generalize the BCM for solving linear 2D diffusion and wave equations. Similar to the previous section, we depict the collocation points associated with the initial conditions and boundary conditions in each case of the diffusion and wave equations for clarity of presentation. Therefore, we consider the following linear 2D diffusion equation:

$$\frac{\partial\psi}{\partial\tau} = \left(\frac{\partial^2\psi}{\partial\eta^2} + \frac{\partial^2\psi}{\partial\xi^2}\right) + \omega(\eta, \xi, \tau), \quad (\eta, \xi, \tau) \in [a_\eta, b_\eta] \times [a_\xi, b_\xi] \times [a_\tau, b_\tau], \tag{22}$$

with the initial condition

$$\psi(\eta, \xi, a_\tau) = f(\eta, \xi), \quad (\eta, \xi) \in [a_\eta, b_\eta] \times [a_\xi, b_\xi], \tag{23}$$

together with the Dirichlet boundary conditions

$$\begin{aligned} &\psi(a_\eta, \xi, \tau) = g_1(\xi, \tau), \\ &\psi(b_\eta, \xi, \tau) = g_2(\xi, \tau), \\ &\qquad (\xi, \tau) \in [a_\xi, b_\xi] \times [a_\tau, b_\tau], \\ &\psi(\eta, a_\xi, \tau) = h_1(\eta, \tau), \\ &\psi(\eta, b_\xi, \tau) = h_2(\eta, \tau), \\ &\qquad (\eta, \tau) \in [a_\eta, b_\eta] \times [a_\tau, b_\tau]. \end{aligned} \tag{24}$$

Also, we will consider the following linear 2D wave equation:

$$\frac{\partial^2\psi}{\partial\tau^2} = \left(\frac{\partial^2\psi}{\partial\eta^2} + \frac{\partial^2\psi}{\partial\xi^2}\right) + \omega(\eta,\xi,\tau), \quad (\eta,\xi,\tau) \in [a_\eta, b_\eta] \times [a_\xi, b_\xi] \times [a_\tau, b_\tau], \tag{25}$$

with the initial condition

$$\psi(\eta,\xi,a_\tau) = f(\eta,\xi), \quad \frac{\partial\psi(\eta,\xi,a_\tau)}{\partial\tau} = k(\eta,\xi), \quad (\eta,\xi) \in [a_\eta, b_\eta] \times [a_\xi, b_\xi], \tag{26}$$

together with the Dirichlet boundary conditions

$$\psi(a_\eta,\xi,\tau) = g_1(\xi,\tau), \quad \psi(b_\eta,\xi,\tau) = g_2(\xi,\tau), \quad (\xi,\tau) \in [a_\xi, b_\xi] \times [a_\tau, b_\tau],$$

$$\psi(\eta,a_\xi,\tau) = h_1(\eta,\tau), \quad \psi(\eta,b_\xi,\tau) = h_2(\eta,\tau), \quad (\eta,\tau) \in [a_\eta, b_\eta] \times [a_\tau, b_\tau]. \tag{27}$$

In the next subsection, we will extend BCM for solving 2D diffusion equations (22)–(24).

4.1. BCM for Solving 2D Diffusion Equations. We suppose that the approximate solution of (22)–(24) can be written in the following form of

$$\begin{aligned}\psi(\eta,\xi,\tau) &\approx \psi_N(\eta,\xi,\tau) \\ &= \sum_{m=0}^{N}\sum_{n=0}^{N}\sum_{p=0}^{N} u_{mnp}\beta_m(\eta)\beta_n(\xi)\beta_p(\tau) \\ &= \beta(\eta,\xi,\tau)U,\end{aligned} \tag{28}$$

where $\beta(\eta,\xi,\tau) = \beta(\eta) \otimes \beta(\xi) \otimes \beta(\tau)$ and $U = [u_{0..}^T \ u_{1..}^T \ \cdots \ u_{N..}^T]^T$ in which

$$u_{i..} = [u_{i00} \ u_{i01} \ \cdots \ u_{i0N} \ u_{i10} \ u_{i11} \ \cdots \ u_{i1N} \ \cdots \ u_{iN0} \ u_{iN1} \ \cdots \ u_{iNN}]. \tag{29}$$

According to discussions in the second section, one can deduce that

$$\begin{aligned}\psi_\tau(\eta,\xi,\tau) &\approx \psi_{N,\tau}(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\widehat{\widetilde{M}}U, \\ \psi_{\eta\eta}(\eta,\xi,\tau) &\approx \psi_{N,\eta\eta}(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\left(\widetilde{\widetilde{M}}\right)^2 U, \\ \psi_{\xi\xi}(\eta,\xi,\tau) &\approx \psi_{N,\xi\xi}(\eta,\xi,\tau) = \beta(\eta,\xi,\tau)\left(\dot{M}\right)^2 U.\end{aligned} \tag{30}$$

Similar to the 1D cases, one can consider the following CGL collocation points in 2D equations in the rectangle cube $[a_\eta, b_\eta] \times [a_\xi, b_\xi] \times [a_\tau, b_\tau]$:

$$\begin{aligned}\eta_i &= \frac{b_\eta - a_\eta}{2}\left(1 - \cos\left(\frac{(i-1)\pi}{N}\right)\right) + a_\eta, \quad i = 1,2,\ldots,N, \\ \xi_j &= \frac{b_\xi - a_\xi}{2}\left(1 - \cos\left(\frac{(j-1)\pi}{N}\right)\right) + a_\xi, \quad j = 1,2,\ldots,N, \\ \tau_k &= \frac{b_\tau - a_\tau}{2}\left(1 - \cos\left(\frac{(k-1)\pi}{N}\right)\right) + a_\tau, \quad k = 1,2,\ldots,N.\end{aligned} \tag{31}$$

Figure 3 shows only the collocation points associated with the initial condition (23) and also the boundary conditions (24) for 2D diffusion equation for $N = 6$. It should be noted that depicting the collocation points associated with the main equation (22) may make the readers confused. Moreover, in this figure x stands for the η variable, y stands for the ξ variable, and z stands for the τ variable.

BCM for solving (22)–(24) has also three steps: main equation collocation; initial condition collocation; and boundary conditions collocation. Despite the 1D cases, we first start with the initial conditions.

Step 1 (initial condition collocation (23)). For the initial condition $\psi(\eta,\xi,a_\tau) = f(\eta,\xi)$ we have $k = 1$ and also $1 \le i \le N+1$, $1 \le j \le N+1$ (see the black squares in Figure 3). In this case we should define A_{in} matrix with $(N+1)^2$ rows and $(N+1)^3$ columns. Moreover, we should define b_{in} column vector with $(N+1)^2$ elements. Therefore, one can refer to the first step of Algorithm 3 for this aim.

Step 2 (boundary conditions collocation (24)). Since we have four boundary conditions, we should discretize them in two different ways. For the x variable, we have $i = 1$ for the left boundary condition and $i = N+1$ for the right boundary condition. In both cases, we have $1 \le j \le N+1$ and $2 \le k \le N+1$. Therefore, we should define A_{bcl} (left boundary condition: see the red circles in Figure 3) and A_{bcr} (right boundary condition: see the green circles in Figure 3)

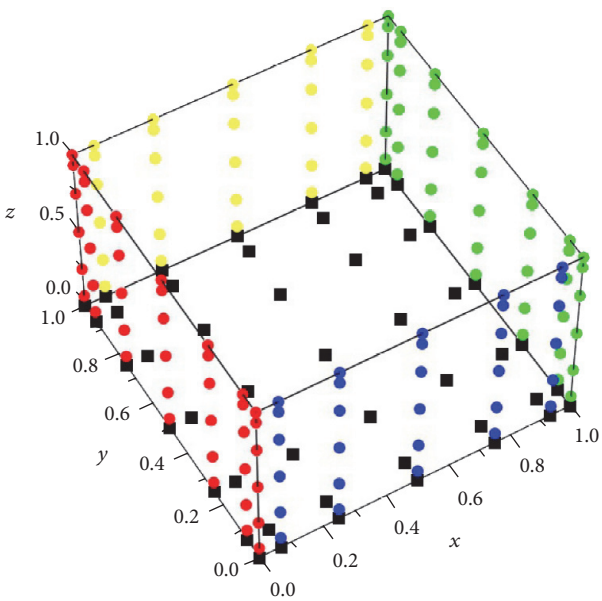

Figure 3: CGL collocation points in the unit cube for $N = 6$ for 2D diffusion equation.

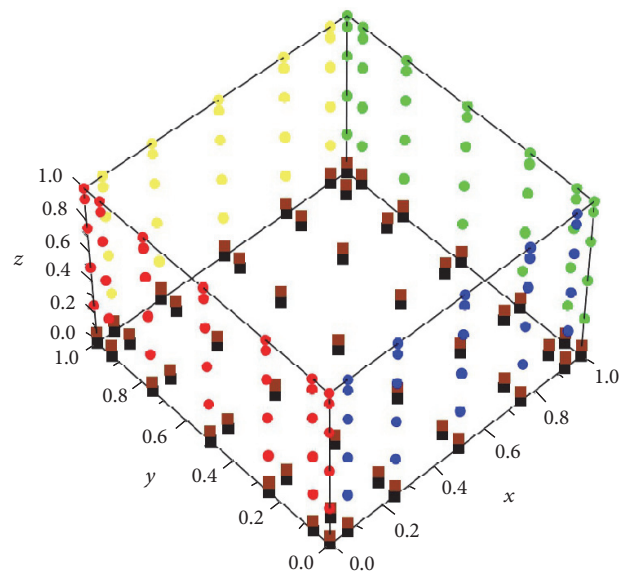

Figure 4: CGL collocation points in the unit cube for $N = 6$ for 2D wave equation.

with $N(N+1)$ rows and $(N+1)^3$ columns. Moreover, we should define column vectors b_{bcl} (left boundary condition) and b_{bcr} (right boundary condition) with $N(N+1)$ elements. For the y variable, we have $j = 1$ for the bottom boundary condition and $j = N+1$ for the top boundary condition. In both cases, we have $2 \le i \le N$ and $2 \le k \le N+1$. Therefore, we should define A_{bcb} (bottom boundary condition: see the blue circles in Figure 3) and A_{bct} (top boundary condition: see the yellow circles in Figure 3) with $N(N-1)$ rows and $(N+1)^3$ columns. Moreover, we should define column vectors b_{bcb} (bottom boundary condition) and b_{bct} (top boundary condition) with $N(N-1)$ elements. Therefore, one can refer to the second step of Algorithm 3 for this goal.

Step 3 (interior collocation points: main equation (22)). In this part of the paper, we have $2 \le i \le N$, $2 \le j \le N$, and $2 \le k \le N+1$. So we should define A_{main} with $N(N-1)^2$ rows and $(N+1)^3$ columns. Moreover, we should define a column vector b_{main} with $N(N-1)^2$ elements. Therefore, one can refer to the third step of Algorithm 3 for this purpose.

In this case, assembling the coefficient matrices and right hand side column vectors is similar to the previous subsection. Algorithm 3 can describe BCM for solving 2D diffusion equations (22)–(24).

4.2. BCM for Solving 2D Wave Equations. Similar to the previous subsection, we should consider an approximate solution in the form of (28) and also the collocation points in the rectangle cube $[a_\eta, b_\eta] \times [a_\xi, b_\xi] \times [a_\tau, b_\tau]$ that is introduced in (31). Moreover, one can write

$$\psi_{\tau\tau}(\eta, \xi, \tau) \approx \psi_{N,\tau\tau}(\eta, \xi, \tau) = \beta(\eta, \xi, \tau)\left(\widehat{\widehat{M}}\right)^2 U. \quad (32)$$

Distribution of collocation points in 2D wave equations is similar to 2D diffusion equations and just has a difference (see the brown squares in Figure 4). Such brown squares are related to the initial condition $\partial\psi(\eta, \xi, a_\tau)/\partial\tau = k(\eta, \xi)$.

For clarity of presentation, we just provide Algorithm 4 for describing BCM for solving (25)–(27). In this algorithm, we first discretize initial conditions (26) and then localize boundary conditions (27) and finally collocate the main equation (25) via the shifted CGL collocation points.

5. Numerical Examples

In this section, we will test our method for solving both of the considered diffusion and wave equations. In the first example, we consider a constructed 1D diffusion equation and in the second example, we provide a 1D wave which is selected from [6]. Moreover, in the third and fourth examples, we will provide two test problems that are selected from [5] in the case of 2D diffusion equations. The presented idea (i.e., BCM) is easy to implement and can be applied for solving other 2D parabolic and hyperbolic equations such as 2D Telegraph equations [26]. All of the programs associated with the implementation of BCM for solving the considered examples are written in Matlab 2015b in a Laptop PC with 12 GB Ram with Cash of 6. Readers of this paper can communicate for receiving the associated codes for each example via email of the corresponding author. It should be noted that the tolerance for the GMRES solver is set to be $10^{-(N-2)}$ and also the iterations associated for applying this solver is set to be "size (A, 1)." GMRES algorithm for solving the linear algebraic systems of the considered problems has a good performance and no preconditioning technique is needed for solving the algebraic systems.

```
Step 1: Collocating the initial condition (23):
for i = 1 : N + 1
for j = 1 : N + 1
A_in((i − 1) ∗ (N + 1) + j, :) = β(η_i, ξ_j, a_τ);
b_in((i − 1) ∗ (N + 1) + j, 1) = f(η_i, ξ_j);
end
end
Step 2: Collocating the boundary conditions (24):
→ x variable discretization:
for j = 1 : N + 1
for k = 2 : N + 1
A_bcl((j − 1) ∗ N + (k − 1), :) = β(a_η, ξ_j, τ_k);
A_bcr((j − 1) ∗ N + (k − 1), :) = β(b_η, ξ_j, τ_k);
b_bcl((j − 1) ∗ N + (k − 1), 1) = g_1(ξ_j, τ_k);
b_bcr((j − 1) ∗ N + (k − 1), 1) = g_2(ξ_j, τ_k);
end
→ y variable discretization:
for i = 2 : N
for k = 2 : N + 1
A_bcb((i − 2) ∗ N + (k − 1), :) = β(η_i, a_ξ, τ_k);
A_bct((i − 2) ∗ N + (k − 1), :) = β(η_i, b_ξ, τ_k);
b_bcl((i − 2) ∗ N + (k − 1), 1) = h_1(η_i, τ_k);
b_bcr((i − 2) ∗ N + (k − 1), 1) = h_2(η_i, τ_k);
end
Step 3: Collocating the main equation (22):
for i = 2 : N
for j = 2 : N
for k = 2 : N + 1
A_main((i − 2) ∗ N ∗ (N − 1) + (j − 2) ∗ N + (k − 1), :) = β(η_i, ξ_j, τ_k) ∗ ((M̂̂) − (M̃̃)^2 − (Ṁ)^2);
b_main((i − 2) ∗ N ∗ (N − 1) + (j − 2) ∗ N + (k − 1), 1) = ω(η_i, ξ_j, τ_k);
end
end
end
Step 4: Assembling the coefficient matrices and right hand side column vectors:
A = [A_main; A_bcl; A_bcr; A_bcb; A_bct; A_in];
b = [b_main; b_bcl; b_bcr; b_bcb; b_bct; b_in];
maxit = size(A, 1);
tol = 10^−(N−2);
U = gmres(A, b, [], tol, maxit)
```

Algorithm 3: Implementation of BCM for solving (22)–(24).

Example 1. As the first example we consider (1)–(3) with the following assumptions:

$$\begin{aligned} \omega(\eta,\tau) &= 0; \\ f(\eta) &= \exp(\eta); \\ g_1(\tau) &= \exp(\tau); \\ g_2(\tau) &= \exp(\tau+1), \end{aligned} \tag{33}$$

with the exact solution $\psi(\eta,\tau) = \exp(\eta+\tau)$ in the unit square $[a_\eta, b_\eta] \times [a_\tau, b_\tau] = [0,1] \times [0,1]$. We have implemented the BCM for solving this constructed example by assuming several values of N starting from 4 up to 14. It is observed that as N gets greater values, the associated errors decreased. For this example and the next example, we have defined the error function $e_N(\eta,\tau) = \psi(\eta,\tau) - \psi_N(\eta,\tau)$, where $\psi_N(\eta,\tau)$ is the approximate solution that is computed via applying BCM for solving the considered problems. Figures 5 and 6 depict the $e_N(\eta,\tau)$ associated with the obtained numerical solution of the this example for $N = 7, 9$ and $N = 11, 13$, respectively. From these figures, one can see the robustness of the proposed method for solving 1D diffusion equations.

Example 2 (see [6]). As the second example we consider (4)–(6) with the following assumptions:

$$\begin{aligned} \omega(\eta,\tau) &= 0; \\ f(\eta) &= \sin(\eta); \\ k(\eta) &= 0; \\ g_1(\tau) &= 0; \\ g_2(\tau) &= 0, \end{aligned} \tag{34}$$

Step 1: Collocating the initial condition (26):
for $i = 1 : N + 1$
for $j = 1 : N + 1$
$A_{\text{in}}((i-1) * (N+1) + j, :) = \beta(\eta_i, \xi_j, a_\tau)$;
$A_{\text{in1}}((i-1) * (N+1) + j, :) = \beta(\eta_i, \xi_j, a_\tau) * \widehat{\widehat{M}}$;
$b_{\text{in}}((i-1) * (N+1) + j, 1) = f(\eta_i, \xi_j)$;
$b_{\text{in1}}((i-1) * (N+1) + j, 1) = k(\eta_i, \xi_j)$;
end
end
Step 2: Collocating the boundary conditions (27):
$\rightarrow x$ variable discretization:
for $j = 1 : N + 1$
for $k = 3 : N + 1$
$A_{\text{bcl}}((j-1) * N + (k-2), :) = \beta(a_\eta, \xi_j, \tau_k)$;
$A_{\text{bcr}}((j-1) * N + (k-2), :) = \beta(b_\eta, \xi_j, \tau_k)$;
$b_{\text{bcl}}((j-1) * N + (k-2), 1) = g_1(\xi_j, \tau_k)$;
$b_{\text{bcr}}((j-1) * N + (k-2), 1) = g_2(\xi_j, \tau_k)$;
end
$\rightarrow y$ variable discretization:
for $i = 2 : N$
for $k = 3 : N + 1$
$A_{\text{bcb}}((i-2) * N + (k-2), :) = \beta(\eta_i, a_\xi, \tau_k)$;
$A_{\text{bct}}((i-2) * N + (k-2), :) = \beta(\eta_i, b_\xi, \tau_k)$;
$b_{\text{bcl}}((i-2) * N + (k-2), 1) = h_1(\eta_i, \tau_k)$;
$b_{\text{bcr}}((i-2) * N + (k-2), 1) = h_2(\eta_i, \tau_k)$;
end
Step 3: Collocating the main equation (25):
for $i = 2 : N$
for $j = 2 : N$
for $k = 3 : N + 1$
$A_{\text{main}}((i-2) * N * (N-1) + (j-2) * N + (k-2), :) = \beta(\eta_i, \xi_j, \tau_k) * ((\widehat{\widehat{M}})^2 - (\widetilde{\widetilde{M}})^2 - (\dot{M})^2)$;
$b_{\text{main}}((i-2) * N * (N-1) + (j-2) * N + (k-2), 1) = \omega(\eta_i, \xi_j, \tau_k)$;
end
end
end
Step 4: Assembling the coefficient matrices and right hand side column vectors:
$A = [A_{\text{main}}; A_{\text{bcl}}; A_{\text{bcr}}; A_{\text{bcb}}; A_{\text{bct}}; A_{\text{in}}; A_{\text{in1}}]$;
$b = [b_{\text{main}}; b_{\text{bcl}}; b_{\text{bcr}}; b_{\text{bcb}}; b_{\text{bct}}; b_{\text{in}}; b_{\text{in1}}]$;
maxit = size(A, 1);
tol = $10^{-(N-2)}$;
U = gmres(A, b, [], tol, maxit)

Algorithm 4: Implementation of BCM for solving (25)–(27).

Table 1: Numerical results of the BCM and TMM associated with the error $e_N(\eta, \tau)$ in the second example.

(η, τ)	BCM			TMM		
	$N = 8$	$N = 10$	$N = 12$	$N = 8$	$N = 10$	$N = 12$
$(0.6, 0.6)$	1.62×10^{-6}	6.90×10^{-11}	2.51×10^{-11}	8.26×10^{-7}	7.08×10^{-9}	4.63×10^{-11}
$(0.7, 0.7)$	1.37×10^{-6}	1.56×10^{-9}	2.29×10^{-11}	3.18×10^{-6}	3.69×10^{-8}	3.27×10^{-10}
$(0.8, 0.8)$	1.12×10^{-6}	2.84×10^{-9}	2.52×10^{-11}	1.01×10^{-5}	1.52×10^{-7}	1.75×10^{-9}
$(0.9, 0.9)$	1.22×10^{-6}	2.52×10^{-9}	2.22×10^{-11}	2.76×10^{-5}	5.24×10^{-7}	7.61×10^{-9}
$(1, 1)$	1.40×10^{-6}	1.59×10^{-9}	1.81×10^{-11}	6.70×10^{-5}	1.55×10^{-6}	2.76×10^{-9}

with the exact solution $\psi(\eta, \tau) = \sin(\eta)\cos(\tau)$ in the rectangle $[a_\eta, b_\eta] \times [a_\tau, b_\tau] = [0, \pi] \times [0, 1]$. Again, we have implemented the BCM for solving this provided example by assuming several values of N starting from 4 up to 14. Since this test problem has been selected from [6], we make some error comparisons with the Taylor matrix method (TMM) in Table 1 at some special points in the considered computational domain. From this table one can see that the errors associated with the BCM are more stable and accurate than TMM [6]. Since in [6] the authors considered

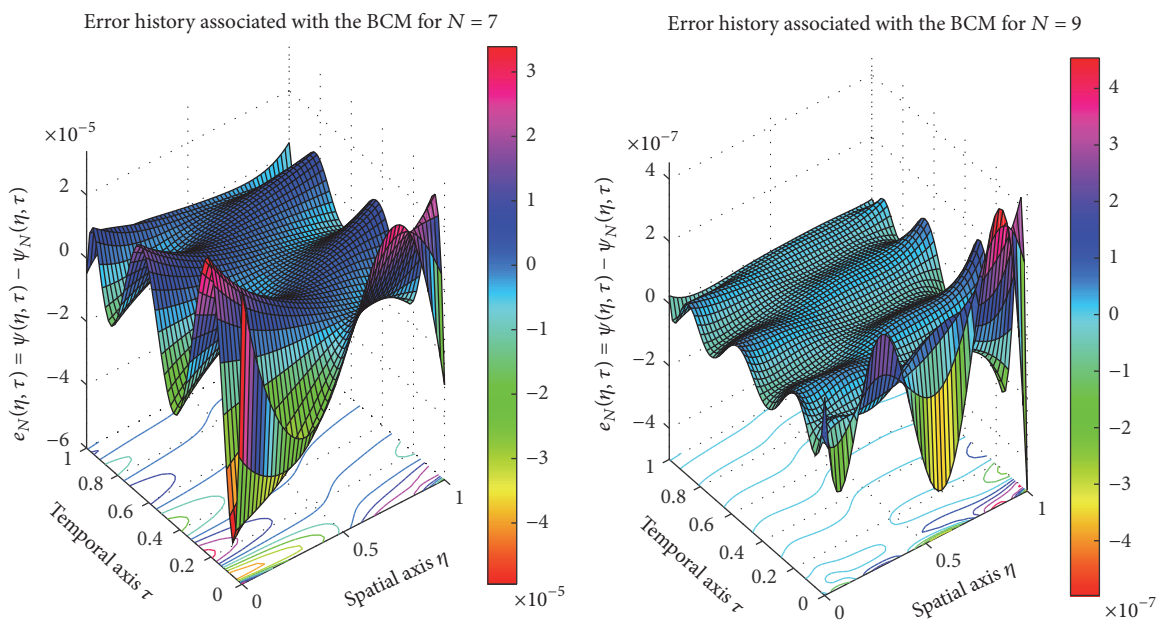

FIGURE 5: Error histories associated with the obtained numerical solution of the first example for $N = 7, 9$.

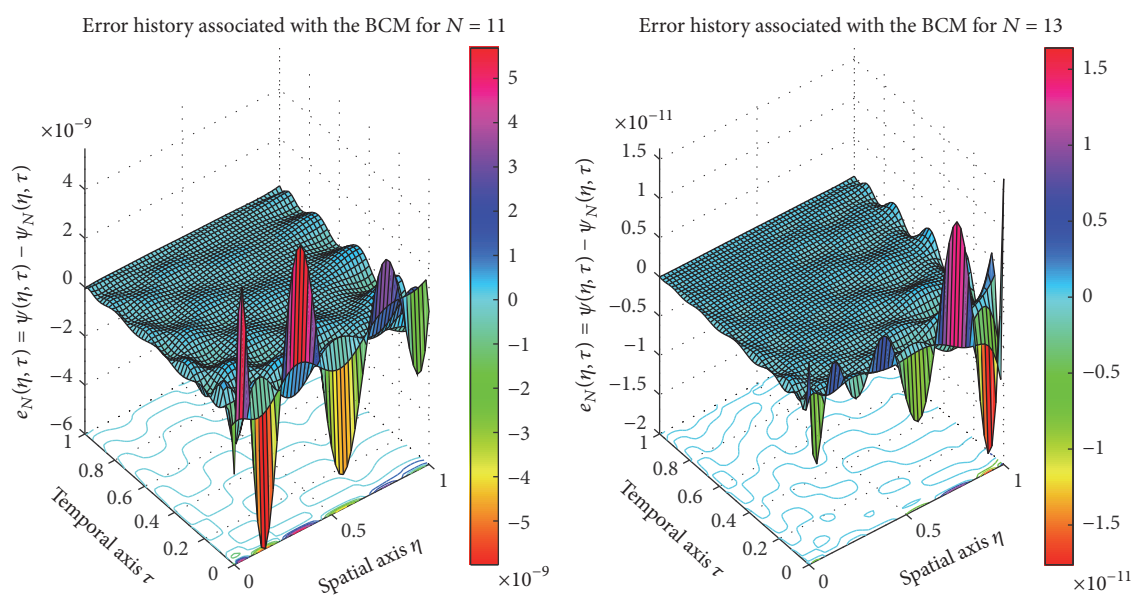

FIGURE 6: Error histories associated with the obtained numerical solution of the first example for $N = 11, 13$.

the Maclaurin approximation for the computational interval $[a_\eta, b_\eta] = [0, \pi]$, this assumption may decrease the accuracy of the TMM. Figures 7 and 8 illustrate $e_N(\eta,\tau)$ associated with the obtained numerical solution of this example via BCM for $N = 6, 8$ and $N = 10, 12$, respectively. From these figures, one can conclude that the proposed BCM is an efficient numerical approach for solving 1D wave equations. For instance, in Figure 8, the error function values $e_{12}(\eta,\tau)$ are around 10^{-11} approximately, while the associated error function values $e_{12}(\eta,\tau)$ of the TMM [6] are around 10^{-7} (see the fourth figure of [6] for this aim).

Example 3 (see [5]). As the third example we consider (22)–(24) with the following assumptions:

$$\begin{aligned} \omega(\eta,\xi,\tau) &= 0; \\ f(\eta,\xi) &= \exp(\eta+\xi); \\ g_1(\xi,\tau) &= \exp(\xi+2t); \\ g_2(\xi,\tau) &= \exp(\xi+2t+1); \\ h_1(\eta,\tau) &= \exp(\eta+2t); \\ h_2(\eta,\tau) &= \exp(\eta+2t+1), \end{aligned} \tag{35}$$

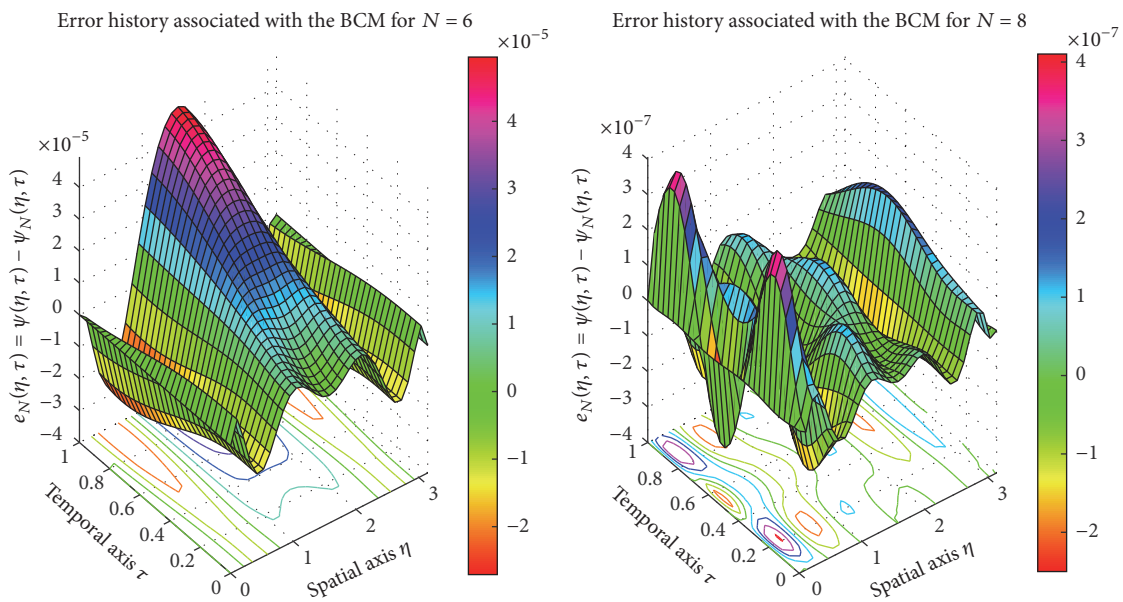

Figure 7: Error histories associated with the obtained numerical solution of the second example for $N = 6, 8$.

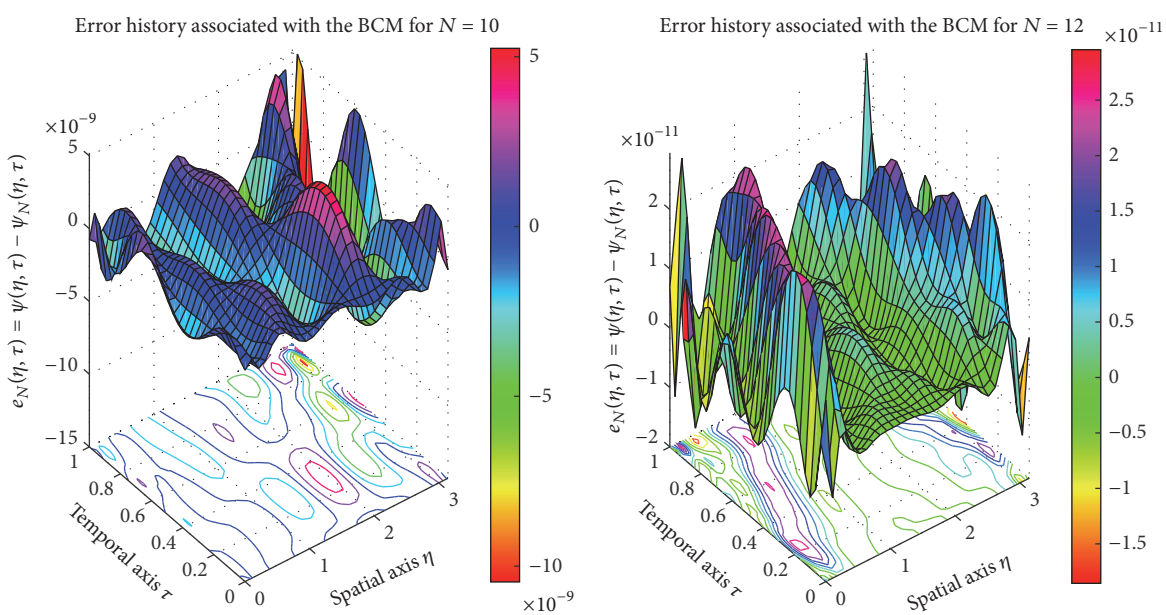

Figure 8: Error histories associated with the obtained numerical solution of the second example for $N = 10, 12$.

with the exact solution $\psi(\eta,\xi,\tau) = \exp(\eta + \xi + \tau)$ in the unit cube $[a_\eta, b_\eta] \times [a_\xi, b_\xi] \times [a_\tau, b_\tau] = [0,1] \times [0,1] \times [0,1]$. The numerical results of this example are provided with the numerical results of the next example in Table 2. Moreover, numerical solution $\psi_7(1,\xi,\tau)$, which is obtained via the proposed BCM, together with the exact solution $\psi(1,\xi,\tau)$ is depicted in Figure 9.

Example 4 (see [5]). As the fourth example we consider (22)–(24) with the following assumptions:

$$\begin{aligned} \omega(\eta,\xi,\tau) &= 0; \\ f(\eta,\xi) &= \cos(\pi\eta)\sin(\pi\xi); \\ g_1(\xi,\tau) &= \sin(\pi\xi)\exp\left(-2\pi^2\tau\right); \\ g_2(\xi,\tau) &= -\sin(\pi\xi)\exp\left(-2\pi^2\tau\right); \\ h_1(\eta,\tau) &= 0; \\ h_2(\eta,\tau) &= 0, \end{aligned} \tag{36}$$

with the exact solution $\psi(\eta,\xi,\tau) = \cos(\pi\eta)\sin(\pi\xi)\exp(-2\pi^2\tau)$ in the unit cube $[a_\eta, b_\eta] \times [a_\xi, b_\xi] \times [a_\tau, b_\tau] = [0,1] \times [0,1] \times [0,1]$. For solving this example and the previous example, we applied several

Table 2: Comparisons of the BCM and BMM associated with the errors E_N in the third and fourth example.

N	Third example		Fourth example	
	BCM	BMM	BCM	BMM
4	3.2×10^{-2}	3.5×10^{-2}	7.3×10^{-3}	2.8×10^{-2}
6	1.6×10^{-3}	2.2×10^{-3}	2.9×10^{-3}	6.1×10^{-3}
8	8.2×10^{-5}	1.7×10^{-4}	2.7×10^{-4}	1.4×10^{-3}
10	3.4×10^{-7}	1.4×10^{-5}	4.2×10^{-5}	3.7×10^{-4}
12	1.8×10^{-9}	1.1×10^{-6}	7.9×10^{-6}	9.1×10^{-5}
14	5.4×10^{-11}	9.2×10^{-8}	1.2×10^{-6}	2.3×10^{-5}

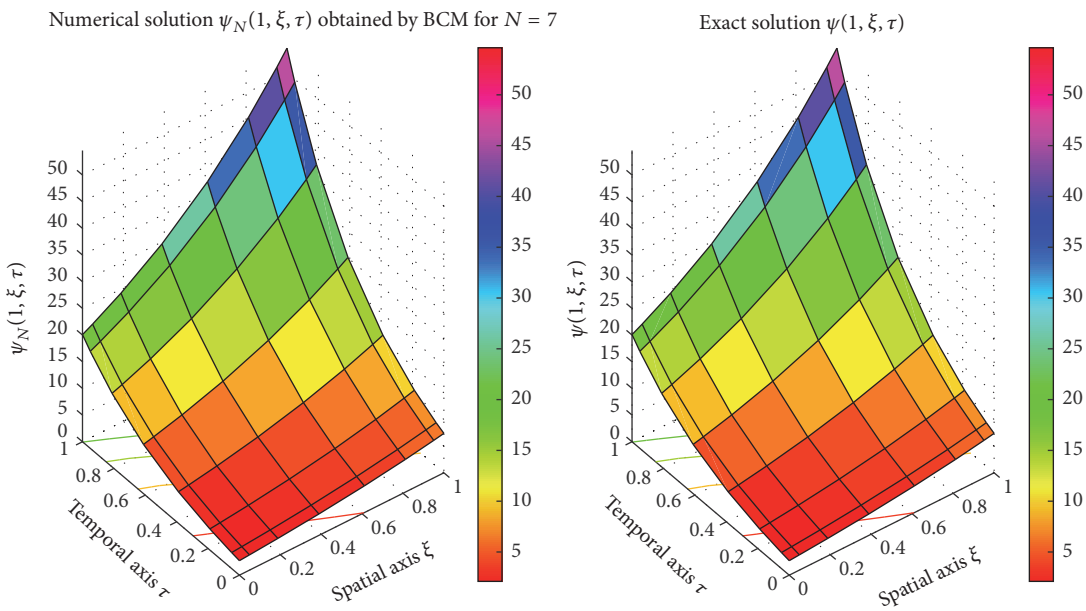

Figure 9: Numerical solution $\psi_7(1, \xi, \tau)$ and exact solution $\psi(1, \xi, \tau)$ of the third example.

values of N such as 4, 6, 8, 10, 12, and 14 and achieve the numerical solution $\psi_N(\eta, \xi, \tau)$ by the suggested BCM. For these numerical solutions, we have defined the discrete error function

$$E_N = \max_{1 \le i \le N+1, 1 \le j \le N+1} \left| \psi\left(\eta_i, \xi_j, 1\right) - \psi_N\left(\eta_i, \xi_j, 1\right) \right| \quad (37)$$

for both of the third and fourth examples and make a comparison with the Bernoulli matrix method (BMM) [5] in Table 2. From this table, one can conclude that the proposed BCM is more accurate with respect to the numerical solutions achieved via BMM [5]. Moreover, numerical solution $\psi_9(1, \xi, \tau)$, which is obtained via the proposed BCM, together with the exact solution $\psi(1, \xi, \tau)$ is depicted in Figure 10.

6. Conclusions

Applying analytical methods for solving multidimensional parabolic and hyperbolic PDEs usually is not efficient, because symbolic computations (such as direct differentiation and integration) in higher dimensions are time consuming. Therefore, numerical methods which are based on the operational matrices of differentiation (instead of direct differentiation) and high accurate Gauss quadrature rules (instead of direct integration) are more favorable for solving PDEs. In this paper, we apply a numerical approach, which is based on the Bernoulli polynomials and their operational matrices of differentiation and also the CGL collocation points for solving multidimensional diffusion and wave equations. In this regard, the above-mentioned equations will be reduced to the associated systems of linear algebraic equations which can be solved by robust iterative solvers such as GMRES algorithm. Moreover, geometrical distribution of collocation points for the considered PDEs is depicted for clarity of presentation. To show the robustness of the proposed method, some test problems are provided. In all of the considered numerical examples, as N gets greater, more accurate solutions will be achieved. Also, more accurate results (even in larger computational domain) in the second example are obtained with respect to the Taylor matrix method (TMM) [6] which confirm high accuracy of the BCM. The presented method is easy to implement in any software such as Matlab and Maple and can be generalized for solving three-dimensional PDEs. In this case,

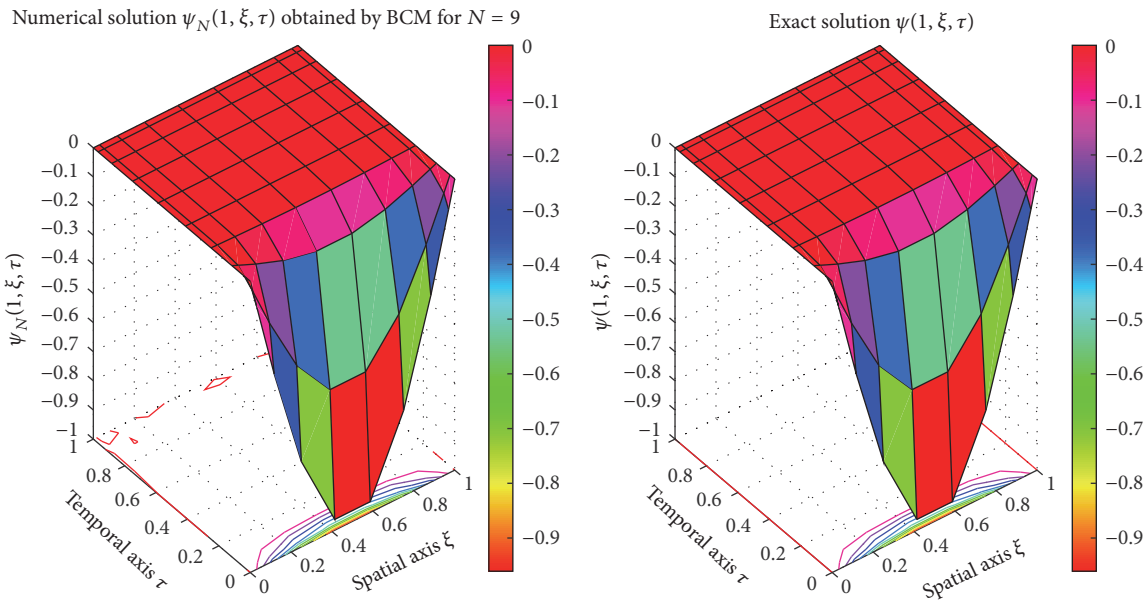

Figure 10: Numerical solution $\psi_9(1,\xi,\tau)$ and exact solution $\psi(1,\xi,\tau)$ of the fourth example.

BCM for collocating the space variables (x, y, z) should be combined with method of lines (MOL) [26, 27] to reduce three-dimensional PDEs into the associated systems of ODEs that can be integrated by some efficient ODEs solvers.

Competing Interests

The authors declare that they do not have any conflict of interests in their submitted paper.

References

[1] E. Tohidi and A. Kılıçman, "An efficient spectral approximation for solving several types of parabolic PDEs with nonlocal boundary conditions," *Mathematical Problems in Engineering*, vol. 2014, Article ID 369029, 6 pages, 2014.

[2] E. Tohidi, "Application of Chebyshev collocation method for solving two classes of non-classical parabolic PDEs," *Ain Shams Engineering Journal*, vol. 6, no. 1, pp. 373–379, 2015.

[3] S. A. Yousefi, M. Behroozifar, and M. Dehghan, "The operational matrices of Bernstein polynomials for solving the parabolic equation subject to specification of the mass," *Journal of Computational and Applied Mathematics*, vol. 235, no. 17, pp. 5272–5283, 2011.

[4] R. Zolfaghari and A. Shidfar, "Solving a parabolic PDE with nonlocal boundary conditions using the sinc method," *Numerical Algorithms*, vol. 62, no. 3, pp. 411–427, 2013.

[5] B. Zogheib and E. Tohidi, "A new matrix method for solving two-dimensional time-dependent diffusion equations with Dirichlet boundary conditions," *Applied Mathematics and Computation*, vol. 291, pp. 1–13, 2016.

[6] B. Bülbül and M. Sezer, "Taylor polynomial solution of hyperbolic type partial differential equations with constant coefficients," *International Journal of Computer Mathematics*, vol. 88, no. 3, pp. 533–544, 2011.

[7] B. Bülbül and M. Sezer, "A Taylor matrix method for the solution of a two-dimensional linear hyperbolic equation," *Applied Mathematics Letters*, vol. 24, no. 10, pp. 1716–1720, 2011.

[8] F. Mirzaee and S. Bimesl, "A new approach to numerical solution of second-order linear hyperbolic partial differential equations arising from physics and engineering," *Results in Physics*, vol. 3, pp. 241–247, 2013.

[9] A. Saadatmandi and M. Dehghan, "Numerical solution of hyperbolic telegraph equation using the Chebyshev tau method," *Numerical Methods for Partial Differential Equations*, vol. 26, no. 1, pp. 239–252, 2010.

[10] S. A. Hosseini, S. Shahmorad, and H. Masoumi, "Extension of the operational Tau method for solving 1-D nonlinear transient heat conduction equations," *Journal of King Saud University—Science*, vol. 25, no. 4, pp. 283–288, 2013.

[11] M. Zayernouri and G. E. Karniadakis, "Fractional spectral collocation method," *SIAM Journal on Scientific Computing*, vol. 36, no. 1, pp. A40–A62, 2014.

[12] G. Yuksel, O. R. Isik, and M. Sezer, "Error analysis of the Chebyshev collocation method for linear second-order partial differential equations," *International Journal of Computer Mathematics*, vol. 92, no. 10, pp. 2121–2138, 2015.

[13] S. Yüzbaşı and N. Şahin, "Numerical solutions of singularly perturbed one-dimensional parabolic convection–diffusion problems by the Bessel collocation method," *Applied Mathematics and Computation*, vol. 220, pp. 305–315, 2013.

[14] S. Yuzbasi, "Numerical solutions of hyperbolic telegraph equation by using the Bessel functions of first kind and residual correction," *Applied Mathematics and Computation*, vol. 287, pp. 83–93, 2016.

[15] F. Gao and C. Chi, "Unconditionally stable difference schemes for a one-space-dimensional linear hyperbolic equation," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1272–1276, 2007.

[16] M. E. Koksal, "An operator-difference method for telegraph equations arising in transmission lines," *Discrete Dynamics in Nature and Society*, vol. 2011, Article ID 561015, 17 pages, 2011.

[17] W. Xu, "Cell-centered finite difference method for parabolic equation," *Applied Mathematics and Computation*, vol. 235, pp. 66–79, 2014.

[18] J. Zhu and J. Qiu, "A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws," *Journal of Computational Physics*, vol. 318, pp. 110–121, 2016.

[19] E. Suli, *Lecture Notes on Finite Element Methods for Partial Differential Equations*, Mathematical Institute, University of Oxford, Oxford, UK, 2012.

[20] S. Phongthanapanich and P. Dechaumphai, "Combined finite volume element method for singularly perturbed reaction-diffusion problems," *Applied Mathematics and Computation*, vol. 209, no. 2, pp. 177–185, 2009.

[21] M. Dehghan and A. Ghesmati, "Solution of the second-order one-dimensional hyperbolic telegraph equation by using the dual reciprocity boundary integral equation (DRBIE) method," *Engineering Analysis with Boundary Elements*, vol. 34, no. 1, pp. 51–59, 2010.

[22] M. Lakestani and B. N. Saray, "Numerical solution of telegraph equation using interpolating scaling functions," *Computers & Mathematics with Applications*, vol. 60, no. 7, pp. 1964–1972, 2010.

[23] S. Pandit, M. Kumar, and S. Tiwari, "Numerical simulation of second-order hyperbolic telegraph type equations with variable coefficients," *Computer Physics Communications*, vol. 187, pp. 83–90, 2015.

[24] F. Toutounian and E. Tohidi, "A new Bernoulli matrix method for solving second order linear partial differential equations with the convergence analysis," *Applied Mathematics and Computation*, vol. 223, pp. 298–310, 2013.

[25] A. Napoli, "Solutions of linear second order initial value problems by using Bernoulli polynomials," *Applied Numerical Mathematics*, vol. 99, pp. 109–120, 2016.

[26] R. C. Mittal and R. Bhatia, "A numerical study of two dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method," *Applied Mathematics and Computation*, vol. 244, pp. 976–997, 2014.

[27] R. C. Mittal and R. Bhatia, "Numerical solution of second order one dimensional hyperbolic telegraph equation by cubic B-spline collocation method," *Applied Mathematics and Computation*, vol. 220, pp. 496–506, 2013.