Computational and Applied Mathematics (2024) 43:16
https://doi.org/10.1007/s40314-023-02530-4

A coupled scheme based on uniform algebraic trigonometric tension B-spline and a hybrid block method for Camassa-Holm and Degasperis-Procesi equations

Anurag Kaur[1] · V. Kanwar[2] · Higinio Ramos[3]

Received: 31 December 2022 / Revised: 20 August 2023 / Accepted: 6 November 2023

Abstract
In this article, high temporal and spatial resolution schemes are combined to solve the Camassa-Holm and Degasperis-Procesi equations. The differential quadrature method is strengthened by using modified uniform algebraic trigonometric tension B-splines of order four to transform the partial differential equation (PDE) into a system of ordinary differential equations. Later, this system is solved considering an optimized hybrid block method. The good performance of the proposed strategy is shown through some numerical examples. The stability analysis of the presented method is discussed. This strategy produces a saving of CPU-time as it involves a reduced number of grid points.

Keywords Degasperis-Procesi equation · Camassa-Holm equation · Uniform algebraic trigonometric tension B-splines · Hybrid block method · Differential quadrature method

Mathematics Subject Classification 65N35 · 35F50

Communicated by Jose Alberto Cuminato.

✉ Higinio Ramos
higra@usal.es

Anurag Kaur
ahsikh@gmail.com

V. Kanwar
vmithil@yahoo.co.in

[1] Department of Mathematics, Panjab University, Chandigarh, India

[2] University Institute of Engineering and Technology, Panjab University, Chandigarh, India

[3] Departamento de Matemática Aplicada, University of Salamanca, Salamanca, Spain

Published online: 11 December 2023

1 Introduction

Many complex physical phenomena can be depicted by nonlinear partial differential equations, one of the famous types being of the form

$$(1-\partial_z^2)w_t = \phi(w, w_z, w_{zz}, w_{zzz}),\ a < z < b,\ t > 0, \tag{1}$$

where ϕ is a multivariate polynomial function. Some prominent high-order nonlinear dispersive partial differential equations (PDEs) of the form (1) are the Camassa-Holm (CH) equation and the Degasperis-Procesi (DP) equation, which have many real-life applications. The CH equation describes the propagation of shallow water waves, where $w(z, t)$ represents the propagation of waves at the free surface of water (Camassa and Holm 1993) and the propagation of nonlinear waves in cylindrical hyper-elastic rods (Dai 1998). The Camassa-Holm equation and its modified form are given respectively by

$$w_t - w_{zzt} + 3ww_z - 2w_z w_{zz} - ww_{zzz} = 0,\ a < z < b,\ t > 0, \tag{2}$$
$$w_t - w_{zzt} + 3w^2 w_z - 2w_z w_{zz} - ww_{zzz} = 0,\ a < z < b,\ t > 0, \tag{3}$$

with initial and boundary conditions

$$\begin{aligned} w(z, 0) =& w_0, \\ w(a, t) = f_1(t),\ & w(b, t) = f_2(t). \end{aligned} \tag{4}$$

Another celebrated equation of type (1) is the Degasperis-Procesi (DP) equation, which models nonlinear dynamic propagation of shallow waters (Coclite and Karlsen 2006). The DP equation and its modified form are given as follows

$$w_t - w_{zzt} + 4ww_z - 3w_z w_{zz} - ww_{zzz} = 0,\ a < z < b,\ t > 0, \tag{5}$$
$$w_t - w_{zzt} + 4w^2 w_z - 3w_z w_{zz} - ww_{zzz} = 0,\ a < z < b,\ t > 0. \tag{6}$$

There is a need for an accurate numerical scheme to solve the CH equation and the DP equation since there is a lack of smoothness in the solutions and also the presence of the non-linear term containing the third order derivative, ww_{zzz}, requires a good approximation. Several numerical methods have been applied to the CH and DP equations, such as the Galerkin method, finite difference, quartic B-spline collocation, quasi-interpolation, meshfree methods, Jacobi wavelet method, and many others (Çelik 2022; Cheng and Wei 2013; Ganji et al. 2008; Hejazi and Mohammadi 2022; Jan et al. 2022; Shaheen et al. 2022; Wasim et al. 2018; Yıldırım 2010; Zhang et al. 2008).

Of particular importance, Bellman et al. (1972) introduced the differential quadrature method (DQM) in 1972. Later, Quan and Chang (1989a, b) contributed to facilitate the calculation of the weighting coefficients, and Shu and Richards (1992) gave the recurrence formula to find those weighting coefficients for the approximation of higher order derivatives. To enhance the DQM, many test functions like Lagrange polynomials, Legendre polynomials, B-splines, and others have been used. B-splines are one of the celebrated basis as it has shape preserving properties and a uniform mathematical representation. Nevertheless, it has certain limitations (Mainar et al. 2001). Polynomial B-splines might not perfectly preserve the shape of the original data, especially when fitting curves to noisy or irregular data. While higher-degree polynomials can provide smoother curves, they also come with computational challenges, such as increased numerical instability and higher computational complexity for both evaluation and manipulation. So, many alternatives to B-splines exist by adding trigonometric functions, exponential functions, and polynomials. A unified form

of splines that includes all such splines over a common space has been introduced, named as unified extended splines (UE-splines) (Wang 2008) that inherited desirable properties of polynomial B-splines. Further, choosing certain value of a tension parameter in UE-splines, uniform algebraic trigonometric (UAT) tension B-splines have been used (Alinia and Zarebnia 2019, 2018). DQM based on UAT-tension B-splines has been implemented recently for solving a system of coupled Burger's equations (Kapoor and Joshi 2021). In this work, we will implement a DQM based on UAT-tension B-splines of order four to transform the considered partial differential equation (PDE) into a system of ordinary differential equations (ODEs).

Several methods, including finite difference schemes and multi-stage ones like Strong Stability Preserving Runge–Kutta methods, may be used for solving initial-value problems. A key drawback in these approaches is the requirement for small grid sizes to obtain acceptable accuracy and stability. These limitations can be overcome by using hybrid block approaches. Data evaluation in hybrid techniques occurs at off-step nodes, which usually leads to zero-stability. Hybrid block methods can also change the step size during simulation and overcome the Dahlquist barrier (Dahlquist 1956). As a result, these methods are more affordable (Ramos and Popescu 2018; Singla et al. 2022) and capable of handling complex systems of equations (Singh et al. 2019). Hybrid block approaches have been used recently (Ramos et al. 2022; Kaur and Kanwar 2022) to solve parabolic PDEs. In this study, we use an efficient one-step hybrid block technique to solve the resulting system of ordinary differential equations (ODEs).

2 Description of the numerical method

The differential quadrature method (DQM) is seen as a potential alternative to other numerical schemes like finite differences, collocation, or the finite element method. The efficiency of the DQM largely depends on the choice of trial functions. In this paper, a uniform algebraic trigonometric (UAT) tension B-spline base has been used. The choice is made because of the desirable inherited properties of polynomial splines as well as those of trigonometric splines.

2.1 UAT-Tension B-spline

Consider a uniform grid on the spatial variable, with $N+1$ nodes given by $a = z_0 < z_1 < z_2 < \cdots < z_N = b$. Let $h = z_{i+1} - z_i,\ i = 0, 1, \ldots, N-1$, be the step size. According to Wang (2008), the UE-splines of order 2 are defined as follows

$$\mathbb{B}_{l,2}(z) = \begin{cases} \frac{\sin[\tau(z-z_{l-2})]}{\sin[\tau(z_{l-1}-z_{l-2})]} & z \in [z_{l-2}, z_{l-1}), \\ \frac{\sin[\tau(z_l-z)]}{\sin[\tau(z_l-z_{l-1})]} & z \in [z_{l-1}, z_l), \\ 0, \quad \text{otherwise}, \end{cases} \tag{7}$$

where $\tau = \sqrt{\eta}$, is a tension parameter such that $\eta \in \mathbb{R}$ and $\eta \leq (\pi/h)^2$. Higher order UE-splines $\mathbb{B}_{l,j}$ for $j \geq 3$ are obtained recursively by

$$\mathbb{B}_{l,j}(z) = \int_{-\infty}^{z} \Big(\sigma_{l,j-1}\mathbb{B}_{l,j-1}(z) - \sigma_{l+1,j-1}\mathbb{B}_{l+1,j-1}(z)\Big)dz,\ \sigma_{l,j} = \Big(\int_{-\infty}^{\infty} \mathbb{B}_{l,j}(z)dz\Big)^{-1}. \tag{8}$$

Table 1 Values of $\mathbb{B}_{l,4}(z)$ and its derivatives at the grid points

z	z_{l-2}	z_{l-1}	z_l	z_{l+1}	z_{l+2}
$\mathbb{B}_{l,4}(z)$	0	d_1	d_2	d_1	0
$\mathbb{B}'_{l,4}(z)$	0	d_3	0	d_4	0

Using equations (7, 8) and taking $\eta > 0$, we get a subclass of UE-splines, uniform algebraic trigonometric (UAT) tension B-splines of order 4, $\mathbb{B}_{l,4}(z)$ at knots are given by

$$\mathbb{B}_{l,4}(z) = \begin{cases} \frac{\sigma_{l,2}\sigma_{l,3}}{\tau\sin(\tau h)}\Big[(z-z_{l-2}) - \frac{\sin(\tau(z-z_{l-2}))}{\tau}\Big], & z\in[z_{l-2},z_{l-1}),\\ \\ \sigma_{l,3}\Big[\frac{\sigma_{l,2}}{\tau\sin(\tau h)}\Big((z_{l-1}-z_{l-2}) - \frac{\sin(\tau h)}{\tau}\Big) + (z-z_{l-1})\\ -\frac{\sigma_{l,2}}{\tau\sin(\tau h)}\Big((z-z_{l-1}) - \frac{1}{\tau}(\sin(\tau(z-z_l)) + \sin(\tau(z_l-z)))\Big)\\ -\frac{\sigma_{l+1,2}}{\tau\sin(\tau h)}\Big((z-z_{l-1}) - \frac{\sin(\tau(z-z_{l-1}))}{\tau}\Big)\Big]\\ -\frac{\sigma_{l+1,2}\sigma_{l+1,3}}{\tau\sin(\tau h)}\Big[(z-z_{l-1}) - \frac{\sin(\tau(z-z_{l-1}))}{\tau}\Big], & z\in[z_{l-1},z_l),\\ \\ 1 - \frac{\sigma_{l+1,2}\sigma_{l,3}}{\tau\sin(\tau h)}\Big[(z_{l+1}-z) + \frac{\sin(\tau(z-z_{l+1}))}{\tau}\Big]\\ -\sigma_{l+1,3}\Big[\frac{\sigma_{l+1,2}}{\tau\sin(\tau h)}\Big((z_l-z_{l-1}) - \frac{\sin(\tau h)}{\tau}\Big) + (z-z_l)\\ -\frac{\sigma_{l+1,2}}{\tau\sin(\tau h)}\Big((z-z_l) - \frac{1}{\tau}(\sin(\tau(z-z_{l+1})) - \sin(\tau(z_l-z_{l+1})))\Big)\\ -\frac{\sigma_{l+2,2}}{\tau\sin(\tau h)}\Big((z-z_l) - \frac{\sin(\tau(z-z_l))}{\tau}\Big)\Big], & z\in[z_l,z_{l+1}),\\ \\ \frac{\sigma_{l+2,2}\sigma_{l+1,3}}{\tau\sin(\tau h)}\Big[(z_{l+2}-z) + \frac{\sin(\tau(z-z_{l+2}))}{\tau}\Big], & z\in[z_{l+1},z_{l+2}),\\ \\ 0, \quad \text{otherwise}, \end{cases}$$

where

$$\sigma_{l,2} = \frac{\tau\sin(\tau h)}{2\big[\sin^2\big(\frac{\tau(z_{l-1}-z_{l-2})}{2}\big) + \sin^2\big(\frac{\tau(z_l-z_{l-1})}{2}\big)\big]}$$

and

$$\begin{aligned}\sigma_{l,3} = 1/\Big[&\frac{2\sigma_{l,2}}{\tau\sin(\tau h)}\Big[\frac{z_{l-1}-z_{l-2}}{2} - \frac{\sin(\tau(z_{l-1}-z_{l-2}))}{2\tau}\Big] + (z_l - z_{l-1})\\ &- \frac{2\sigma_{l,2}}{\tau\sin(\tau h)}\Big[\frac{z_l-z_{l-1}}{2} - \frac{\sin(\tau(z_l-z_{l-1}))}{2\tau}\Big]\\ &- \frac{2\sigma_{l+1,2}}{\tau\sin(\tau h)}\Big[\frac{z_l-z_{l-1}}{2} - \frac{\sin(\tau(z_l-z_{l+1}))}{2\tau}\Big]\\ &+ \frac{2\sigma_{l+1,2}}{\tau\sin(\tau h)}\Big[\frac{z_{l+1}-z_l}{2} - \frac{\sin(\tau(z_l-z_{l+1}))}{2\tau}\Big]\Big].\end{aligned}$$

Over the space domain $[a, b]$, the set of UAT tension B-spline functions $\{\mathbb{B}_{-1,4},\mathbb{B}_{0,4},...,\mathbb{B}_{N+1,4}\}$ forms a basis and Table 1 lists the values of function $\mathbb{B}_{l,4}$ and its first order derivative at grid

Table 2 Comparison of absolute errors for Example 1 by taking $N = 60$ and $k = 0.05$

z	t	VIM (Yıldırım 2010)	ADM (Ganji et al. 2008)	QuBSMs (Wasim et al. 2018)	JWCM (Çelik 2022)	RBF (Jan et al. 2022)	Present
6	0.05	2.005E-3	–	3.349E-4	4.268E-6	8.079E-6	1.92E-08
8		2.807E-4	3.332E-4	4.359E-5	2.221E-7	1.326E-6	7.77E-08
9		–	1.229E-4	1.596E-5	7.350E-7	5.152E-7	4.32E-08
10		3.817E-5	4.521E-5	5.860E-6	6.831E-7	1.870E-7	2.13E-08
12		5.170E-6	–	7.900E-7	5.413E-7	5.154E-8	4.29E-09
6	0.10	4.226E-3	–	8.847E-4	1.097E-5	1.797E-5	4.29E-08
8		5.911E-4	7.108E-4	1.159E-4	1.105E-6	2.926E-6	1.63E-07
9		–	2.623E-4	4.248E-5	2.682E-6	1.136E-6	9.09E-08
10		8.035E-5	9.659E-5	1.560E-5	2.509E-6	4.139E-7	4.48E-08
12		1.088E-5	–	2.100E-6	1.816E-6	1.007E-7	9.02E-09
6	0.15	–	-	–	–	–	7.35E-08
8		–	1.139E-3	2.238E-4	2.404E-6	4.844E-6	2.58E-07
9		–	4.203E-4	8.208E-5	5.416E-6	1.878E-6	1.43E-07
10		–	1.547E-4	3.014E-5	5.086E-6	6.866E-7	7.08E-08
12		—	–	–	1.42E-08		
6	0.20	–	–	–	–	–	1.14E-07
8		–	1.624E-3	3.765E-4	3.771E-6	7.125E-6	3.62E-07
9		–	5.992E-4	1.381E-4	8.324E-6	2.760E-6	2.01E-07
10		–	2.207E-4	5.073E-5	7.845E-6	1.012E-6	9.94E-08
12		–	–	–	–	–	2.00E-08

Table 3 Absolute errors using the proposed method for Example 1 by taking different values of N and τ at $t = 0.1$

N	z	$\tau = 1.0$	$\tau = 1.25$	$\tau = 1.50$	$\tau = 1.75$	$\tau = 2.0$
20	6	6.16E-05	1.49E-05	6.09E-05	2.03E-04	6.38E-04
	9	9.46E-06	9.40E-06	7.77E-06	1.19E-06	4.82E-05
	12	7.25E-07	7.58E-07	5.08E-07	1.60E-06	3.75E-05
30	6	3.98E-06	2.39E-06	1.06E-05	2.10E-05	3.42E-05
	9	1.60E-06	1.82E-06	2.09E-06	2.43E-06	2.84E-06
	12	1.50E-07	1.86E-07	2.33E-07	2.90E-07	3.59E-07
60	6	4.29E-08	3.04E-07	7.33E-07	1.25E-06	1.85E-06
	9	9.09E-08	1.05E-07	1.23E-07	1.45E-07	1.70E-07
	12	9.02E-09	1.13E-08	1.42E-08	1.76E-08	2.16E-08

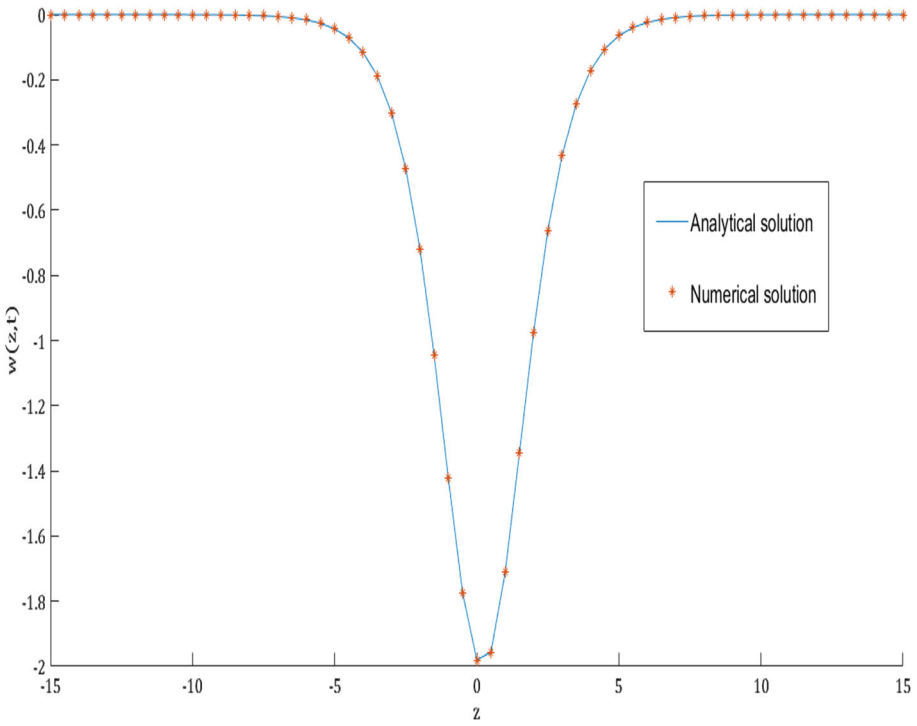

Fig. 1 Analytical and numerical solutions of Example 1 at $t = 0.1$

points where the d_i's for $i = 1, 2, 3, 4$ are defined as

$$d_1 = \frac{1}{4h[\sin^2(\frac{\tau h}{2})]}\Big[h - \frac{\sin(\tau h)}{\tau}\Big],$$

$$d_2 = 1 - \frac{1}{2h[\sin^2(\frac{\tau h}{2})]}\Big[h - \frac{\sin(\tau h)}{\tau}\Big],$$

$$d_3 = \frac{1}{2h},\ d_4 = -\frac{1}{2h}.$$

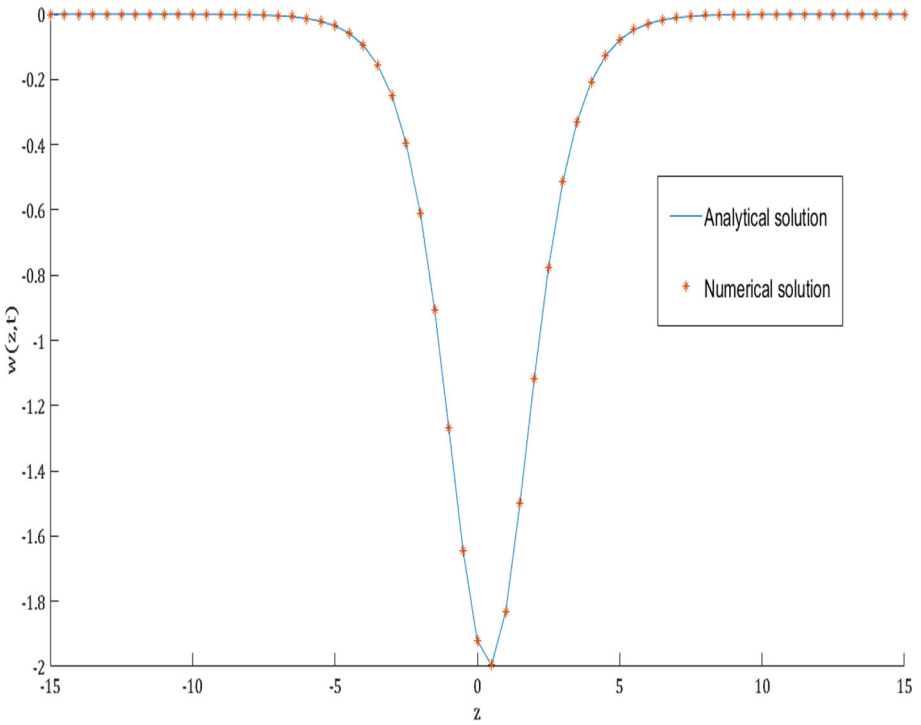

Fig. 2 Analytical and numerical solutions of Example 1 at $t = 0.2$

Further, to make the matrix diagonally dominant, $\mathbb{B}$-basis functions are modified as (Tamsir et al. 2018)

$$\mathbb{MB}_l(z) = \begin{cases} \mathbb{B}_{0,4}(z) + 2\mathbb{B}_{-1,4}(z), & l = 0 \\ \mathbb{B}_{1,4}(z) - \mathbb{B}_{-1,4}(z), & l = 1 \\ \mathbb{B}_{l,4}(z), & l = 2, 3, ..., N-2 \\ \mathbb{B}_{N-1,4}(z) - \mathbb{B}_{N+1,4}(z), & l = N-1 \\ \mathbb{B}_{N,4}(z) + 2\mathbb{B}_{N+1,4}(z), & l = N \end{cases} \tag{9}$$

in the domain $[a, b]$. The approximate values of the derivatives of w with respect to z at z_i can now be estimated by making the assumption that the function $w(z)$ is sufficiently smooth throughout its entire solution domain. We have that the p-order derivatives may be approximated by

$$w_z^{(p)}(z_i) = \frac{\partial^p u(z_i, t)}{\partial z^p} = \sum_{j=0}^{N} \alpha_{ij}^{(p)} w(z_j, t),\ i = 0, 1, 2, ..., N;\ \ p = 1, 2, 3, \tag{10}$$

where $\alpha_{ij}^{(p)}$ are the corresponding weighting coefficients.

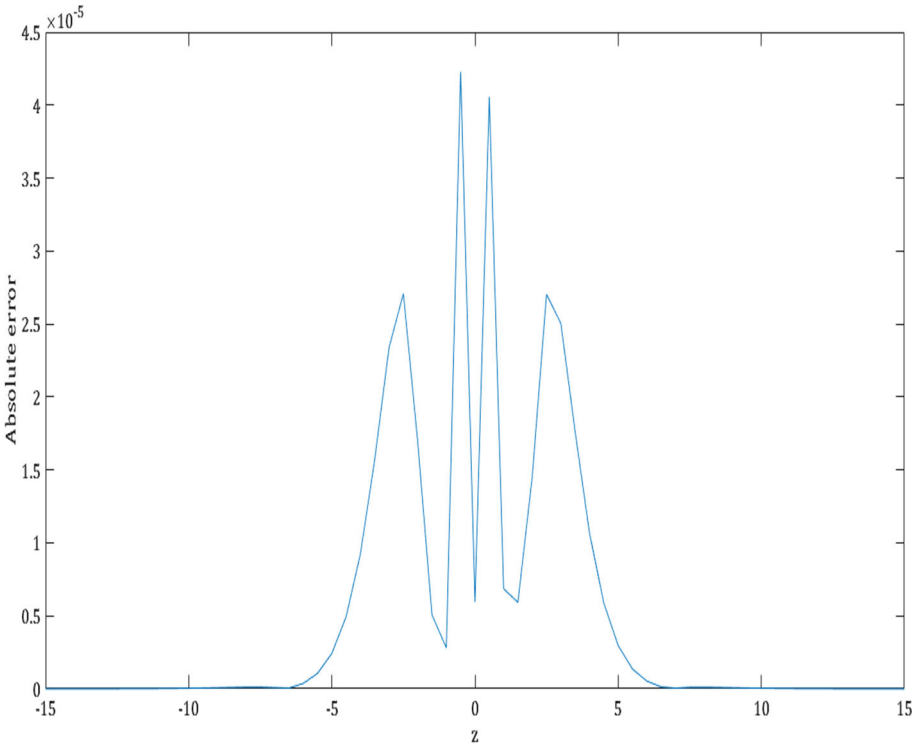

Fig. 3 Absolute errors for Example 1 at $t = 0.05$

2.2 Determination of the weighting coefficients

In order to find the weighting coefficients for the first order derivative, consider $p = 1$ in eq. (10) and trial functions to be $\mathbb{MB}_l$ in (9) such that one gets

$$\mathbb{MB}_l'(z_i) = \sum_{j=0}^{N} \alpha_{ij}^{(1)} \mathbb{MB}_l(z_j), \text{ for } l = 0, 1, ..., N; i = 0, 1, ..., N. \tag{11}$$

As a consequence, we get the following system of equations

$$KA = S,$$

where the j-th column of A is

$$A_j = [\alpha_{j-1,0}^{(1)}, \alpha_{j-1,1}^{(1)}, ..., \alpha_{j-1,N}^{(1)}]^T, \ j = 1, 2, \ldots, N+1,$$

$$K = \begin{bmatrix} d_2 + 2d_1 & d_1 & 0 & & & \\ 0 & d_2 & d_1 & 0 & & \\ 0 & d_1 & d_2 & d_1 & & \\ & & \cdot\cdot & \cdot\cdot & \cdot\cdot & \\ & & \cdot\cdot & \cdot\cdot & \cdot\cdot & \\ & & & d_1 & d_2 & 0 \\ & & & 0 & d_1 & d_2 + 2d_1 \end{bmatrix},$$

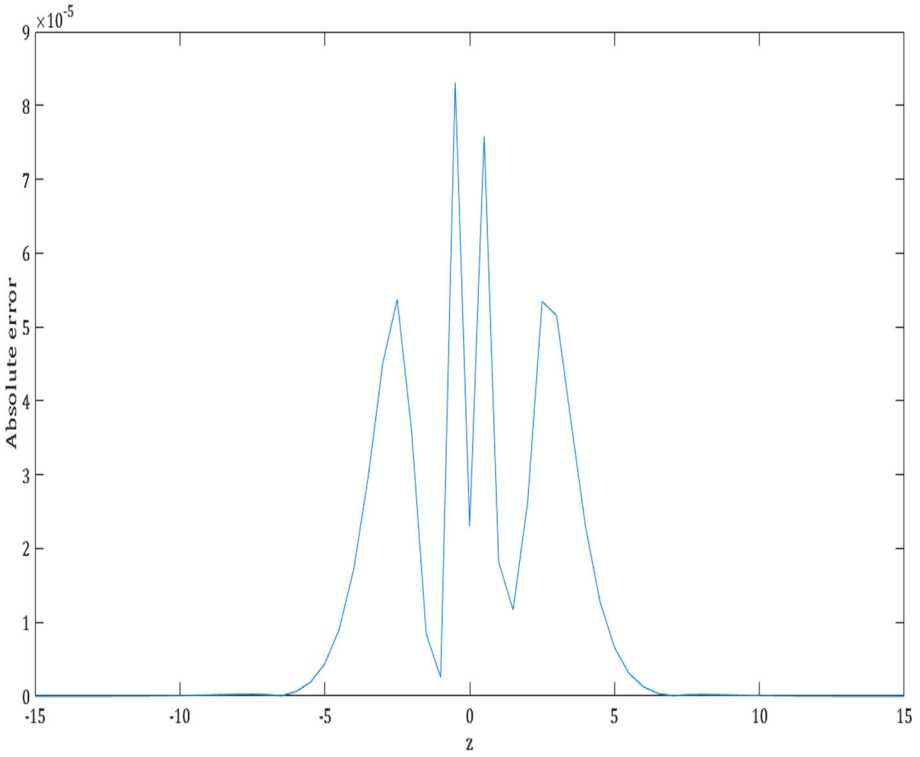

Fig. 4 Absolute errors for Example 1 at $t = 0.1$

and the columns of S are

$$\begin{gathered} S_1 = [2d_4, d_3 - d_4, 0, ..., 0]^T, \\ S_2 = [d_3, 0, d_4, 0, ..., 0]^T, \\ S_3 = [0, d_3, 0, d_4, 0, ..., 0]^T, \\ \cdots \\ \cdots \\ S_{N+1} = [0, ..., 0, d_4 - d_3, 2d_3]^T. \end{gathered}$$

Solving the above system of equations, we obtain the weighting coefficients $\{\alpha_{i0}^{(1)}, \alpha_{i1}^{(1)}, ..., \alpha_{iN}^{(1)}\}$ for $i = 0, 1, 2, ..., N$. Further, using Shu's recurrence formula (Shu 2012), the weighting coefficients for the second order derivative can be evaluated as

$$\alpha_{ij}^{(2)} = 2\Big[\alpha_{ij}^{(1)}\alpha_{ii}^{(1)} - \frac{\alpha_{ij}^{(1)}}{z_i - z_j}\Big], \text{ for } i \neq j \tag{12}$$

and

$$\alpha_{ii}^{(2)} = -\sum_{j=0, i\neq j}^{N} \alpha_{ij}^{(2)}.$$

By matrix multiplication approach, we have

$$[\alpha_{ij}^{(3)}] = [\alpha_{ij}^{(1)}][\alpha_{ij}^{(2)}],$$

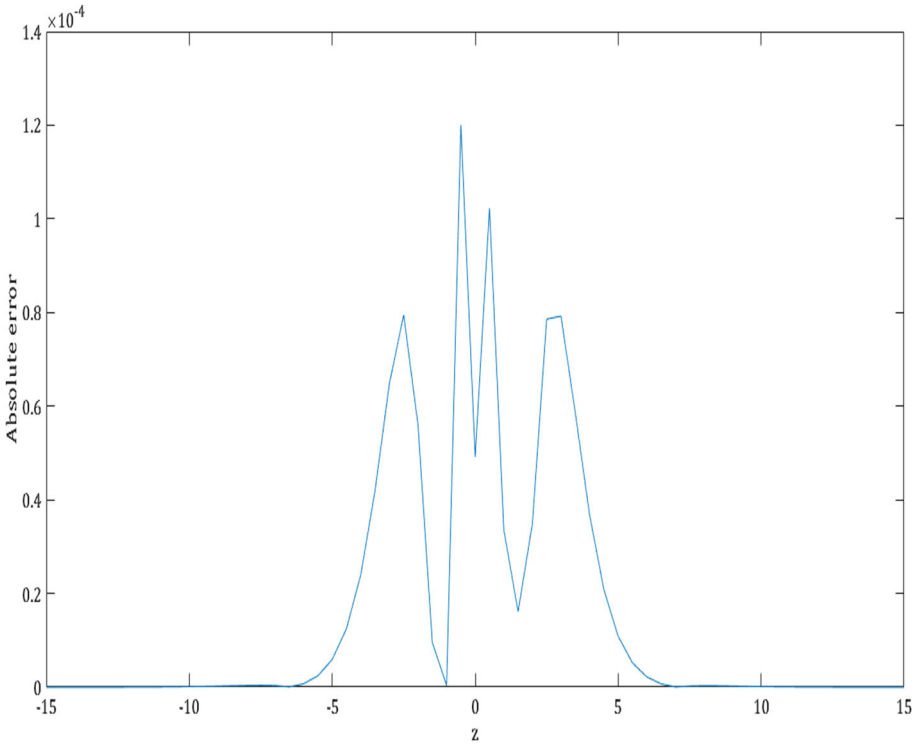

Fig. 5 Absolute errors for Example 1 at $t = 0.15$

where $[\alpha_{ij}^{(1)}]$, $[\alpha_{ij}^{(2)}]$ and $[\alpha_{ij}^{(3)}]$ are weighting coefficients for the approximations of first, second and third order derivatives, respectively. Hence, approximations to the partial derivatives are attained by using these weighting coefficients.

3 Formulation of the proposed discetization

The approximations of the derivatives obtained in (10) are substituted in the CH eq. (2) and its modified form (3) to obtain the following

$$\begin{aligned}\frac{\mathrm{d}w_i}{\mathrm{d}t} - \sum_{j=0}^{N} \alpha_{ij}^{(2)} \frac{\mathrm{d}w_j}{\mathrm{d}t} = &-3w_i \left(\sum_{j=0}^{N} \alpha_{ij}^{(1)} w_j \right) \\ &+ 2 \left(\sum_{j=0}^{N} \alpha_{ij}^{(1)} w_j \sum_{j=0}^{N} \alpha_{ij}^{(2)} w_j \right) + w_i \sum_{j=0}^{N} \alpha_{ij}^{(3)} w_j,\end{aligned} \tag{13}$$

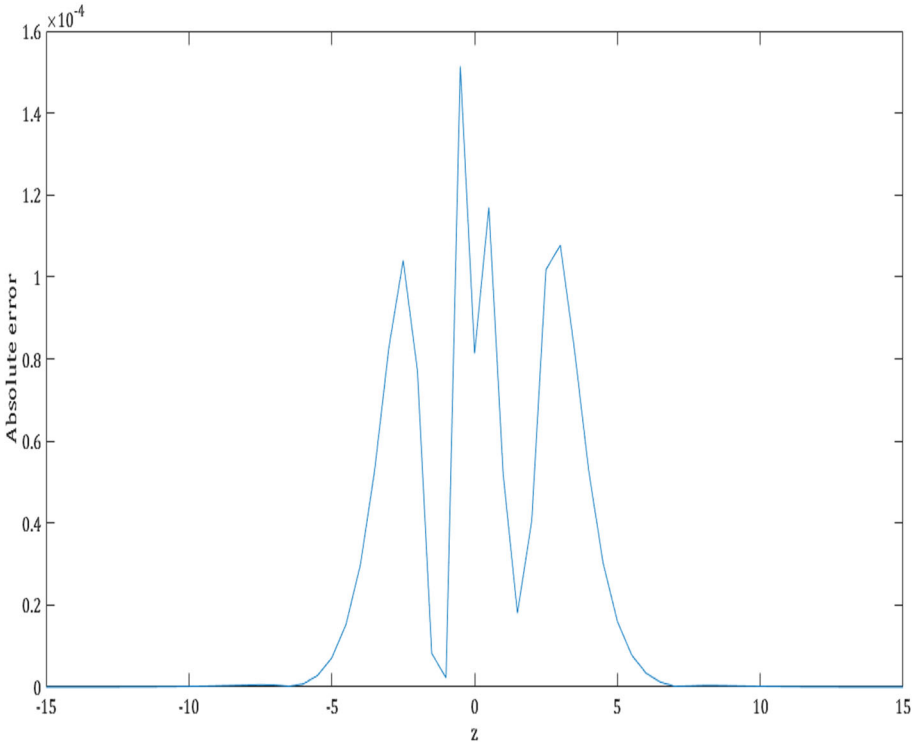

Fig. 6 Absolute errors for Example 1 at $t = 0.2$

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} - \sum_{j=0}^{N} \alpha_{ij}^{(2)} \frac{\mathrm{d}w_j}{\mathrm{d}t} = -3w_i^2 \left(\sum_{j=0}^{N} \alpha_{ij}^{(1)} w_j \right) + 2 \left(\sum_{j=0}^{N} \alpha_{ij}^{(1)} w_j \sum_{j=0}^{N} \alpha_{ij}^{(2)} w_j \right) + w_i \sum_{j=0}^{N} \alpha_{ij}^{(3)} w_j, \tag{14}$$

for $i = 1, 2, ..., N-1$. Similarly, substitutions are done for the DP eq. (5) and its modified form (6) which results, respectively, in the following approximations

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} - \sum_{j=0}^{N} \alpha_{ij}^{(2)} \frac{\mathrm{d}w_j}{\mathrm{d}t} = -4w_i \left(\sum_{j=0}^{N} \alpha_{ij}^{(1)} w_j \right) + 3 \left(\sum_{j=0}^{N} \alpha_{ij}^{(1)} w_j \sum_{j=0}^{N} \alpha_{ij}^{(2)} w_j \right) + w_i \sum_{j=0}^{N} \alpha_{ij}^{(3)} w_j, \tag{15}$$

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} - \sum_{j=0}^{N} \alpha_{ij}^{(2)} \frac{\mathrm{d}w_j}{\mathrm{d}t} = -4w_i^2 \left(\sum_{j=0}^{N} \alpha_{ij}^{(1)} w_j \right) + 3 \left(\sum_{j=0}^{N} \alpha_{ij}^{(1)} w_j \sum_{j=0}^{N} \alpha_{ij}^{(2)} w_j \right) + w_i \sum_{j=0}^{N} \alpha_{ij}^{(3)} w_j, \tag{16}$$

Table 4 Convergence rate with respect to z in terms of L_∞ for $t = 0.2$

N	L_∞	Convergence rate
20	1.7193e−02	–
30	3.2520e−03	4.1070
40	9.2640e−04	4.3650
50	3.1403e−04	4.8481
60	1.5141e−04	4.0012

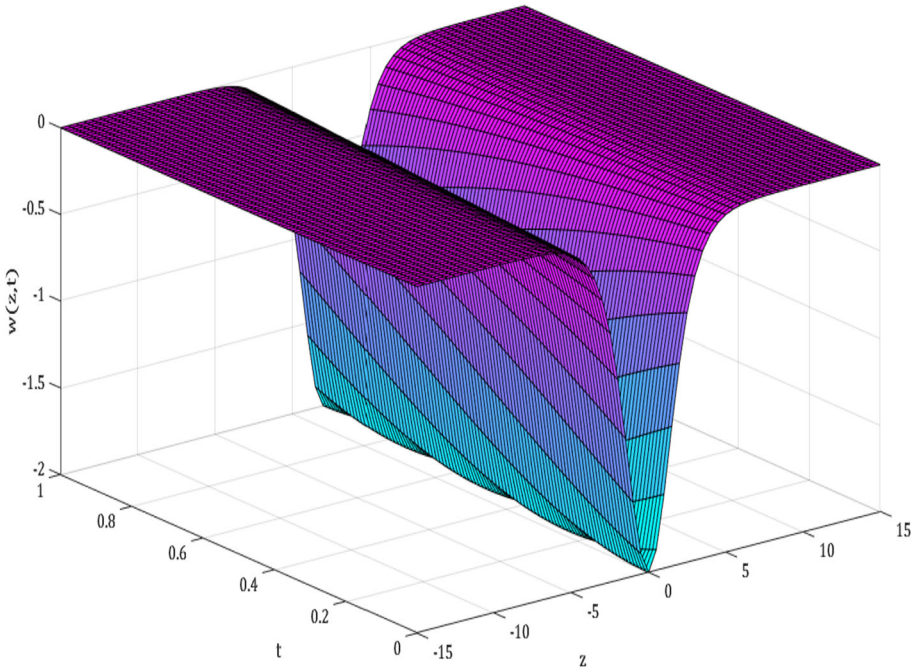

Fig. 7 3D plot of numerical solution of Example 1 for $t \leq 1$ taking $k = 0.01$, $N = 60$

where $i = 1, 2, \ldots, N-1$. The corresponding boundary conditions considered along with the above equations are given below

$$w_t(z_0) = f_1'(t) \text{ and } w_t(z_N) = f_2'(t).$$

This gives us a system of $N+1$ ordinary differential equations of first order with initial condition

$$w(z, 0) = w_0(z), z \in [a, b]. \tag{17}$$

Now, this system will be solved using an optimized hybrid block method which is described in the following section.

Table 5 Comparison of absolute errors for Example 2 by taking $N = 40$ and $k = 0.5$ for $t = 0.5$

z	(Jan et al. 2022) $N = 400$	Present $N = 40$
–10	3.3855E-8	8.4703E-22
–8	1.4744E-8	7.4016E-09
–6	8.5791E-8	5.5703E-08
–4	8.1819E-7	4.1163E-07
–2	6.4433E-6	3.0415E-06
0	4.9875E-5	4.9617E-05
2	6.4771E-6	3.0459E-06
4	8.2273E-7	4.1229E-07
6	8.6376E-8	5.5791E-08
8	1.4703E-8	7.4133E-09
10	3.3906E-8	3.7058E-22

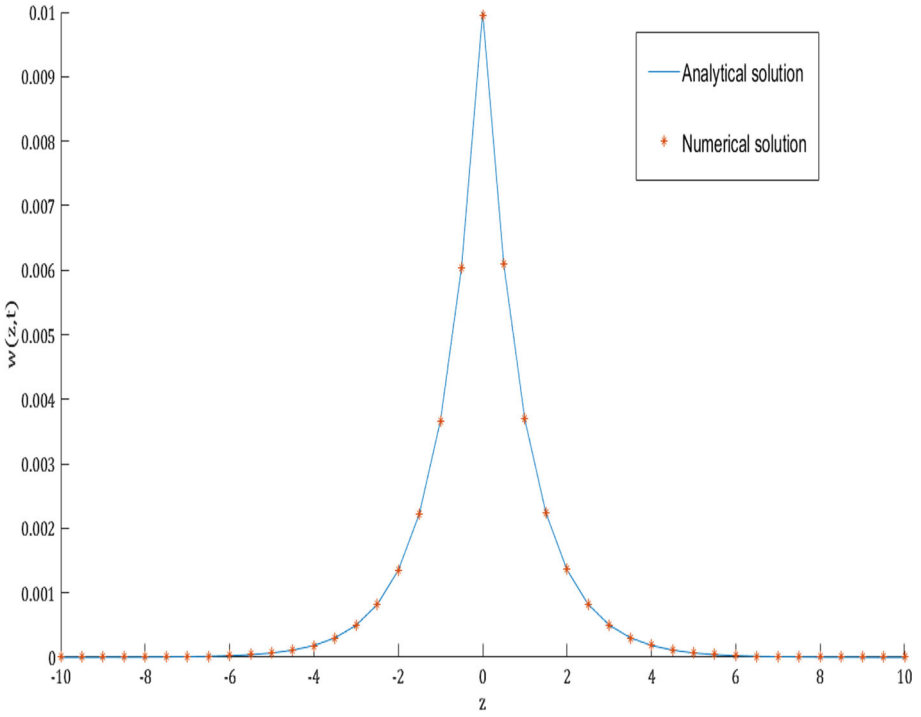

Fig. 8 Analytical and numerical solutions of Example 2 at $t = 0.5$

4 Hybrid block method

With the evolution of numerical methods for spatial variables, a better independent time-stepping algorithm that integrates the first order initial value problems (IVPs) while overcoming the accuracy-grid size trade-off are required. Such issues can be resolved via hybrid block approaches. Considering a first order initial value problem (IVP) of the form $\{y' = f(t, y); y(0) = y_0\}$, an optimized one-step hybrid block method has been proposed

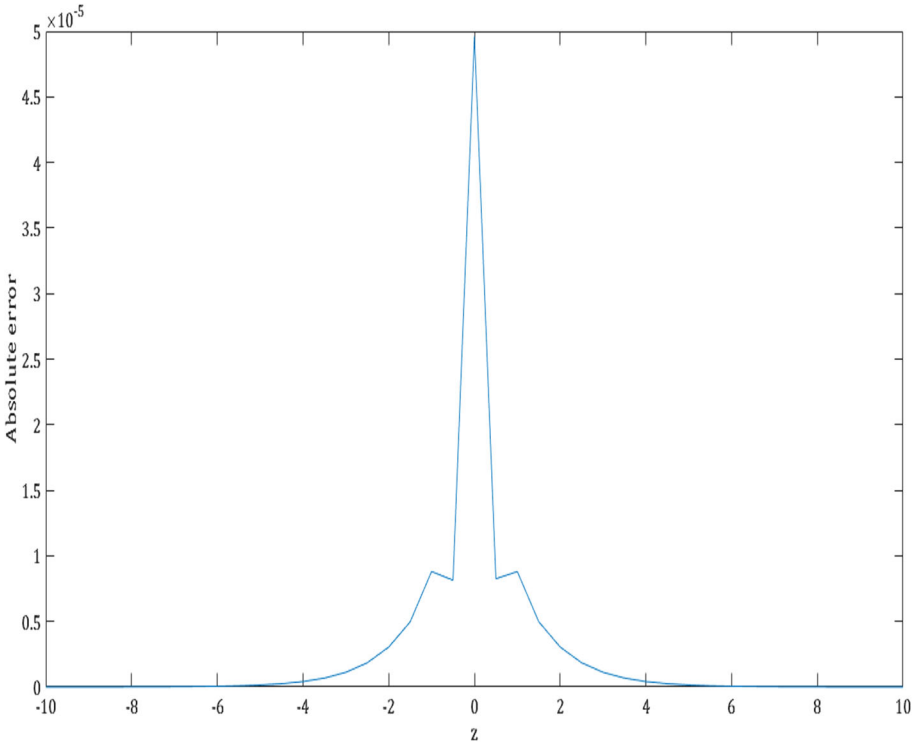

Fig. 9 Absolute errors for Example 2 at $t = 0.5$

in Kaur and Kanwar (2022) using the concepts of interpolation and collocation. This method achieves attributes such as zero-stability, convergence, A-stability, and consistency that have at least fourth algebraic order. The so-called hybrid block approach is made up of the following equations

$$kf_{i+1} = -7y_i + 5\sqrt{5}y_{i+r} - 5\sqrt{5}y_{i+s} + 7y_{i+1} - kf_i, \tag{18a}$$

$$kf_{i+r} = \frac{(-17-\sqrt{5})}{2\sqrt{5}}y_i + \frac{(5+\sqrt{5})}{2}y_{i+r} + \frac{(-5+3\sqrt{5})}{2}y_{i+s} + \frac{(-3+\sqrt{5})}{2\sqrt{5}}y_{i+1} - \frac{k}{\sqrt{5}}f_i, \tag{18b}$$

$$kf_{i+s} = \frac{(17-\sqrt{5})}{2\sqrt{5}}y_i + \frac{(-5-3\sqrt{5})}{2}y_{i+r} + \frac{(5-\sqrt{5})}{2}y_{i+s} + \frac{(3+\sqrt{5})}{2\sqrt{5}}y_{i+1} + \frac{k}{\sqrt{5}}f_i, \tag{18c}$$

where k is the fixed step-size, $r = \frac{1}{2} - \frac{\sqrt{5}}{10}$, $s = \frac{1}{2} + \frac{\sqrt{5}}{10}$ and $f_i = f(t_i, y_i)$. The system of equations of first order IVPs obtained in the previous section, is solved by the block method (18).

Table 6 Comparison of absolute errors for Example 3 by taking $N = 60$ and $k = 0.05$

z	t	VIM (Yıldırım 2010)	ADM (Ganji et al. 2008)	QuBSMs (Wasim et al. 2018)	JWCM (Çelik 2022)	RBF (Jan et al. 2022)	Present
6	0.05	2.005E-3	–	4.490E-4	5.814E-6	9.105E-6	9.41E-10
8		2.807E-4	3.332E-4	6.312E-5	2.392E-7	1.511E-6	8.92E-08
9		–	1.229E-4	2.332E-5	9.460E-7	5.887E-7	5.02E-08
10		3.817E-5	4.521E-5	8.590E-6	8.990E-7	2.138E-7	2.49E-08
12		5.169E-6	–	1.160E-6	7.063E-7	6.149E-8	5.04E-09
6	0.10	4.226E-3	–	9.037E-4	1.462E-5	2.045E-5	1.97E-08
8		5.911E-4	7.108E-4	1.276E-4	1.273E-6	3.371E-6	1.92E-07
9		–	2.623E-4	4.720E-5	3.378E-6	1.313E-6	1.08E-07
10		8.036E-5	9.659E-5	1.740E-5	3.209E-6	4.788E-7	5.33E-08
12		1.088E-5	–	2.350E-6	2.325E-6	1.204E-7	1.08E-08
6	0.15	–	–	–	–	–	6.30E-08
8		–	1.139E-3	1.932E-4	2.608E-6	5.644E-6	3.09E-07
9		–	4.203E-4	1.461E-5	6.438E-6	2.196E-6	1.73E-07
10		–	1.547E-4	2.635E-5	6.138E-6	8.045E-7	8.57E-08
12		–	–	–	–	–	1.73E-08
6	0.20	–	–	–	–	–	1.39E-07
8		–	1.624E-3	2.585E-4	3.632E-6	8.406E-6	4.45E-07
9		–	5.992E-4	9.568E-5	9.038E-6	3.269E-6	2.48E-07
10		–	2.207E-4	3.529E-5	8.672E-6	1.202E-6	1.23E-07
12		–	–	–	–	–	2.47E-08

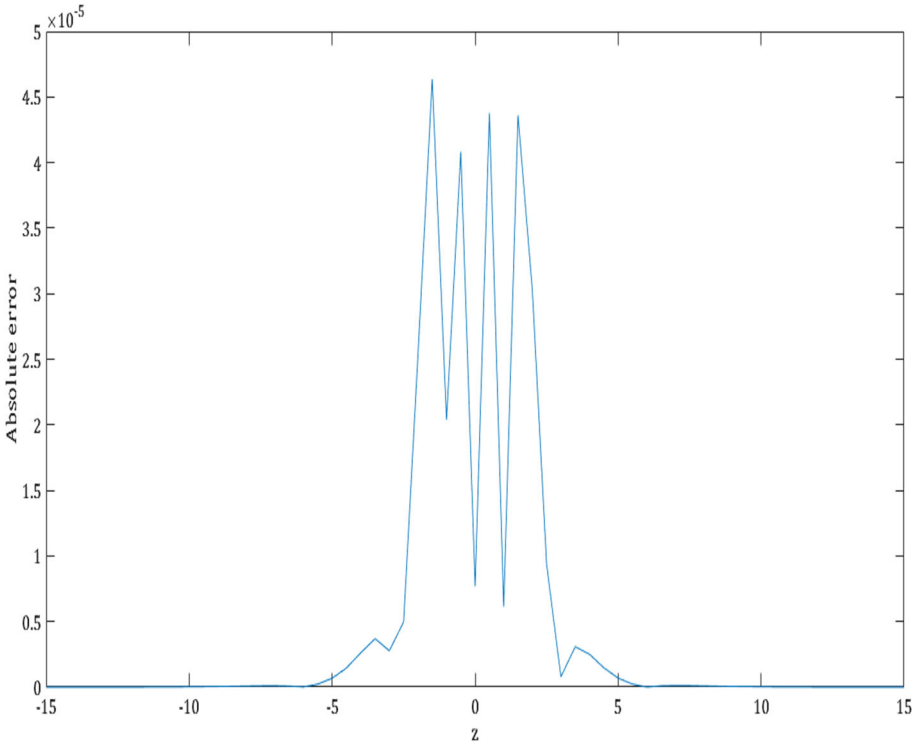

Fig. 10 Absolute errors for Example 3 at $t = 0.05$

5 Numerical results

In order to prove the efficiency of the proposed method, this section comprises of numerical experiments and errors are evaluated using following formulas:

$$L_\infty = \max_i |w_i^{exact} - w_i^{num}|,$$

$$L_2 = \sqrt{h \sum_{i=1}^{N} |w_i^{exact} - w_i^{num}|^2},$$

where w_i^{exact} and w_i^{num} refer to the exact and approximate solutions evaluated at (z_i, t), where t will be specified on each case. The formula of the convergence rate with respect to z is

$$\frac{\log_{10}[Error(h_i)/Error(h_{i+1})]}{\log_{10}[h_i/h_{i+1}]},$$

where h_i is the spatial step size. Numerical results are obtained using MATLAB R2017a on i5-4210U CPU with 64-bit operating system and 4GB RAM.

Example 1: We first investigate the Camassa-Holm eq. (3) with initial condition

$$w(z, 0) = -2sech^2\left(\frac{z}{2}\right).$$

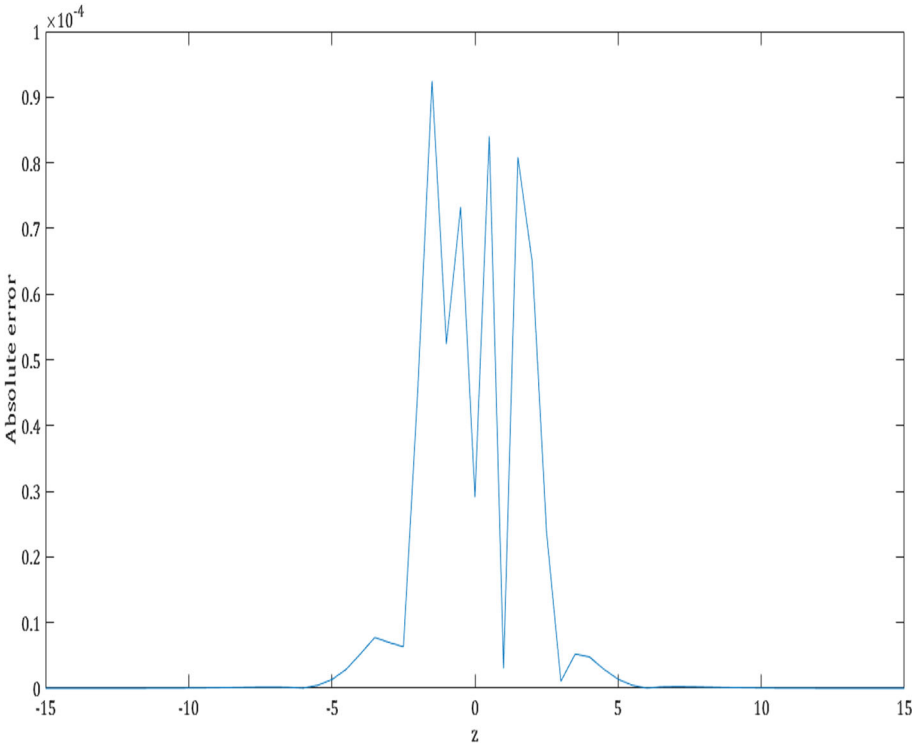

Fig. 11 Absolute errors for Example 3 at $t = 0.1$

The space domain is [-15,15] and the boundary conditions can be easily derived from the analytical solution which is given as

$$w(z,t) = -2sech^2\Big(\frac{z}{2} - t\Big).$$

To analyze the effectiveness of the derived method, the results are compared in Table 2 to those provided by other numerical methods in Yıldırım (2010); Ganji et al. (2008); Wasim et al. (2018); Çelik (2022); Jan et al. (2022). The results tabulated in Table 2 are obtained using $k = 0.05$, $N = 60$ and $\tau = 1.75$ for different values of t. Comparisons show that the results provided by our method are remarkably better than others. The L_∞ errors at $t = 0.1$ are tabulated in Table 3 for different values of N and τ which depicts that value of τ closer to 1 gives more accurate solutions for any value of N. Figures 1 and 2 display analytical and numerical solutions at $t = 0.1$ and $t = 0.2$. The proficiency of hybrid block method can be seen through absolute errors obtained in single time step for $t = 0.05, 0.1, 0.15, 0.20$ that are displayed in Figs. 3, 4, 5 and 6, respectively. Convergence rate with respect to z in terms of L_∞ error is tabulated in Table 4 for $t = 0.2$ by taking $k = 0.2$. A 3D-plot for numerical solution for $t \in [0, 1]$ is shown in Fig. 7.

Example 2: Consider the CH eq. (2), for which ϕ is a homogeneous polynomial in eq. (1) for $z \in [-10, 10]$ and whose exact solution is

$$w(z,t) = c\exp^{-|z-ct|}.$$

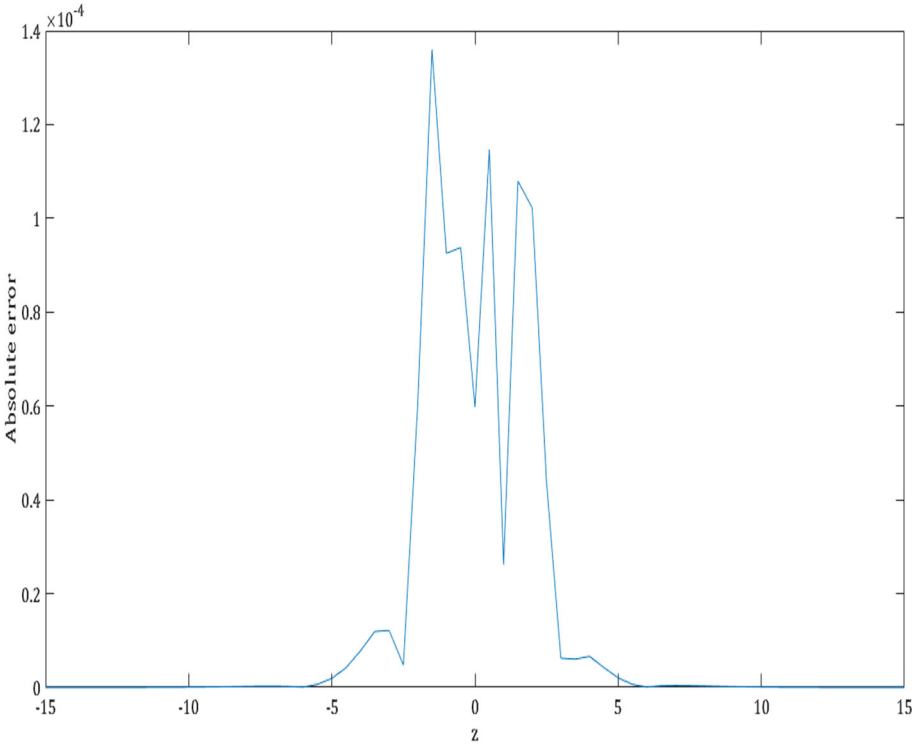

Fig. 12 Absolute errors for Example 3 at $t = 0.15$

The comparison of the results obtained by Jan et al. (2022) and that of the present method for $t = 0.5$ and $c = 0.01$ are given in Table 5. It is evident that the number of nodes for the derived method is less and the obtained results are more accurate even using just a single time step. Figure 8 presents the analytical and the numerical solutions. The obtained absolute errors are plotted in Fig. 9 for $t = 0.5$.

Example 3: The modified Degasperis-Procesi eq. (6) is considered in the space domain [-15,15] and has an analytical solution given by

$$w(z,t) = -\frac{15}{8} sech^2\Big(\frac{z}{2} - \frac{-5t}{4}\Big).$$

The initial and boundary conditions are obtained from the exact solution. Many numerical methods have been applied to this equation and are compared with the results obtained by the presented method. We choose $N = 60$, $k = 0.05$ and $\tau = 1.5$ to evaluate the absolute error for different values of t and results are given in Table 6. We can see that the obtained results with the proposed method are the best among all the numerical methods in Yıldırım (2010); Ganji et al. (2008); Wasim et al. (2018); Çelik (2022); Jan et al. (2022). Figures 10, 11, 12 and 13 display absolute errors for $t = 0.05, 0.1, 0.15$ and 0.20, respectively. Table 7 shows the absolute error for $t = 0.01$ by considering different values of N along with different values of τ. The accuracy shifts with the value of τ from 1.5 to 1 with the increase of grid-size N. The rate of convergence with respect to z is given in Table 8 at time $t = 0.2$ considering just a single time step. Figure 14 represents a 3D-plot of the numerical solution for $t \in [0, 1]$.

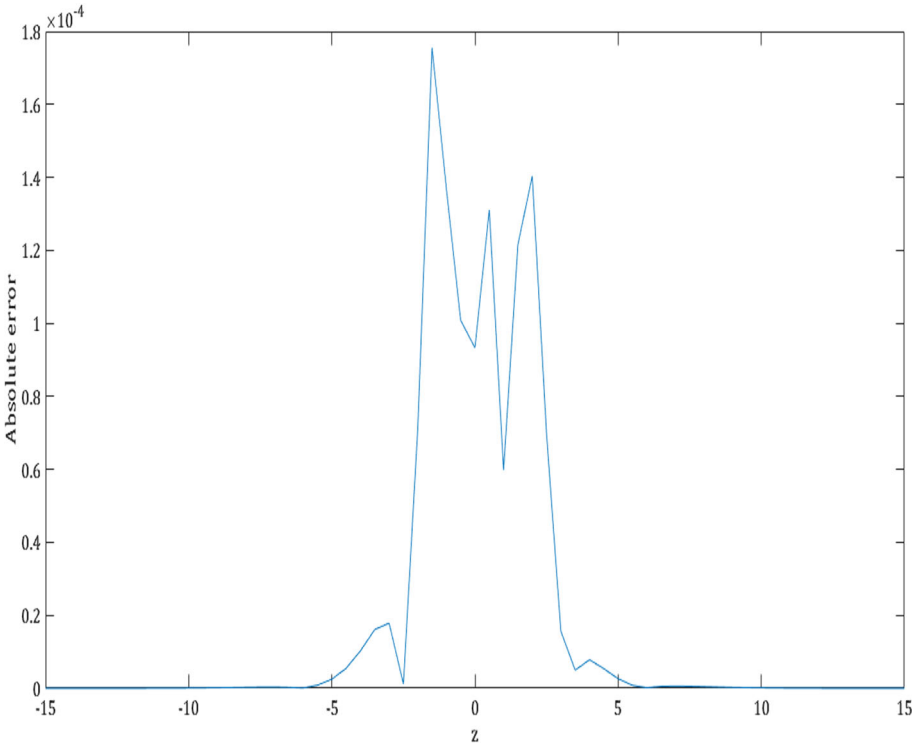

Fig. 13 Absolute errors for Example 3 at $t = 0.2$

Table 7 Absolute errors using the proposed method for Example 3 by taking different values of N and τ at $t = 0.1$

N	z	$\tau = 1.0$	$\tau = 1.25$	$\tau = 1.50$	$\tau = 1.75$	$\tau = 2.0$
20	6	7.57E-05	1.88E-05	7.54E-05	2.60E-04	8.87E-04
	9	1.14E-05	1.13E-05	9.21E-06	2.84E-06	7.45E-05
	12	8.76E-07	9.15E-07	5.99E-07	2.15E-06	5.44E-05
30	6	4.39E-06	3.32E-06	1.33E-05	2.60E-05	4.19E-05
	9	1.90E-06	2.15E-06	2.47E-06	2.87E-06	3.36E-06
	12	1.79E-07	2.23E-07	2.78E-07	3.46E-07	4.29E-07
60	6	1.97E-08	4.03E-07	9.27E-07	1.55E-06	2.29E-06
	9	1.08E-07	1.25E-07	1.46E-07	1.71E-07	2.01E-07
	12	1.08E-08	1.35E-08	1.70E-08	2.10E-08	2.58E-08

Example 4: We consider Degasperis-Procesi (DP) eq. (5) which is a subcase of equation (1) with ϕ, a homogeneous polynomial. The initial condition for single peakon traveling wave solution is given by

$$w(z, 0) = c \exp^{-|z|}.$$

Table 8 Convergence rate with respect to z in terms of L_∞ for $t = 0.2$

N	L_∞	Convergence rate
20	1.8568E-02	–
30	3.2823E-03	4.2738
40	8.3656E-04	4.7518
50	2.8600E-04	4.8099
60	1.3349E-04	4.1792

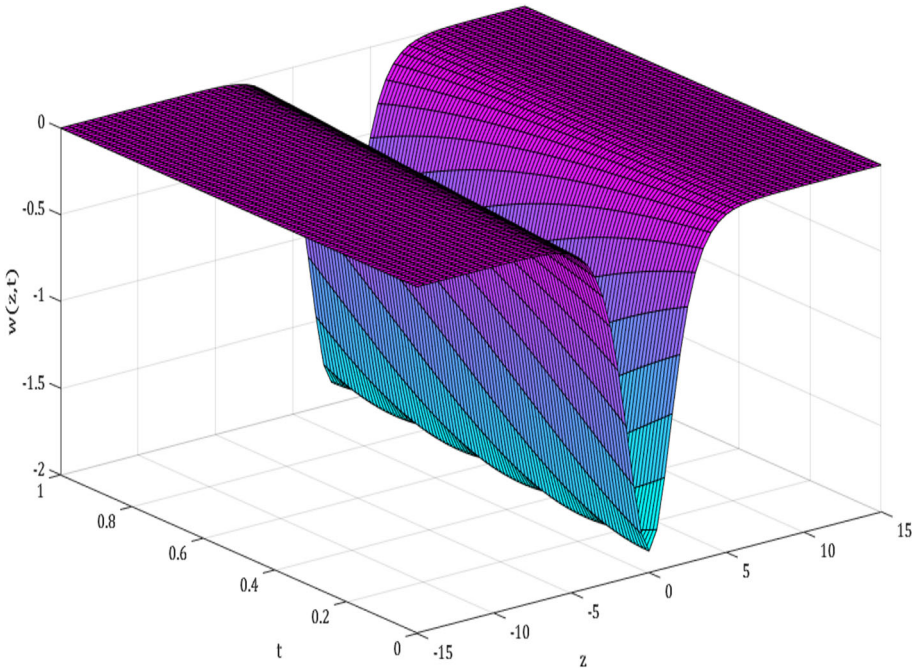

Fig. 14 3D plot of numerical solution of Example 3 for $t \leq 1$ taking $k = 0.01$, $N = 60$

The boundary conditions can be extracted from the exact solution $w(z, t) = c \exp^{-|z-ct|}$. The interval [0,1] is chosen as the space domain and the wave speed $c = 0.25$. The remarkable difference in CPU time in comparison to the methods considered in Shaheen et al. (2022) is recorded in Table 9 for different values of t. The corresponding L_∞ and L_2 errors were obtained taking $N = 10$ and a single time-step.

6 Stability analysis

The choice of the L-stable optimized hybrid block method (18) ensures that the error in approximating the solution of a well-posed system of initial value problems (IVPs) is not magnified. The stability of the proposed algorithm is investigated using matrix stability analysis (Saka et al. 2011; Kaur and Kanwar 2022) which means that the stability depends on the eigenvalues of the coefficient matrix of the system which is obtained after the implementation of the differential quadrature method (DQM) on a given partial differential equation (PDE).

Table 9 L_2 and L_∞ errors with CPU times for Example 4

t	MQ(Shaheen et al. 2022)			IQ(Shaheen et al. 2022)			GA(Shaheen et al. 2022)			Present		
	L_2	L_∞	CPU	L_2	L_∞	CPU	L_2	L_∞	CPU	L_2	L_∞	CPU (in sec)
0.01	8.59E-04	3.02E-03	1.210	1.66E-04	1.51E-04	1.413	1.79E-03	1.15E-03	1.394	5.90E-04	1.03E-03	0.546
0.02	7.07E-04	3.08E-03	1.240	1.67E-03	1.08E-03	1.541	3.63E-03	2.27E-03	1.351	1.21E-03	2.07E-03	0.608
0.04	2.27E-03	3.13E-03	1.222	4.67E-03	3.09E-03	1.819	7.27E-03	4.48E-03	1.433	2.36E-03	4.14E-03	0.878
0.06	5.88E-03	5.88E-03	1.241	7.64E-03	5.07E-03	2.084	1.08E-02	6.66E-03	1.538	3.54E-03	6.20E-03	0.965

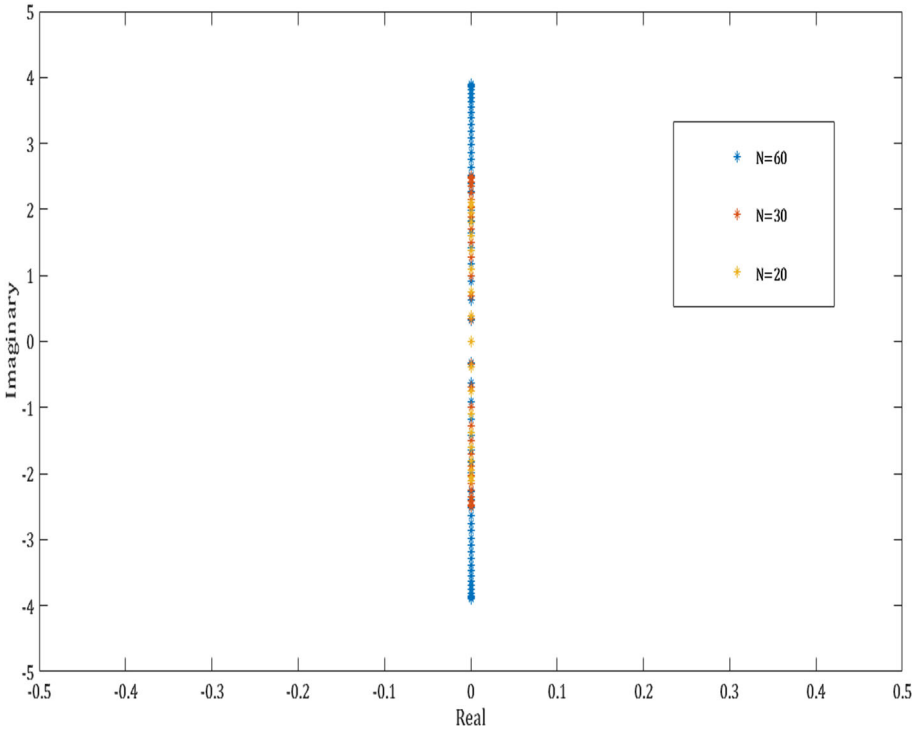

Fig. 15 Eigenvalues for Camassa-Holm equation

After transforming the PDE into the system of ordinary differential equations (ODEs), we get a matrix equation of the form

$$W' = LW + F,$$

where L is the coefficient matrix and F is formed by the non-homogeneous part and the boundary conditions. If real part of eigenvalues of coefficient matrix are either negative or zero then the system of ODEs obtained is stable. Plots of the eigenvalues of L for different number of nodes N are shown in Figs. 15 and 16. As it is evident from these plots, that all the eigenvalues lie on the stability region. Further, the convergence analysis of the hybrid block method (18) has been done in Kaur and Kanwar (2022). Therefore, this results in the unconditional stability of the presented method.

7 Conclusion

The novel algorithm that combines the differential quadrature method (DQM) using fourth order UAT-tension B-splines as the basis to convert a partial differential equation (PDE) into a system of ordinary differential equations (ODEs) and an optimized hybrid block method of order four provides accurate numerical solutions with high efficiency. Some numerical experiments have been presented and analyzed in terms of L_2 and L_∞ errors. This study shows that the proposed method achieves better accuracy considering less number of nodes and saves computational time in comparison to other numerical methods available in the

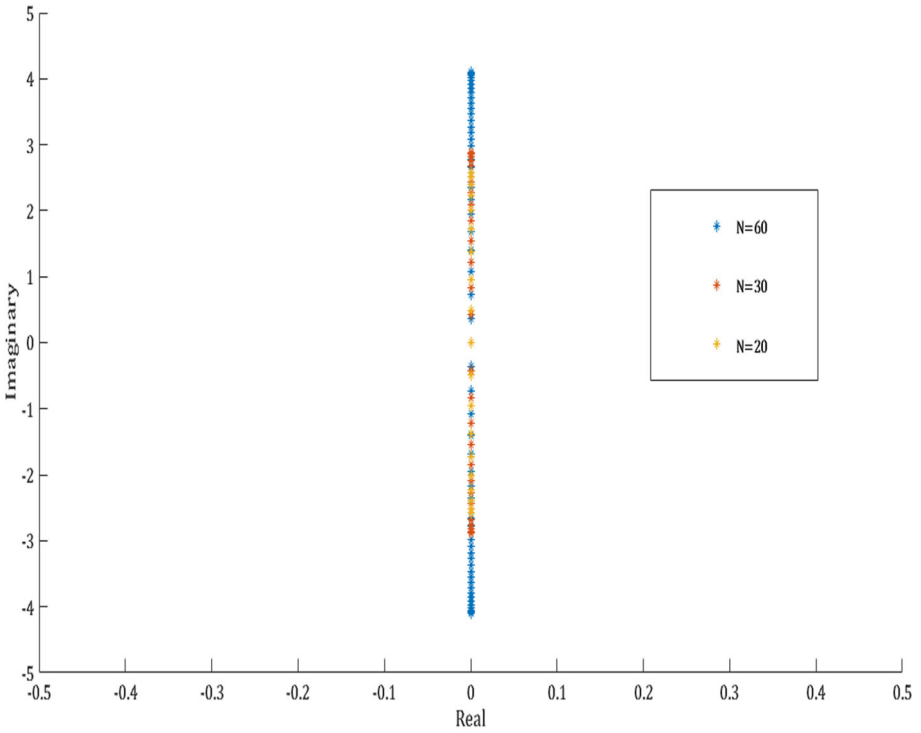

Fig. 16 Eigenvalues for Degasperis-Procesi equation

literature. The stability analysis shows that the combination of two high resolution numerical methods is unconditionally stable for the CH equation as well as for the DP equation. Many other mixed derivative type PDEs can be solved accurately by applying the presented method.

Acknowledgements Anurag Kaur is supported financially by the funding agency, the University Grants Commission (UGC), New Delhi, India under the scheme of UGC-CSIR NET-SRF with reference id 403645.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability The findings of this study are available within the article.

References

Alinia N, Zarebnia M (2018) A new tension B-spline method for third-order self-adjoint singularly perturbed boundary value problems. J Comput Appl Math 342:521–533

Alinia N, Zarebnia M (2019) A numerical algorithm based on a new kind of tension B-spline function for solving Burgers-Huxley equation. Num Algorithms 82:1121–1142

Bellman R, Kashef B, Casti J (1972) Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations. J Comput Phys 10:40–52
Camassa R, Holm DD (1993) An integrable shallow water equation with peaked solitons. Phys Rev Lett 71:1661–1664
Çelik İ (2022) Jacobi wavelet collocation method for the modified Camassa-Holm and Degasperis-Procesi equations. Eng Comput 38:2271–2287
Cheng R-J, Wei Q (2013) Analysis of the generalized Camassa and Holm equation with the improved element-free Galerkin method. Chin Phys B 22:060209
Coclite GM, Karlsen KH (2006) On the well-posedness of the Degasperis-Procesi equation. J Funct Anal 233(1):60–91
Dahlquist G (1956) Convergence and stability in the numerical integration of ordinary differential equations. Mathematica Scandinavica 33-53
Dai H-H (1998) Model equations for nonlinear dispersive waves in a compressible Mooney-Rivlin rod. Acta Mech 127:193–207
Ganji DD, Sadeghi E, Rahmat M (2008) Modified Camassa-Holm and Degasperis-Procesi equations solved by Adomian's decomposition method and comparison with HPM and exact solutions. Acta Appl Math 104:303–311
Hejazi SR, Mohammadi S (2022) Lie symmetry, numerical solution with spectral method and conservation laws of Degasperis-Procesi equation by homotopy and direct methods. International Journal of Modelling and Simulation 1-16
Jan HU, Uddin M, Abdeljawad T, Zamir M (2022) Numerical study of high order nonlinear dispersive PDEs using different RBF approaches. Appl Num Math 182:356–369
Kapoor M, Joshi V (2021) A new technique for numerical solution of 1D and 2D non-linear coupled Burgers' equations by using cubic uniform algebraic trigonometric (UAT) tension B-spline based differential quadrature method. Ain Shams Eng J 12:3947–3965
Kaur A, Kanwar V (2022) Numerical solution of generalized Kuramoto-Sivashinsky equation using cubic trigonometric B-spline based differential quadrature method and one-step optimized hybrid block method. Int J Appl Comput Math 8:1–19
Mainar E, Peña JM, Sánchez-Reyes J (2001) Shape preserving alternatives to the rational Bezier model. Comput Aided Geomet Design 18:37–60
Quan J, Chang C (1989) New insights in solving distributed system equations by the quadrature method-I analysis. Comput Chem Eng 13:779–788
Quan J, Chang C-T (1989) New insights in solving distributed system equations by the quadrature method-II numerical experiments. Comput Chem Eng 13:1017–1024
Ramos H, Kaur A, Kanwar V (2022) Using a cubic B-spline method in conjunction with a one-step optimized hybrid block approach to solve nonlinear partial differential equations. Comput Appl Math 41:1–28
Ramos H, Popescu P (2018) How many k-step linear block methods exist and which of them is the most efficient and simplest one? Appl Math Comput 316:296–309
Saka B, Şahin A, Dağ İ (2011) B-spline collocation algorithms for numerical solution of the RLW equation. Num Methods Partial Differential Equ 27:581–607
Shaheen S, Haq S, Ghafoor A (2022) A meshfree technique for the numerical solutions of nonlinear Fornberg-Whitham and Degasperis-Procesi equations with their modified forms. Comput Appl Math 41:1–22
Shu C (2012) Differential quadrature and its application in engineering. Springer Science & Business Media
Shu C, Richards BE (1992) Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations. Int J Numer Meth Fluids 15:791–798
Singh G, Garg A, Kanwar V, Ramos H (2019) An efficient optimized adaptive step-size hybrid block method for integrating differential systems. Appl Math Comput 362:124567
Singla R, Singh G, Ramos H, Kanwar V (2022) A family of-stable optimized hybrid block methods for integrating stiff differential systems. Mathematical Problems in Engineering, 2022
Tamsir M, Dhiman N, Srivastava VK (2018) Cubic trigonometric B-spline differential quadrature method for numerical treatment of Fisher's reaction-diffusion equations. Alexandria Eng J 57:2019–2026
Wang G et al (2008) Unified and extended form of three types of splines. J Comput Appl Math 216:498–508
Wasim I, Abbas M, Iqbal MK (2018) Numerical solution of modified forms of Camassa-Holm and Degasperis-Procesi equations via quartic B-spline collocation method. Commun Math Appl 9:393–409
Yıldırım A (2010) Variational iteration method for modified Camassa-Holm and Degasperis-Procesi equations. Int J Num Methods Biomed Eng 26:266–272
Zhang B, Li S, Liu Z (2008) Homotopy perturbation method for modified Camassa-Holm and Degasperis-Procesi equations. Phys Lett A 372:1867–1872