

Research Article

Numerical Solution of a Class of Nonlinear Partial Differential Equations by Using Barycentric Interpolation Collocation Method

Hongchun Wu,¹ Yulan Wang², and Wei Zhang¹

¹Institute of Economics and Management, Jining Normal University, Jining 012000, Inner Mongolia, China

²Department of Mathematics, Inner Mongolia University of Technology, Hohhot 010051, China

Correspondence should be addressed to Yulan Wang; wylnei@163.com and Wei Zhang; jnsfxyzw@163.com

Received 12 October 2018; Accepted 3 December 2018; Published 30 December 2018

Academic Editor: Francisco R. Villatoro

Copyright © 2018 Hongchun Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Partial differential equations (PDEs) are widely used in mechanics, control processes, ecological and economic systems, chemical cycling systems, and epidemiology. Although there are some numerical methods for solving PDEs, simple and efficient methods have always been the direction that scholars strive to pursue. Based on this problem, we give the meshless barycentric interpolation collocation method (MBICM) for solving a class of PDEs. Four numerical experiments are carried out and compared with other methods; the accuracy of the numerical solution obtained by the present method is obviously improved.

1. Introduction

Consider the following partial differential equation:

$$u_{tt} + \gamma u_{xx} + g(u) = f(x, t), \quad x \in (a, b), \quad t \in [0, T], \quad (1)$$

the initial conditions are expressed as

$$\begin{aligned} u(x, 0) &= \phi(x), \\ u_t(x, 0) &= \psi(x), \end{aligned} \quad (2)$$
$$x \in [a, b],$$

and the boundary conditions are expressed as

$$\begin{aligned} u(a, t) &= p_0(t), \\ u(b, t) &= p_1(t), \end{aligned} \quad (3)$$
$$t \geq t_0,$$

or

$$\begin{aligned} u_x(a, t) &= p_0(t), \\ u_x(b, t) &= p_1(t), \end{aligned} \quad (4)$$
$$t \geq t_0,$$

where $u = u(x, t)$ represents an unknown function at position x and time t . γ is a known constant. If $g(u)$ is nonlinear, the equation is a nonlinear PDEs. If $g(u)$ is linear, then the equation is linear PDEs.

PDEs are paid more attention, and there are many studies on the numerical solution of PDEs. For PDEs, a quintic polynomial spline technique is used for the numerical solutions of fourth-order PDEs with Caputo time fractional derivative on a finite domain [1]. The adaptive grid Haar wavelet collocation method is used for the numerical solution of parabolic PDEs [2]. The numerical stability of PDEs with piecewise constant arguments is considered [3] along with a quantum algorithm for preparing states that encode solutions of nonhomogeneous linear PDEs [4]. A time discretization scheme approximates the time fractional PDEs [5]. We also consider the application of a domain-type method of fundamental solutions together with a Picard iteration scheme for solving nonlinear elliptic PDEs [6] and a deep learning-based approach that can handle general high-dimensional parabolic PDEs [7]. Couple double Laplace transform with iterative method solves nonlinear Klein-Gordon equation subject to initial and boundary conditions [8]. The variational iteration method (VIM) and homotopy perturbation method (HPM) are used for various kinds of nonlinear problems ([9–12]). Numerical

solutions to linear and nonlinear PDEs use the peridynamic differential operator [13]. Numerical collocation approach solves PDEs [14]. A modified method using wavelet intergrid operators known as new wavelet based full-approximation scheme similar to multigrid full-approximation scheme is developed for the numerical solution of nonlinear elliptic PDEs [15]. B-spline collocation method ([16–23]) had solved various kinds of nonlinear PDEs. Meshless Barycentric interpolation collocation method ([24–30]) is a high precision numerical method. This paper gives the meshless barycentric interpolation collocation method (MBICM) for solving a class of PDEs.

This paper is organized as follows. In Section 2, the meshless barycentric interpolation collocation method (MBICM) is introduced in detail. Several numerical examples are included to demonstrate the practicability of the method and some relevant figures, tables, and analyses are provided in Section 3. Some concluding remarks are given in Section 4.

2. Meshless Barycentric Interpolation Collocation Method

The MBICM uses the barycentric interpolation as the approximate function and uses the differential matrix of the barycentric interpolation to discrete the differential operator. It does not need to divide the calculation grid, and the calculation program is easy to write. It is an efficient and high precision meshless numerical method for calculating linear differential equations. For nonlinear PDEs, we construct a linearized iterative scheme to transform the equations into linear equations.

First, we introduce the MBICM for linear PDEs.

2.1. The Partial Differential Matrix of Barycentric Interpolation. For the nonlinear (1), we give an initial value $u_0(x, t)$ and construct a linear iterative scheme

$$(u_n)_{tt} + \gamma (u_n)_{xx} + g(u_{n-1}) = f(x, t), \quad (5)$$

$$n = 0, 1, 2, \dots,$$

Next, we give meshless barycentric interpolation collocation method of (5).

In the interval $[a, b]$ takes M different nodes, $a \leq x_1 < x_2 < \dots < x_M \leq b$, respectively, in the interval $[0, T]$ takes N different nodes, $0 \leq t_1 < t_2 < \dots < t_N \leq T$, respectively. These nodes can generate two dimensional nodes on the rectangular area $\Omega = [a, b] \times [0, T]$, as follows: $\{(x_i, t_j), i = 1, 2, \dots, M; j = 1, 2, \dots, N\}$.

The barycenter interpolation form of function $u(x, t)$ at the nodes $\{(x_i, t_j), i = 1, 2, \dots, M; j = 1, 2, \dots, N\}$ can be expressed as

$$u(x, t) = \sum_{i=1}^M \sum_{j=1}^N \xi_i(x) \eta_j(t) u_{ij}, \quad (6)$$

$$i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N,$$

where $\xi_i(x) = \prod_{k=1, k \neq i}^M (x - x_k) / \prod_{k=1, k \neq i}^M (x_i - x_k)$, $i = 1, 2, \dots, M$, $\eta_j(t) = \prod_{k=1, k \neq j}^N (t - t_k) / \prod_{k=1, k \neq j}^N (t_j - t_k)$,

$j = 1, 2, \dots, N$. Moreover, $\xi_i(x)$ is the barycentric interpolation basis function of M nodes on interval $[a, b]$, $\eta_j(t)$ is the barycentric interpolation basis function of N nodes on interval $[0, T]$. $u_{ij} = u(x_i, t_j)$, $i = 1, 2, \dots, M; j = 1, 2, \dots, N$ is the function value of $u(x, t)$ at the nodes $\{(x_i, t_j), i = 1, 2, \dots, M; j = 1, 2, \dots, N\}$.

According to formula (6), the partial derivative of $l + k$ order of function $u(x, t)$ can be written as

$$\frac{\partial^{l+k} u}{\partial x^l \partial t^k} = \sum_{i=1}^M \sum_{j=1}^N \xi_i^{(l)}(x) \eta_j^{(k)}(t) u_{ij}, \quad l, k = 1, 2, \dots, \quad (7)$$

and the function values of the formula (7) at the nodes (x_p, t_q) are

$$u^{(l,k)}(x_p, t_q) := \frac{\partial^{l+k} u(x_p, t_q)}{\partial x^l \partial t^k}$$

$$= \sum_{i=1}^M \sum_{j=1}^N \xi_i^{(l)}(x_p) \eta_j^{(k)}(t_q) u_{ij}, \quad (8)$$

$$p = 1, 2, \dots, M; \quad q = 1, 2, \dots, N.$$

The nodes on the interval $[a, b]$ and $[0, T]$ constitute two groups of column vectors respectively. They are defined as

$$x^0 = [x_1, x_2, \dots, x_M]^T, \quad (9)$$

$$t^0 = [t_1, t_2, \dots, t_N]^T,$$

then the matrixes X, T above the region Ω can be generated by the above column vectors x^0, t^0 , and the form of matrixes X and T is as follows:

$$X^T = [(x^0)^T, (x^0)^T, \dots, (x^0)^T]^T, \quad (10)$$

$$T = [t^0, t^0, \dots, t^0].$$

The matrixes X, T are respectively written as column vectors x, t of $N \times M$ dimension:

$$x = [X_1, X_2, \dots, X_{M \times N}]^T, \quad (11)$$

$$t = [T_1, T_2, \dots, T_{M \times N}]^T.$$

The relations between the components x_i, y_j of the formula (8) and the components X_K, Y_K of formula (10) are as follows:

$$X_k = X_{(i-1)N+j} = x_i, \quad (12)$$

$$T_k = T_{(i-1)N+j} = t_j,$$

$$i = 1, 2, \dots, M; \quad j = 1, 2, \dots, N; \quad k = 1, 2, \dots, M \times N.$$

The function values of formula (6) and formula (7) at the node form column vectors $u, u^{(l,k)}$, and they are as follows:

$$\begin{aligned} u &= [u_1, u_2, \dots, u_{M \times N}]^T, \\ u^{(l,k)} &= [u_1^{(l,k)}, u_2^{(l,k)}, \dots, u_{M \times N}^{(l,k)}]^T, \\ u_p &= u(X_p, T_p), \\ u_p^{(l,k)} &= u^{(l,k)}(X_p, T_p), \end{aligned} \quad (13)$$

$$p = 1, 2, \dots, M \times N.$$

Therefore, formula (8) can be expressed in the following matrix form:

$$u^{(l,k)} = D^{(l,k)}u, \quad (14)$$

in formula (12), $D^{(l,k)} = C^{(l)} \otimes D^{(k)}$ is the Kronecker product of matrixes $C^{(l)}$ and $D^{(k)}$, and it is also called $l+k$ order partial differential matrix at nodes $\{(x_i, t_j), i = 1, 2, \dots, M; j = 1, 2, \dots, N\}$. $C^{(l)}$ and $D^{(k)}$ are l, k order differential matrixes formed by barycenter interpolation at node interval $[a, b]$ and interval $[0, T]$, respectively.

Denote that

$$\begin{aligned} C^{(0)} &= I_M, \\ D^{(0)} &= I_N, \end{aligned} \quad (15)$$

where I_M and I_N are M order unit matrix and N order unit matrix, respectively.

By using notation (6), (7), and (14), the calculation formula of BICM for model (1) can be written in following matrix form:

$$[D^{(0,2)} + \gamma D^{(2,0)} + \text{diag}(g(u_{n-1}))]u = f(x, t). \quad (16)$$

Here, diag is a symbol of diagonal matrix composed of vectors. If the linear iterative scheme (5) is convergent, then $u_n(x, t) \rightarrow u(x, t) (n \rightarrow \infty)$. In the calculation process, we set a calculation accuracy $\varepsilon = 1e - 15$ or $\varepsilon = 1e - 10$, if $|u_n(x, t) - u_{n-1}(x, t)| \leq \varepsilon$, then the iteration stops, and at this point, approximate solution of nonlinear differential equations is obtained.

2.2. The Discrete Formula and Application Method of Initial Boundary Conditions. The discretization of initial boundary conditions requires the use of the barycenter interpolation formula (6). By acting on the initial boundary conditions, the discrete formulae of initial boundary conditions are given:

$$\sum_{i=1}^M \sum_{j=1}^N \xi_i(x_p) \eta_j(t_0) u_{ij} = \phi(x_p), \quad p = 1, 2, \dots, M; \quad (17)$$

$$\sum_{i=1}^M \sum_{j=1}^N \xi_i(x_p) \eta'_j(t_0) u_{ij} = \varphi(x_p), \quad p = 1, 2, \dots, M; \quad (18)$$

$$\sum_{i=1}^M \sum_{j=1}^N \xi_i(a) \eta_j(t_q) u_{ij} = p_0(t_q), \quad q = 1, 2, \dots, N; \quad (19)$$

$$\sum_{i=1}^M \sum_{j=1}^N \xi_i(b) \eta_j(t_q) u_{ij} = p_1(t_q), \quad q = 1, 2, \dots, N. \quad (20)$$

The boundary conditions are replaced by displacement method. The following is a description of the specific displacement process:

The first $N + 1$ equation, the first $2N + 1$ equation, ..., and the first $(M - 2)N + 1$ equation of the system of equations are replaced by M equations of the initial condition (17); the first $2N$ equation, the first $3N$ equation, ..., and the first $(M - 1)N$ of the system of equations are replaced by M equations of the initial condition (18); the first equation to the first N equation of the system of equations are separated by N equations of the boundary condition (19); the first $(M - 1)N + 1$ equation to the first $M \times N$ equation of the system of equations are replaced by N equations of the boundary condition (20).

3. Numerical Experiment

Now, numerical examples of nonlinear Klein-Gordon equations, other types of equations, and systems of equations are solved by using MBICM. In the following numerical examples, we compare the errors of different methods to illustrate the accuracy of the barycentric interpolation collocation method. The errors are expressed by the error norm L_2, L_∞ , and L_2, L_∞ error norms and absolute error are expressed as follows:

L₂ Error Norm

$$L_2(u) = \left(\sum_{i=1}^M |u(x_i, t) - u_n(x_i, t)|^2 \Delta x \right)^{1/2}, \quad (21)$$

L_∞ Error Norm

$$L_\infty(u) = \max_{1 \leq i \leq M} |u(x_i, t) - u_n(x_i, t)|, \quad (22)$$

Absolute Error

$$Ab(u) = |u(x_i, t) - u_n(x_i, t)|. \quad (23)$$

where $u(x_i, t)$ and $u_n(x_i, t)$ are exact and numerical solution, respectively.

Example 1 (see [31]). Consider the nonlinear Klein-Gordon equation (1) with $\gamma = -1$ and $g(u) = u^2$

$$u_{tt} - u_{xx} + u^2 = -x \cos t + x^2 \cos^2 t. \quad (24)$$

where $x \in (-1, 1), t > 0$. Equation (24) satisfies the initial conditions:

$$\begin{aligned} u(x, 0) &= x, \\ u_t(x, 0) &= 0, \\ x &\in [-1, 1], \end{aligned} \quad (25)$$

TABLE 1: Comparison of error norms at different time t for Example 1.

t	$O(k^2 + k^2h^2 + h^2)$ method[31]		$O(k^2 + k^2h^2 + h^4)$ method[31]		Present method	
	$L_\infty(u)$	$L_2(u)$	$L_\infty(u)$	$L_2(u)$	$L_\infty(u)$	$L_2(u)$
1	1.03E-09	7.01E-09	7.68E-10	4.91E-09	7.9192E-13	1.5457E-12
3	1.00E-09	6.59E-09	7.52E-10	4.69E-09	1.0825E-13	2.3010E-13
5	2.56E-10	1.29E-09	1.76E-10	9.46E-10	3.9573E-13	7.6876E-13
7	1.13E-09	7.47E-09	7.63E-10	5.11E-09	9.6356E-13	2.1556E-12
10	9.46E-10	5.84E-09	6.55E-10	3.98E-09	2.4178E-10	5.8307E-10

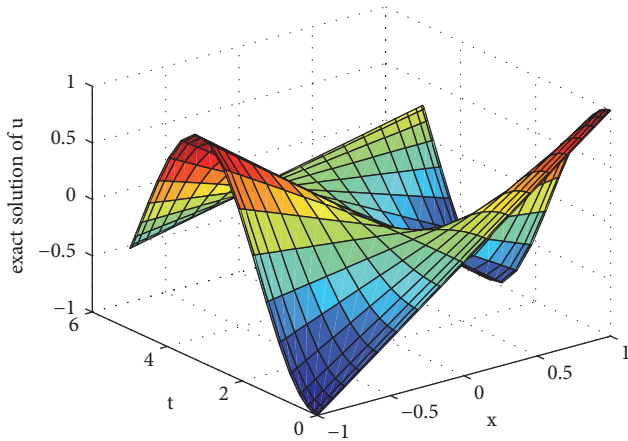


FIGURE 1: The exact solution of u for Example 1.

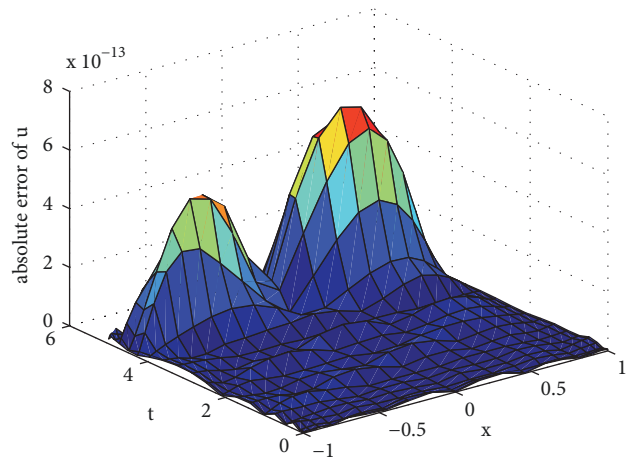


FIGURE 2: The absolute error of u for Example 1 ($t = 5, M = 20, N = 20$).

and with Dirichlet boundary conditions

$$\begin{aligned}
 u(-1, t) &= -\cos t, \\
 u(1, t) &= \cos t, \\
 t &\geq 0.
 \end{aligned}
 \tag{26}$$

The exact solution for this problem is

$$u(x, t) = x \cos t.
 \tag{27}$$

In Table 1, we enumerate the error norms in different time $t = 1, t = 3, t = 5, t = 7$ and $t = 10$ using different methods. By comparison, it is found that the present method is more accurate than other methods.

In Figure 1, the exact solution of u obtained by the present method at $t = 5$ with the number of nodes $M = 20, N = 20$. Figure 3, the numerical and exact solutions of u obtained by present method at different time $t = 1, t = 3, t = 5$ and $t = 7$ with the number of nodes $M = 20, N = 20$. From this we can see that the numerical solution and the exact solution have high reunification, so the accuracy of the present method is high.

In Figure 2, the absolute error of u obtained by present method at $t = 5$ with the number of nodes $M = 20, N = 20$. In Figure 4, the absolute errors of u are obtained by the present method at different times $t = 1, t = 3, t = 5$, and $t = 7$ with the number of nodes $M = 20, N = 20$. We can get the absolute error of u changing with x at $t = 3$ as the most stable.

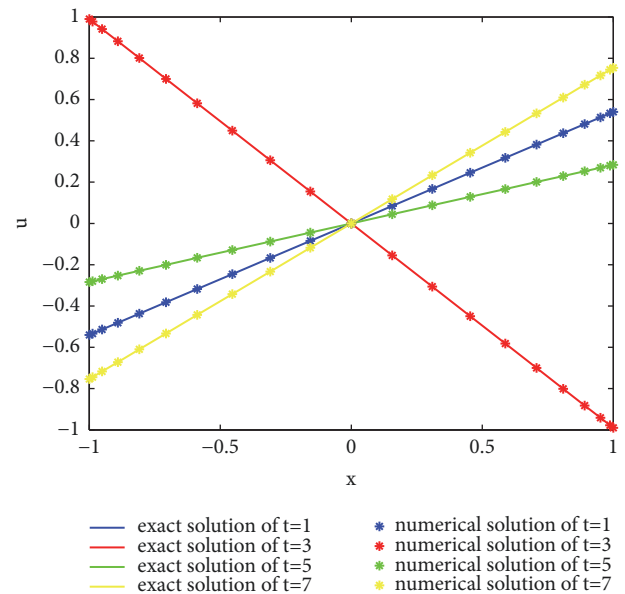


FIGURE 3: Comparison of numerical and exact solutions of u for Example 1 ($M = 20, N = 20$).

Example 2 (see [31]). Consider the nonlinear Klein-Gordon equation (1) with $\gamma = -1$ and $g(u) = (\pi^2/4)u + u^2$,

$$u_{tt} - u_{xx} + \frac{\pi^2}{4}u + u^2 = x^2 \sin^2\left(\frac{\pi}{2}t\right).
 \tag{28}$$

TABLE 2: Comparison of error norms at different time t for Example 2.

t	$O(k^2 + k^2h^2 + h^2)$ method[31]		$O(k^2 + k^2h^2 + h^4)$ method[31]		Present method	
	$L_\infty(u)$	$L_2(u)$	$L_\infty(u)$	$L_2(u)$	$L_\infty(u)$	$L_2(u)$
1	3.97E-06	2.71E-05	3.97E-06	2.71E-05	2.9723E-12	6.5955E-12
2	1.51E-06	8.97E-06	1.51E-06	8.97E-06	3.0783E-13	6.1985E-13
3	2.14E-06	1.49E-05	2.14E-06	1.49E-05	5.5911E-13	1.4550E-12
4	1.86E-06	1.05E-05	1.86E-06	1.05E-05	8.4829E-13	1.8202E-12
5	5.08E-06	3.36E-05	5.08E-06	3.36E-05	9.6175E-12	2.3494E-11

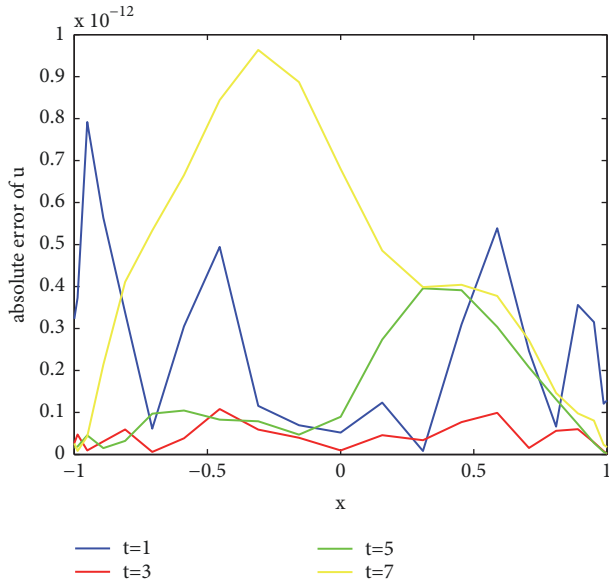


FIGURE 4: The absolute errors of u at different time t for Example 1 ($M = 20, N = 20$).

where $x \in (-1, 1), t > 0$. Equation (28) satisfies the initial conditions:

$$\begin{aligned}
 u(x, 0) &= 0, \\
 u_t(x, 0) &= \frac{\pi}{2}x, \\
 x &\in [-1, 1],
 \end{aligned}
 \tag{29}$$

and with Dirichlet boundary conditions

$$\begin{aligned}
 u(-1, t) &= -\sin\left(\frac{\pi}{2}t\right), \\
 u(1, t) &= \sin\left(\frac{\pi}{2}t\right), \\
 t &\geq 0.
 \end{aligned}
 \tag{30}$$

The exact solution for this problem is

$$u(x, t) = x \sin\left(\frac{\pi}{2}t\right).
 \tag{31}$$

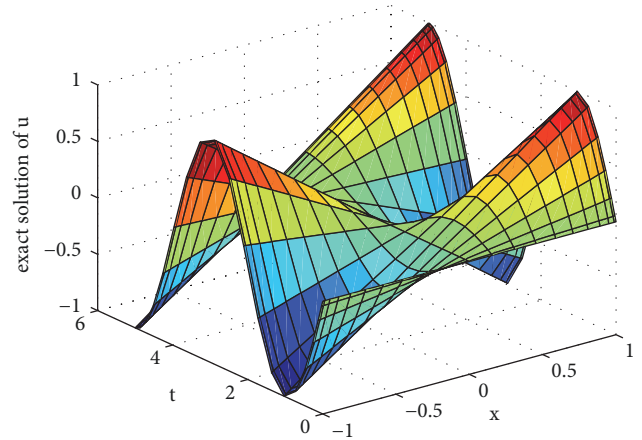


FIGURE 5: The exact solution of u for Example 2 ($t = 5, M = 20, N = 20$).

In Table 2, we list the error norms in different times $t = 1, t = 2, t = 3, t = 4$, and $t = 5$ using different methods. By comparison, we found that the present method is much more accurate than other methods.

In Figure 5, the exact solution of u is obtained by the present method at $t = 5$ with the number of nodes $M = 20, N = 20$. In Figure 7, numerical and exact solutions of u are obtained by the present method at different times $t = 1, t = 2, t = 3, t = 4$, and $t = 5$ with the number of nodes $M = 20, N = 20$. From this we can see that the numerical solution and the exact solution have high reunification, so the accuracy of the present method is high. Moreover, we find the monotonicity of function u at different time.

In Figure 6, the absolute error of u is obtained by the present method at $t = 5$ with the number of nodes $M = 20, N = 20$. In Figure 8, the absolute errors of u are obtained by the present method at different times $t = 1, t = 2, t = 3, t = 4$, and $t = 5$ with the number of nodes $M = 20, N = 20$. We can get the absolute errors of u changing with x at $t = 2, t = 3$ and $t = 4$ are the most stable. But the absolute error of u changing with x fluctuates greatly at $t = 5$.

Example 3 (see [32]). We consider nonlinear Hunter-Saxton equation:

$$u_{xt} = -uu_{xx} - \frac{1}{2}u_x^2.
 \tag{32}$$

TABLE 3: Comparing the results $t = 1, M = N = 15$ for Example 3.

x	Numerical solution	Exact solution	Absolute error
0	0.0000	0.0000	0.0002E-10
0.0432	0.0432	0.0432	0.0035E-10
0.1654	0.1654	0.1654	0.0166E-10
0.3455	0.3455	0.3455	0.0836E-10
0.5523	0.5523	0.5523	0.0829E-10
0.7500	0.7500	0.7500	0.0797E-10
0.9045	0.9045	0.9045	0.0971E-10
0.9891	0.9891	0.9891	0.1047E-10
1.0000	1.0000	1.0000	0.1053E-10

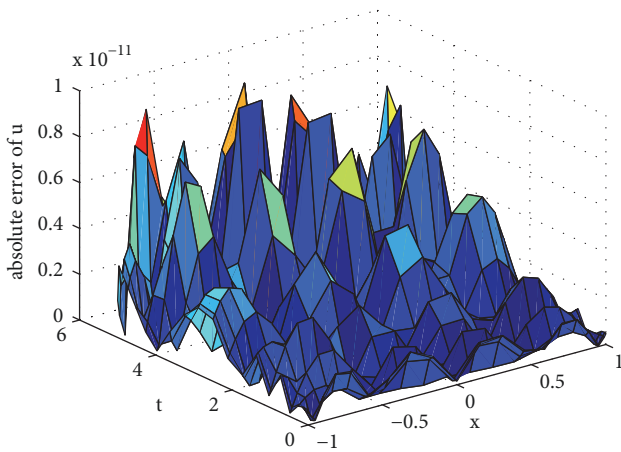


FIGURE 6: The absolute error of u for Example 2 ($t = 5, M = 20, N = 20$).

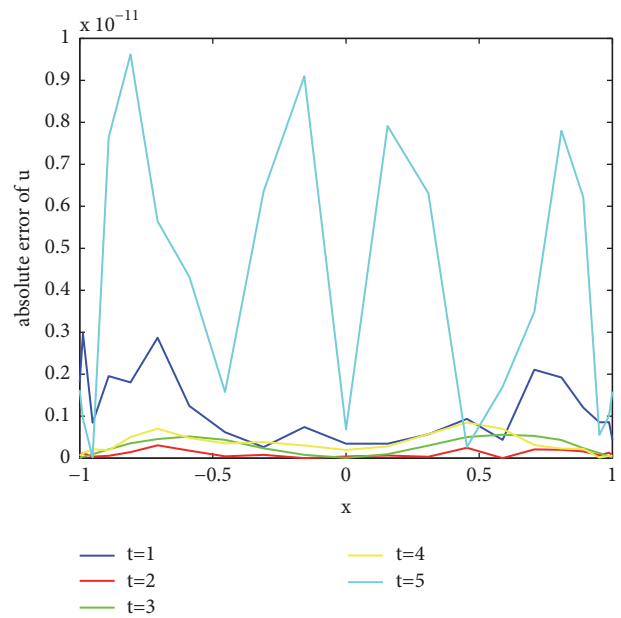


FIGURE 8: The absolute errors of u at different time t for Example 2 ($M = 20, N = 20$).

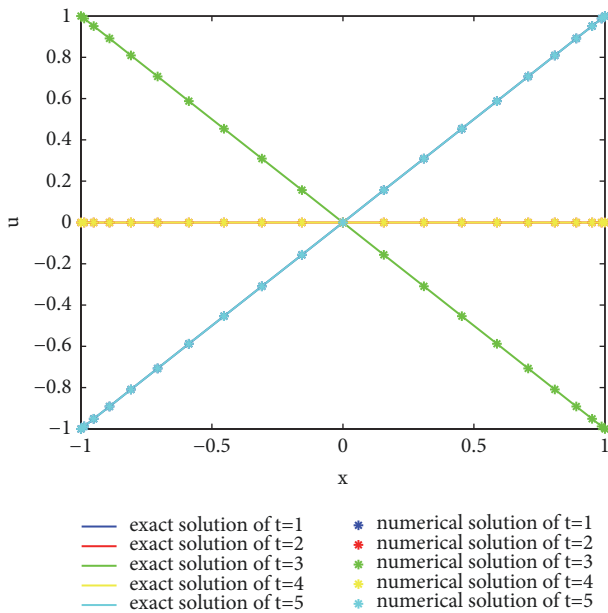


FIGURE 7: Comparison of numerical and exact solutions of u for Example 2 ($M = 20, N = 20$).

where $0 \leq x \leq 1, 0 \leq t \leq T$. Equation (32) satisfies the initial conditions:

$$\begin{aligned} u(x, 0) &= 2x, \\ u_t(x, 0) &= -2x, \end{aligned} \tag{33}$$

$x \in [0, 1],$

and the boundary conditions

$$\begin{aligned} u(0, t) &= 0, \\ u_x(0, t) &= \frac{2}{1+t}. \end{aligned} \tag{34}$$

The exact solution of this problem is as follows:

$$u(x, t) = \frac{2x}{1+t}. \tag{35}$$

In Tables 3 and 4, the absolute errors of u are obtained by the present method for different x with the number

TABLE 4: Comparing the results at $t = 0.1, M = N = 15$ for Example 3.

x	Numerical solution	Exact solution	Absolute error
0	0.0000	0	0.0017E-12
0.0432	0.0786	0.0786	0.0709E-12
0.1654	0.3008	0.3008	0.3931E-12
0.3455	0.6282	0.6282	0.2666E-12
0.5523	1.0041	1.0041	0.1452E-12
0.7500	1.3636	1.3636	0.1499E-12
0.9045	1.6446	1.6446	0.0708E-12
0.9891	1.7983	1.7983	0.0315E-12
1.0000	1.8182	1.8182	0.0298E-12

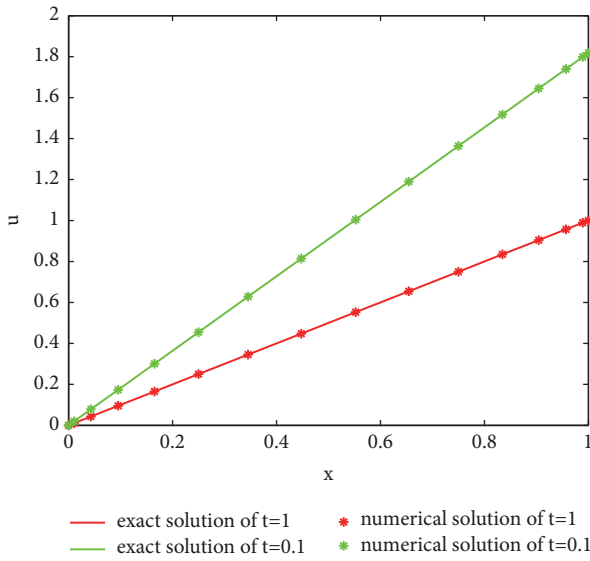


FIGURE 9: Comparison of numerical and exact solutions of u for Example 3 ($M = 15, N = 15$).

of nodes $M = 15, N = 15$ at $t = 1$ and $t = 0.1$, respectively. It can be found that the absolute error decreases with the decrease of time t in the same number of nodes and x .

In Figure 9, the numerical and exact solutions of u are obtained by the present method at different times $t = 1$ and $t = 0.1$ with the number of nodes $M = 15, N = 15$. From this we can see that the numerical solution and the exact solution have high reunification, so the accuracy of the present method is high. Moreover, the smaller the value of t is, the faster the function u changes with x . In Figure 10, the absolute error of u is obtained by the present method with the number of nodes $M = 15, N = 15$.

In Figure 11, the logarithmic absolute errors of u are obtained by the present method at different times $t = 1$ and $t = 0.1$ with the number of nodes $M = 15, N = 15$. In Figure 12, the absolute errors of u are obtained by the present method at different times $t = 1$ and $t = 0.1$ with the number of nodes $M = 15, N = 15$. Both figures show that the errors fluctuate and then tend to be stable with the increase of x at

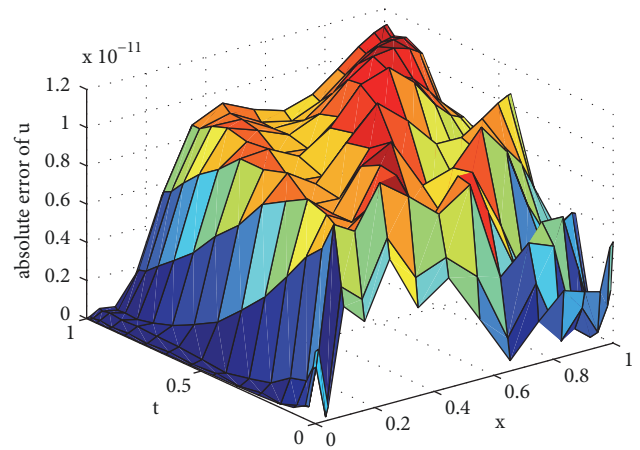


FIGURE 10: The absolute error of u for Example 3 ($t = 1, M = 15, N = 15$).

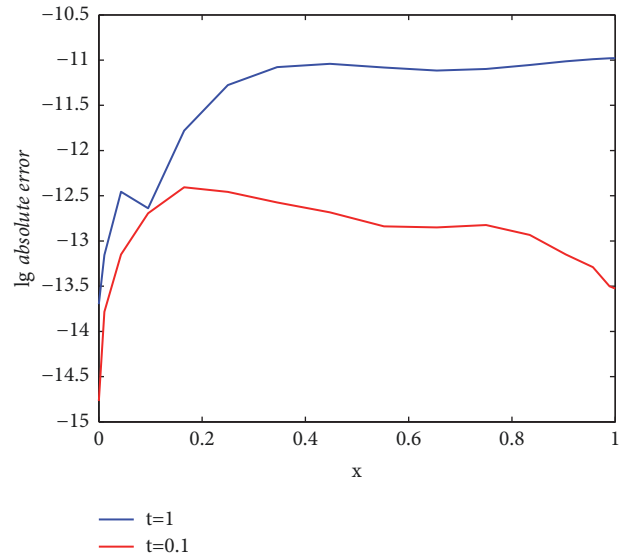


FIGURE 11: The logarithmic absolute errors of u for Example 3 ($M = 15, N = 15$).

$t = 1$; the errors fluctuate slightly and remain stable with the increase of x at $t = 0.1$.

TABLE 5: Comparison of error norms at different time t for Example 4.

t	$M = N = 15$		$M = N = 19$		$M = N = 23$	
	$L_{\infty}(u)$	$L_2(u)$	$L_{\infty}(u)$	$L_2(u)$	$L_{\infty}(u)$	$L_2(u)$
0.01	$2.1370E - 10$	$5.2854E - 10$	$3.3950E - 11$	$8.8235E - 11$	$2.5417E - 11$	$8.0727E - 11$
0.1	$9.2474E - 10$	$2.2981E - 09$	$2.5642E - 12$	$5.9925E - 12$	$3.3769E - 12$	$9.2082E - 12$
1	$9.8737E - 09$	$2.5369E - 08$	$8.4526E - 11$	$2.4854E - 10$	$2.5882E - 12$	$6.6491E - 12$

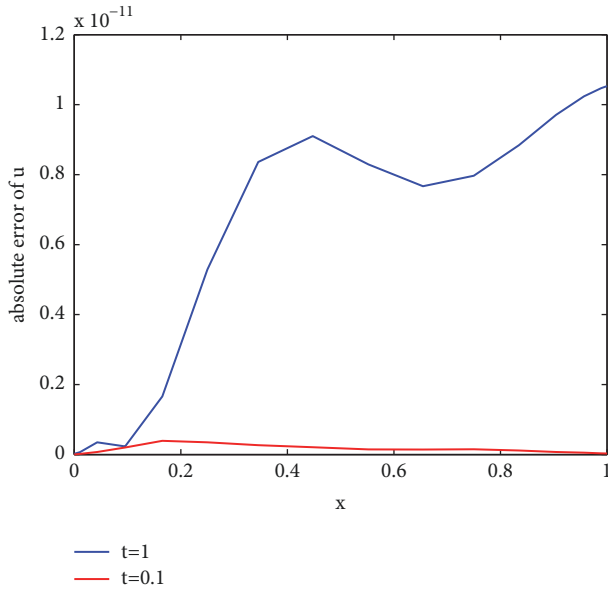


FIGURE 12: The absolute errors of u at different time t for Example 3 ($M = 15, N = 15$).

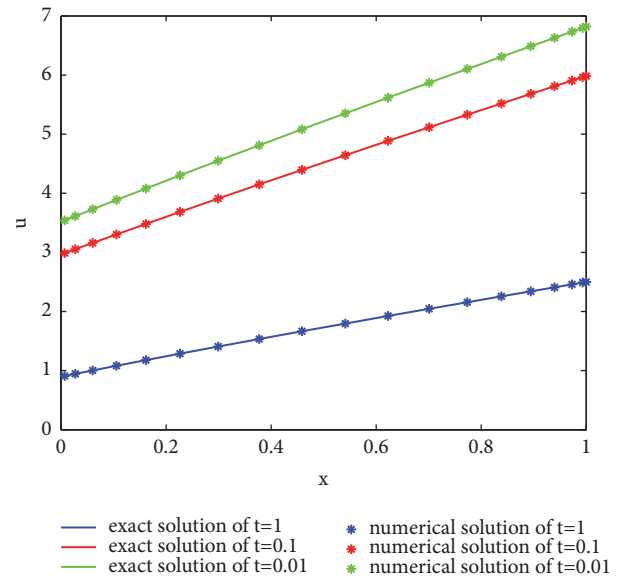


FIGURE 13: Comparison of numerical and exact solutions of u for Example 4 ($M = 19, N = 19$).

Example 4 (see [32]). For nonlinear Hunter-Saxton equation (32) with the following initial conditions

$$\begin{aligned}
 u(x, 0) &= (2 + 3x)^{2/3} + 2x + 2, \\
 u_t(x, 0) &= -4(x + 1) \left((2 + 3x)^{-1/3} + 1 \right), \\
 x &\in [0, 1].
 \end{aligned}
 \tag{36}$$

The exact solution is as follows:

$$u(x, t) = \frac{(2 + 3\nu)^{2/3} + 2\nu + 2}{(1 + t)^2}
 \tag{37}$$

where $\nu = x(1 + t)$.

In Table 5, there are the error norms of u with different values nodes M, N at $t = 1, t = 0.1$ and $t = 0.01$. We can see that the accuracy of u is improved with the increase of the number of nodes; when the number of nodes increases to a certain number, the accuracy tends to be stable; it shows that the convergence of the present method is better. On the other hand, u has the highest accuracy at $t = 0.1$.

In Figure 13, the numerical and exact solutions of u are obtained by the present method at different times $t = 1, t = 0.1$, and $t = 0.01$ with the number of nodes $M = 19, N = 19$. From this we can see that the numerical solution and the exact

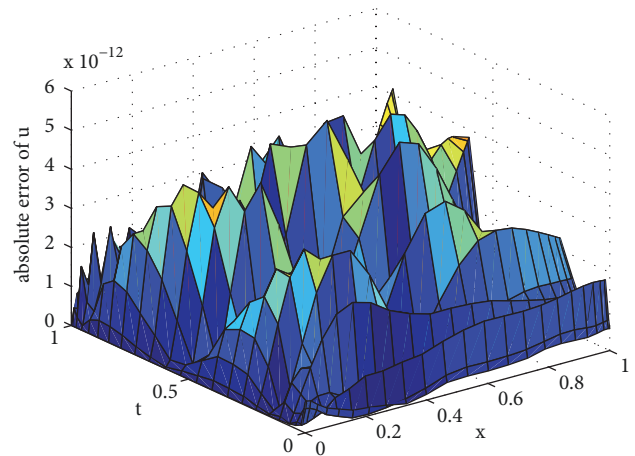


FIGURE 14: The absolute error of u for Example 4 ($t = 1, M = 23, N = 23$).

solution have high reunification. Moreover, the bigger the t , the smaller the u value. In Figure 14, the absolute error of u is obtained by the present method with the number of nodes $M = 23, N = 23$. The accuracy of the present method can be very high.

In Figure 15, the logarithmic absolute errors of u are obtained by the present method at different times $t = 1, t =$

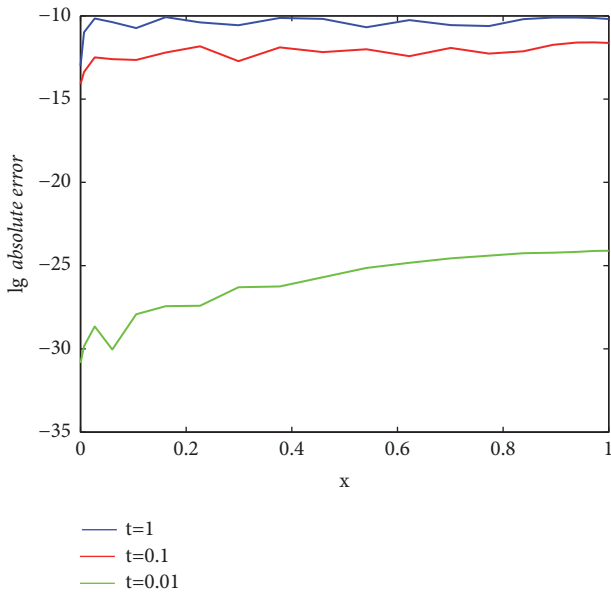


FIGURE 15: The logarithmic absolute errors of u for Example 4 ($M = 19, N = 19$).

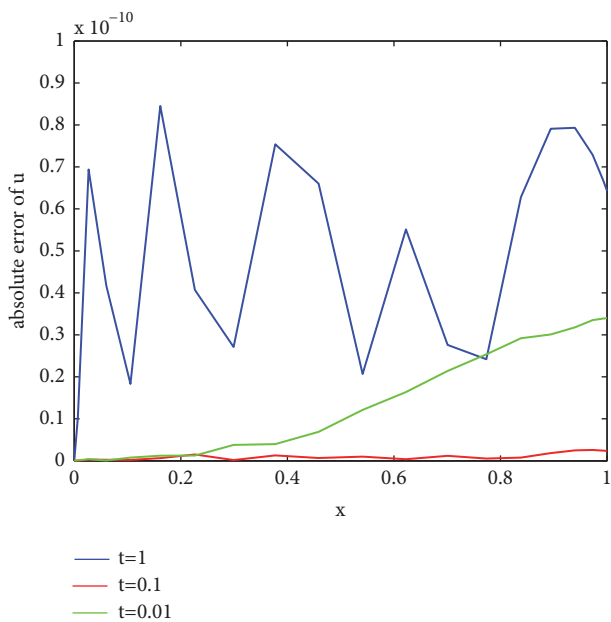


FIGURE 16: The absolute errors of u at different time t for Example 4 ($M = 19, N = 19$).

0.1, and $t = 0.01$ with the number of nodes $M = 19, N = 19$. In Figure 16, the absolute errors of u are obtained by the present method at different times $t = 1, t = 0.1$, and $t = 0.01$ with the number of nodes $M = 19, N = 19$. We can get that the absolute error of function u has obvious fluctuation at $t = 1$, and the absolute error of u is always relatively stable at $t = 0.1$.

4. Conclusions and Remarks

In this paper, we first transform a nonlinear partial differential equation into a linear differential equation by using the direct linearization iterative method and then solve the linear partial differential equation by using the meshless barycentric interpolation collocation method. The error estimation and convergence analysis of the method are proved. Through four examples and their charts, it is found that this method has higher computation accuracy than the existing methods. Finally, this method is applied to nonlinear partial differential equations and the calculated accuracy is higher.

All computations are performed by the MatlabR2007b software packages.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This paper is supported by the Natural Science Foundation of Inner Mongolia [2017MS0103], Inner Mongolia Maker Collaborative Innovation Center of Jining Normal University and the Numerical Analysis of Graduate Course Construction Project of Inner Mongolia University of Technology [grant number KC2014001], and the National Natural Science Foundation of China [11361037].

References

- [1] X. Li and P. J. Wong, "An efficient numerical treatment of fourth-order fractional diffusion-wave problems," *Numerical Methods for Partial Differential Equations*, vol. 34, no. 4, pp. 1324–1347, 2018.
- [2] S. C. Shiralashetti, L. M. Angadi, and A. B. Deshi, "Wavelet Based Lifting Schemes for the Numerical Solution of Parabolic Partial Differential Equations," *Journal of Computer and Mathematical Sciences*, vol. 9, no. 3, pp. 174–185, 2018.
- [3] Q. Wang, "Stability of numerical solution for partial differential equations with piecewise constant arguments," *Advances in Difference Equations*, Paper No. 71, 13 pages, 2018.
- [4] J. M. Arrazola, T. Kalajdzievski, and C. Weedbrook, *Quantum Algorithm for Non-Homogeneous Linear Partial Differential Equations*, 2018.
- [5] Y. Xing and Y. Yan, "A higher order numerical method for time fractional partial differential equations with nonsmooth data," *Journal of Computational Physics*, vol. 357, pp. 305–323, 2018.
- [6] C. J. Alves and A. L. Silvestre, "On the application of the method of fundamental solutions to nonlinear partial differential equations," *Engineering Analysis with Boundary Elements*, vol. 92, pp. 267–274, 2018.
- [7] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the*

- National Academy of Sciences of the United States of America*, vol. 115, no. 34, pp. 8505–8510, 2018.
- [8] R. R. Dhunde and G. L. Waghmare, “Solutions of some linear fractional partial differential equations in mathematical physics,” *The Journal of the Indian Mathematical Society*, vol. 85, no. 3-4, pp. 313–327, 2018.
- [9] T. M. Elzaki, “Solution of nonlinear partial differential equations by new laplace variational iteration method,” *Differential Equations-Theory and Current Research*, 2018.
- [10] J. He, “Approximate solution of nonlinear differential equations with convolution product nonlinearities,” *Computer Methods Applied Mechanics and Engineering*, vol. 167, no. 1-2, pp. 69–73, 1998.
- [11] J.-H. He, “Homotopy perturbation method: a new nonlinear analytical technique,” *Applied Mathematics and Computation*, vol. 135, no. 1, pp. 73–79, 2003.
- [12] J.-H. He and X.-H. Wu, “Construction of solitary solution and compacton-like solution by variational iteration method,” *Chaos, Solitons & Fractals*, vol. 29, no. 1, pp. 108–113, 2006.
- [13] E. Madenci, M. Dorduncu, A. Barut, and M. Futch, “Numerical solution of linear and nonlinear partial differential equations using the peridynamic differential operator,” *Numerical Methods for Partial Differential Equations*, vol. 33, no. 5, pp. 1726–1753, 2017.
- [14] K. Mredula and D. Vakaskar, “Collocation Method Using Wavelet for Partial Differential Equation,” *Indian Journal of Industrial and Applied Mathematics*, vol. 8, no. 1, p. 14, 2017.
- [15] S. C. Shiralashetti, M. H. Kantli, and A. B. Deshi, “New wavelet based full-approximation scheme for the numerical solution of nonlinear elliptic partial differential equations,” *Alexandria Engineering Journal*, vol. 55, no. 3, pp. 2797–2804, 2016.
- [16] S. A. Khuri and A. Sayfy, “A spline collocation approach for a generalized wave equation subject to non-local conservation condition,” *Applied Mathematics and Computation*, vol. 217, no. 8, pp. 3993–4001, 2010.
- [17] G. Arora and B. K. Singh, “Numerical solution of Burgers’ equation with modified cubic B-Spline differential quadrature method,” *Applied Mathematics and Computation*, vol. 224, pp. 166–177, 2013.
- [18] H. S. Shukla, M. Tamsir, V. K. Srivastava, and J. Kumar, “Numerical solution of two dimensional coupled viscous Burger equation using modified cubic B-spline differential quadrature method,” *AIP Advances*, vol. 4, no. 11, 2014.
- [19] H. S. Shukla, M. Tamsir, and V. K. Srivastava, “Numerical simulation of two dimensional sine-Gordon solitons using modified cubic B-spline differential quadrature method,” *AIP Advances*, vol. 5, no. 1, 2015.
- [20] H. S. Shukla, M. Tamsir, V. K. Srivastava, and M. M. Rashidi, “Modified cubic B-spline differential quadrature method for numerical solution of three-dimensional coupled viscous BURger equation,” *Modern Physics Letters B. Condensed Matter Physics, Statistical Physics, Applied Physics*, vol. 30, no. 11, 1650110, 17 pages, 2016.
- [21] M. Tamsir, V. K. Srivastava, N. Dhiman, and A. Chauhan, “Numerical computation of nonlinear Fisher’s reaction-diffusion equation with exponential modified cubic B-spline differential quadrature method,” *International Journal of Applied and Computational Mathematics*, vol. 4, no. 1, Art. 6, 13 pages, 2018.
- [22] M. S. Hashmi, M. Awais, A. Waheed, and Q. Ali, “Numerical treatment of Hunter Saxton equation using cubic trigonometric B-spline collocation method,” *AIP Advances*, vol. 7, no. 9, 2017.
- [23] N. Dhiman and M. Tamsir, “A collocation technique based on modified form of trigonometric cubic B-spline basis functions for Fisher’s reaction-diffusion equation,” *Multidiscipline Modeling in Materials and Structures*, vol. 14, no. 5, pp. 923–939, 2018.
- [24] S. Li, Z. Wang, and C. Yuan, “Barycentric interpolation collocation method for solving elastic problems,” *Journal of Central South University (Science and Technology)*, vol. 44, no. 5, pp. 2031–2040, 2013.
- [25] S. P. Li and Z. Q. Wang, “Barycentric Interpolation Collocation Method for Solving Nonlinear Vibration Problems,” *Noise Vibration Control*, vol. 28, pp. 49–52, 2018.
- [26] S. P. Li and Z. Q. Wang, *Barycentric interpolation collocation method for nonlinear problems*, National defense industry press, 2015.
- [27] S. P. Li and Z. Q. Wang, *High-Precision Non-Grid Center of Gravity Interpolation Collocation Method: Algorithm, Program and Engineering Application*, Science Press, 2012.
- [28] H. Liu, J. Huang, Y. Pan, and J. Zhang, “Barycentric interpolation collocation methods for solving linear and nonlinear high-dimensional Fredholm integral equations,” *Journal of Computational and Applied Mathematics*, vol. 327, pp. 141–154, 2018.
- [29] W.-H. Luo, T.-Z. Huang, X.-M. Gu, and Y. Liu, “Barycentric rational collocation methods for a class of nonlinear parabolic partial differential equations,” *Applied Mathematics Letters*, vol. 68, pp. 13–19, 2017.
- [30] F. Liu, Y. Wang, and S. Li, “Barycentric interpolation collocation method for solving the coupled viscous Burgers’ equations,” *International Journal of Computer Mathematics*, vol. 95, no. 11, pp. 2162–2173, 2018.
- [31] J. Rashidinia and R. Mohammadi, “Tension spline approach for the numerical solution of nonlinear Klein-Gordon equation,” *Computer Physics Communications*, vol. 181, no. 1, pp. 78–91, 2010.
- [32] K. Parand and M. Delkosh, “An efficient numerical solution of nonlinear Hunter-Saxton equation,” *Communications in Theoretical Physics*, vol. 67, no. 5, pp. 483–492, 2017.

