International Journal for Numerical Methods in Engineering

WILEY

RESEARCH ARTICLE OPEN ACCESS

An Optimized Block Hybrid Spectral Simple Iteration Methods for Solving Nonlinear Evolution Equations

Salma Ahmedai[1,2] | Precious Sibanda[3] | Sandile Motsa[3,4] | Sicelo Goqo[3] | Osman A. I. Noreldin[5]

[1]Department of Geography, Geo-Informatics and Meteorology, University of Pretoria, Hatfield, South Africa | [2]Faculty of Mathematical Sciences and Statistics, Al Neelain University, Khartoum, Sudan | [3]School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg, South Africa | [4]Department of Mathematics, University of Eswatini, Kwaluseni, Eswatini, South Africa | [5]Department of Mathematical Sciences, University of Zululand, KwaDlangezwa, South Africa

Correspondence: Salma Ahmedai (salma.a.ahmedai@gmail.com)

Received: 12 March 2025 | **Revised:** 17 November 2025 | **Accepted:** 26 November 2025

Keywords: Chebyshev–Gauss–Lobatto points | partial differential equations | shifted Legendre polynomials | simple iteration method

ABSTRACT

This study presents a new optimized block hybrid method and spectral simple iteration method (OBHM-SSIM) for solving nonlinear evolution equations. In this method, we employed a combination of the spectral collocation method in space and the optimized block hybrid method in time, along with a simple iteration scheme to linearize the equations. The performance of OBHM-SSIM is compared with other established numerical methods for various nonlinear evolution equations, including the Stokes' second problem equation, Burgers–Fisher equation, Burgers–Huxley equation, the FitzHugh–Nagumo equation with time-dependent coefficients, and coupled Burgers' equations. Furthermore, the proposed OBHM-SSIM is implemented to solve $(2+1)$-dimensional problems, specifically the nonlinear Burgers' equation and the cubic Klein–Gordon equation, demonstrating its capability to solve nonlinear systems efficiently. The extension to two-dimensional cases further validates the flexibility and accuracy of the OBHM-SSIM method, achieved with a notably reduced computational cost. Unlike conventional spectral methods, the proposed OBHM-SSIM achieves high-order accuracy with fewer grid points by optimizing intra-step points and maintaining A-stability for large time domains. We demonstrate that the OBHM-SSIM method gives highly accurate solutions with fewer grid points. This results in enhanced computational efficiency and reduced complexity, particularly for large time domains of nonlinear evolution equations. The findings of this study offer a new approach for the application of the spectral block hybrid method, ultimately improving the accuracy and efficiency of computational solutions for nonlinear evolution equations.

1 | Introduction

Nonlinear evolution equations are prevalent in scientific disciplines, but their analytical solutions are often difficult to find [1, 2]. These equations play a fundamental role in modeling a wide range of phenomena across various scientific fields, including physics, engineering, chemistry, and biology [3–5]. Solving nonlinear ordinary differential equations

(N-ODEs) and nonlinear partial differential equations (N-PDEs) accurately and efficiently is crucial for gaining insight into the behavior of complex systems and predicting their dynamics [6–8]. Several authors have proposed techniques for obtaining analytical and approximate solutions to nonlinear evolution partial differential equations (N-EPDEs), including the modified KdV–Burgers, modified KdV, Fisher's, Burgers–Fisher, Burgers–Huxley, and FitzHugh–Nagumo equations [9, 10]. Researchers have found that numerical methods provide effective tools for solving these equations [11]. Numerical techniques are indispensable in this regard, providing reliable computational approximations to the solutions of N-EPDEs [12, 13].

Coupled Burgers' equations, in particular, are nonlinear partial differential equations that arise in various physical phenomena such as boundary layer motion, shock wave formation, and turbulence. Researchers have proposed several numerical methods to solve these equations efficiently, such as the Fourier pseudo-spectral method, variational iteration method, and Laplace transform homotopy perturbation method [14, 15]. Ahmad et al. [15] developed modified variational iteration algorithms (MVIA-I and MVIA-II) incorporating auxiliary parameters to accelerate the convergence rate of the series solutions compared with the traditional variational iteration method. Mittal and Arora [16] proposed a cubic B-spline collocation scheme coupled with the Crank–Nicolson method for time integration, demonstrating reliability and accuracy for viscous Burgers' equations. Mohammadi and Mokhtari [17] introduced a robust reproducing kernel Hilbert space (RKHS) based technique for solving challenging classes of N-PDEs, highlighting its advantages of being mesh-free, flexible, and capable of handling various boundary conditions without requiring time discretization or ODE integrators.

The existing literature can be broadly grouped into three categories: spectral methods, block hybrid and optimized hybrid methods, and linearization techniques for handling nonlinear terms. Several researchers have investigated spectral collocation methods for solving nonlinear differential equations using different strategies [18–20]. Pozrikidis [21] presented the theoretical foundation and practical implementation of spectral and finite element methods. Meuris et al. [22] introduced deep operator network–based spectral methods for solving N-PDEs, while Otegbeye and Motsa [23] developed the paired spectral quasi-linearization method. These spectral methods are known for achieving high accuracy and spectral convergence by representing the solution through global basis functions such as Fourier series or Lagrange polynomials [24, 25]. Motsa et al. [26] implemented the bivariate spectral quasi-linearization method (BSQLM), combining spectral collocation in space and time with the quasi-linearization method. They utilized the BSQLM method to solve N-EPDEs for large time domains. They concluded that the approximate solutions align with the exact solutions, demonstrating a high degree of accuracy. Magagula et al. [27] later extended the BSQLM to a multi-domain Legendre–Gauss–Lobatto bivariate spectral collocation method (L-BSQLM). Researchers have concluded that the L-BSQLM is more accurate and uses less computational time compared to the BSQLM. Khumalo et al. [28] solved the N-EPDEs by employing the bivariate spectral linear partition method (BSLPM). They observed that the BSLPM was precise, as its solution was compared to established exact solutions and found that suitable linearization plays a key role in ensuring consistency and convergence.

The block hybrid method (BHM) has attracted significant attention as a robust approach for solving nonlinear differential equations [29–32], following the pioneering work of Shampine and Watts [33]. Ahmedai et al. [34] proposed a BHM for solving third-order initial value problems using equally spaced intra-step points. They demonstrated that increasing the number of intra-step points enhances the accuracy of the BHM. Subsequent studies have refined the BHM through the introduction of optimized intra-step points to minimize local truncation errors [35–37]. The optimized block hybrid method (OBHM) has since been applied to various classes of N-EPDEs. For instance, Kaur and Kanwar [38] used the differential quadrature method (DQM) with optimized cubic trigonometric B-splines for the Kuramoto–Sivashinsky equation, achieving high accuracy with fewer grid points. Similarly, Ramos et al. [39] proposed a modified cubic B-spline method combined with an optimized hybrid block approach, demonstrating its efficiency for large time domains problems.

Linearization techniques play an essential role in approximating solutions to nonlinear differential equations by transforming them into linear systems that are easier to analyze and solve [40]. The choice of the linearization approach and reference point can significantly affect the accuracy and stability of the numerical solution [41, 42]. Classical linearization methods such as Taylor series, local, and piecewise approximations to nonlinear systems have been successfully applied in many engineering and physical models [43]. Modern linearization techniques, including perturbation, relaxation, quasi-linearization, Newton–Kantorovich, local linearization, linear partition, Picard and simple iteration methods, extend these classical concepts through iterative schemes to achieve higher accuracy and broader applicability [44–46]. The simple iteration method (SIM) has been employed for its simplicity and robustness in achieving convergence [34]. The present work builds on these ideas by integrating the SIM with spectral and optimized block hybrid methods to

improve the stability and accuracy of numerical solutions. The SIM adopts principles analogous to fixed-point iteration, serving as an efficient iterative technique that eliminates the need for derivative evaluations. Unlike conventional linearization approaches that rely on Taylor series expansions thereby introducing truncation errors even before numerical discretization. The SIM maintains accuracy by directly iterating on the nonlinear terms.

Thus, the aim of this study is to utilize spectral and block hybrid methods to develop an optimized block hybrid spectral simple iteration method (OBHM-SSIM) and to assess its performance against established numerical techniques in the literature. The novelty of the proposed OBHM-SSIM lies in the integration of the optimized block hybrid method with a spectral collocation scheme and a simple iteration linearization technique. To the best of our knowledge, this OBHM-SSIM combination has not previously been applied to N-EPDEs. The primary objective of this article is to develop and validate the OBHM-SSIM method by demonstrating its accuracy and efficiency in comparison with other established numerical methods. The study also aims to formulate a computationally straightforward and adaptable scheme applicable to diverse nonlinear problems. To achieve this, we address the following research questions:

- How accurate and efficient is the OBHM-SSIM compared to other methods in the literature for solving a range of N-EPDEs?
- Can we formulate the OBHM-SSIM scheme in a way that is straightforward to implement and apply to diverse N-EPDEs?
- How can the OBHM-SSIM be extended and effectively applied to system and two-dimensional partial differential equations while maintaining stability and spectral accuracy?

2 | Development of the Scheme

The main feature of the OBHM-SSIM is the use of a Chebyshev spectral collocation procedure in the space variable (x), and the discretization of the time variable (t) domain into smaller sub-intervals called blocks. To develop the OBHM-SSIM scheme, we consider the rth order N-PDE of the form

$$\frac{\partial u}{\partial t} = f(\mathbf{u}), \quad \mathbf{u} = \left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^3 u}{\partial x^3}, \ldots, \frac{\partial^r u}{\partial x^r}\right), \quad a < x < b, \quad 0 < t < t_F, \tag{1}$$

here r represents the order of higher derivative with respect to the variable x, and $r \geq 1$. Subject to the initial condition

$$u(0, x) = u_0(x), \tag{2}$$

and boundary conditions

$$\sigma_{11} u(t, a) + \sigma_{12} \left.\frac{\partial u}{\partial x}\right|_{(t,a)} = u_a(t), \quad \sigma_{13} u(t, b) + \sigma_{14} \left.\frac{\partial u}{\partial x}\right|_{(t,b)} = u_b(t), \tag{3}$$

where $\sigma_{11}, \sigma_{12}, \sigma_{13}$, and σ_{14} are constant parameters. In equation (1), the right hand side f is a continuous nonlinear function in u, $\frac{\partial u}{\partial x}$, $\frac{\partial^2 u}{\partial x^2}$, …, $\frac{\partial^r u}{\partial x^r}$.

We discretize the spatial domain, as shown in Figure 1. We assume that $x \in [a, b]$ is discretized into $N_x + 1$ Chebyshev–Gauss–Lobatto (CGL) points. The CGL points are chosen in this work primarily for their computational efficiency, as their Chebyshev-based structure enables fast and cost-effective computation of derivatives [19]. The collocation points are $\hat{x}_i = \cos \frac{\pi i}{N_x}$, where $i = 0, 1, \cdots, N_x$. We transform the interval $[a, b]$ into the interval $[-1, 1]$ through linear transformation mapping [47]

$$x = \frac{(b-a)}{2} \hat{x}_i + \frac{(b+a)}{2}. \tag{4}$$

We define that $t \in [0, t_F]$ is divided into N integrating blocks (see Figure 2) denoted as

$$I_n = [t_n, t_{n+1}], \quad n = 0, 1, 2, \ldots, N-1. \tag{5}$$

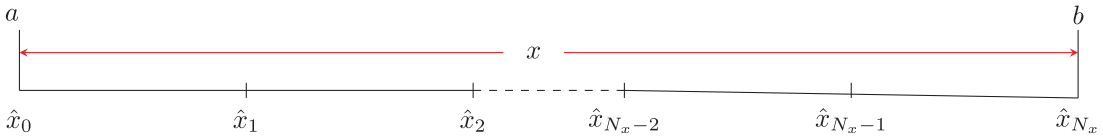

FIGURE 1 | Discretization of the space domain to $N_x + 1$ Chebyshev–Gauss–Lobatto points.

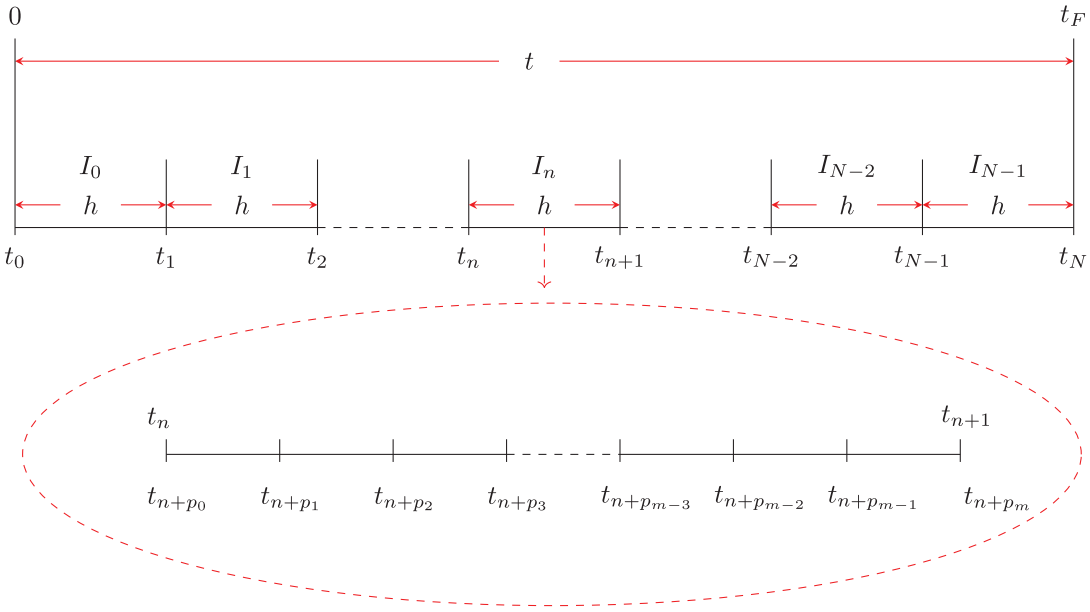

FIGURE 2 | Discretization of the time domain to N blocks.

The fixed step-size is denoted as $h = t_{n+1} - t_n$. We utilized the m intra-step, which satisfies

$$t_{n+p_j} = t_n + p_j\, h, \quad 0 = p_0 < p_1 < p_2 < \cdots < p_{m-1} < p_m = 1.$$

Equally spaced grid points were defined as

$$p_j = \frac{j}{m}, \quad j = 0, 1, 2, 3, \ldots, m. \tag{6}$$

Orakwelu [37] introduced that the optimal intra-step points are the roots of the polynomial

$$G_j(p) = \frac{1}{j(j+1)} \sum_{q=1}^{j} \frac{q(-1)^{j+q} p^{q-1}(j+q)!}{(q!)^2 (j-q)!}, \quad j = 2, 3, 4, \ldots. \tag{7}$$

Equation (7) is proportional to the first derivative of the shifted Legendre polynomials of degree m. The shifted Legendre polynomials are defined on $[0, 1]$. By expanding Equation (7), we obtain the polynomials that give the optimal intra-step points with $m \geq 2$. Table 1 presents the intra-step points for both the equally spaced and optimal cases for various values of m.

Approximating u at each block n, using a Lagrange interpolating polynomial of the form

$$u(t, x) \approx \sum_{j=0}^{m} \sum_{i=0}^{N_x} u(t_{n+p_j}, \hat{x}_i) L_i(x) L_n(t), \tag{8}$$

where $L_i(x)$ and $L_n(t)$ are the Lagrange polynomials. $L_i(x)$ is the characteristic Lagrange polynomial based on the Chebyshev–Gauss–Lobatto points [48], defined as

$$L_i(x) = \prod_{\substack{i=0 \\ i \neq k}}^{N_x} \frac{x - x_k}{x_i - x_k}. \tag{9}$$

TABLE 1 | Equally spaced and optimal intra-step points for $m = 2, 3, 4$, and 5.

m	p_j	Equally spaced	Optimal
2	$\{p_0, p_1, p_2\}$	$\left\{0, \frac{1}{2}, 1\right\}$	$\left\{0, \frac{1}{2}, 1\right\}$
3	$\{p_0, p_1, p_2, p_3\}$	$\left\{0, \frac{1}{3}, \frac{2}{3}, 1\right\}$	$\left\{0, \frac{1}{10}(5-\sqrt{5}), \frac{1}{10}(5+\sqrt{5}), 1\right\}$
4	$\{p_0, p_1, p_2, p_3, p_4\}$	$\left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}$	$\left\{0, \frac{1}{14}(7-\sqrt{21}), \frac{1}{2}, \frac{1}{14}(7+\sqrt{21}), 1\right\}$
5	$\{p_0, p_1, p_2, p_3, p_4, p_5\}$	$\left\{0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1\right\}$	$\left\{0, \frac{1}{2}-\frac{1}{2}\sqrt{\frac{1}{3}\pm\frac{2\sqrt{7}}{21}}, \frac{1}{2}, \frac{1}{2}+\frac{1}{2}\sqrt{\frac{1}{3}\pm\frac{2\sqrt{7}}{21}}, 1\right\}$

From Equation (8), the approximated spatial derivative is defined as

$$\left.\frac{\partial u}{\partial x}\right|_{(t_{n+p_j}, \hat{x}_i)} \approx \sum_{j=0}^{m}\sum_{l=0}^{N_x} u(t_{n+p_j}, \hat{x}_i)\frac{d\,L_i(x)}{dx}L_n(t_{n+p_j}), \tag{10}$$

$$= \sum_{j=0}^{m}\sum_{l=0}^{N_x} D_{i,l}\; u(t_{n+p_j}, \hat{x}_i) = DU_{n+p_j}. \tag{11}$$

The spatial derivatives are approximated using a differentiation matrix D as defined in [19]. Where the matrix D is of size $(N_x+1)\times(N_x+1)$ of the form

$$D = \begin{bmatrix} D_{0,0} & D_{0,1} & \cdots & D_{0,N_x-1} & D_{0,N_x} \\ D_{1,0} & D_{1,1} & \cdots & D_{1,N_x-1} & D_{1,N_x} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ D_{N_x,0} & D_{N_x,1} & \cdots & D_{N_x,N_x-1} & D_{N_x,N_x} \end{bmatrix}, \tag{12}$$

then the unknown vector U_{n+p_j} is of size $(N_x+1)\times 1$ defined as

$$U_{n+p_j} = \left[u(t_{n+p_j}, \hat{x}_0), u(t_{n+p_j}, \hat{x}_1), u(t_{n+p_j}, \hat{x}_2), \ldots, u(t_{n+p_j}, \hat{x}_{N_x})\right]^T, \tag{13}$$

and where the superscript (T) represents the vector transpose. Similarly, the rth order spatial derivatives are computed as

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{(t_{n+p_j}, \hat{x}_i)} \approx \sum_{l=0}^{N_x} D_{i,l}^2\; u(t_{n+p_j}, \hat{x}_i) = D^2U_{n+p_j}, \cdots, \left.\frac{\partial^r u}{\partial x^r}\right|_{(t_{n+p_j}, \hat{x}_i)} \approx \sum_{l=0}^{N_x} D_{i,l}^r\; u(t_{n+p_j}, \hat{x}_i) = D^rU_{n+p_j}. \tag{14}$$

The N-EPDE (1) must be linearized before applying the spectral collocation method. To simplify the computations and develop the iteration scheme, we linearized f using the simple iteration method. Equation (1) is expressed as a sum of its linear and nonlinear components as

$$f = \mathscr{L}^* + \mathscr{N}^*, \tag{15}$$

where $\mathscr{L}^*$ can be written as the sum of linear terms (unknown functions) at iteration $(s+1)$, as

$$\mathscr{L}^* = \mathscr{L}_r^*(t,x)\overset{(s+1)}{\frac{\partial^r u}{\partial x^r}} + \mathscr{L}_{r-1}^*(t,x)\overset{(s+1)}{\frac{\partial^{r-1} u}{\partial x^{r-1}}} + \cdots + \mathscr{L}_2^*(t,x)\overset{(s+1)}{\frac{\partial^2 u}{\partial x^2}} + \mathscr{L}_1^*(t,x)\overset{(s+1)}{\frac{\partial u}{\partial x}} + \mathscr{L}_0^*(t,x)\overset{(s+1)}{u}, \tag{16}$$

and the nonlinear terms $\mathscr{N}^*$ are a combination of known (at iteration (s)) and unknown functions (at iteration $(s+1)$). Here, s denotes the iteration index used in the simple iteration process, where $s \geq 1$. In the nonlinear terms, the term with a higher derivative is taken as unknown. The $\mathscr{N}^*$ can be written as the sum of nonlinear terms as

$$\mathcal{N}^{*} = \mathcal{N}_r\left(\overset{(s)}{u}, \frac{\partial \overset{(s)}{u}}{\partial x}, \frac{\partial^2 \overset{(s)}{u}}{\partial x^2}, \ldots, \frac{\partial^r \overset{(s)}{u}}{\partial x^r}\right)\frac{\partial^r \overset{(s+1)}{u}}{\partial x^r} + \mathcal{N}_{r-1}\left(\overset{(s)}{u}, \frac{\partial \overset{(s)}{u}}{\partial x}, \frac{\partial^2 \overset{(s)}{u}}{\partial x^2}, \ldots, \frac{\partial^{r-1} \overset{(s)}{u}}{\partial x^{r-1}}\right)\frac{\partial^{r-1} \overset{(s+1)}{u}}{\partial x^{r-1}} + \ldots$$
$$+ \mathcal{N}_2\left(\overset{(s)}{u}, \frac{\partial \overset{(s)}{u}}{\partial x}, \frac{\partial^2 \overset{(s)}{u}}{\partial x^2}\right)\frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2} + \mathcal{N}_1\left(\overset{(s)}{u}, \frac{\partial \overset{(s)}{u}}{\partial x}\right)\frac{\partial \overset{(s+1)}{u}}{\partial x} + \mathcal{N}_0(\overset{(s)}{u})\overset{(s+1)}{u} + \mathcal{H}(t,x). \tag{17}$$

Substituting Equations (16) and (17) into Equation (15), we obtain

$$\overset{(s+1)}{f} = \mathcal{L}_r\frac{\partial^r \overset{(s+1)}{u}}{\partial x^r} + \mathcal{L}_{r-1}\frac{\partial^{r-1} \overset{(s+1)}{u}}{\partial x^{r-1}} + \cdots + \mathcal{L}_2\frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2} + \mathcal{L}_1\frac{\partial \overset{(s+1)}{u}}{\partial x} + \mathcal{L}_0\overset{(s+1)}{u} + \mathcal{H}, \tag{18}$$

where $\mathcal{L}_r, \mathcal{L}_{r-1}, \ldots, \mathcal{L}_0$ are known functions from a previous iteration (s) given by

$$\mathcal{L}_r = \mathcal{L}_r^{*}(t,x) + \mathcal{N}_r\left(\overset{(s)}{u}, \frac{\partial \overset{(s)}{u}}{\partial x}, \frac{\partial^2 \overset{(s)}{u}}{\partial x^2}, \ldots, \frac{\partial^r \overset{(s)}{u}}{\partial x^r}\right),$$
$$\mathcal{L}_{r-1} = \mathcal{L}_{r-1}^{*}(t,x) + \mathcal{N}_{r-1}\left(\overset{(s)}{u}, \frac{\partial \overset{(s)}{u}}{\partial x}, \frac{\partial^2 \overset{(s)}{u}}{\partial x^2}, \ldots, \frac{\partial^{r-1} \overset{(s)}{u}}{\partial x^{r-1}}\right),$$
$$\vdots$$
$$\mathcal{L}_2 = \mathcal{L}_2^{*}(t,x) + \mathcal{N}_2\left(\overset{(s)}{u}, \frac{\partial \overset{(s)}{u}}{\partial x}, \frac{\partial^2 \overset{(s)}{u}}{\partial x^2}\right),$$
$$\mathcal{L}_1 = \mathcal{L}_1^{*}(t,x) + \mathcal{N}_1\left(\overset{(s)}{u}, \frac{\partial \overset{(s)}{u}}{\partial x}\right),$$
$$\mathcal{L}_0 = \mathcal{L}_0^{*}(t,x) + \mathcal{N}_0(\overset{(s)}{u}).$$

Equation (18) is the linearized form of Equation (15). The spectral method with $N_x + 1$ Chebyshev–Gauss–Lobatto points on Equations (1) to (3) gives

$$\dot{U}_{n+p_j} = f_{n+p_j} = L_{n+p_j}\, U_{n+p_j} + H_{n+p_j}, \tag{19}$$

where

$$H_{n+p_j} = \mathcal{H}(t_{n+P_j}, \hat{x}_i),$$
$$L_{n+p_j} = \mathcal{L}_r(t_{n+P_j}, \hat{x}_i) + \mathcal{N}_r\left(t_{n+p_j}, \hat{x}_i, \overset{(s)}{u}_{n+p_j}, D\overset{(s)}{u}_{n+p_j}, D^2\overset{(s)}{u}_{n+p_j}, \ldots, D^r\overset{(s)}{u}_{n+p_j}\right)D^r$$
$$+ \mathcal{L}_{r-1}(t_{n+P_j}, \hat{x}_i) + \mathcal{N}_{r-1}\left(t_{n+p_j}, \hat{x}_i, \overset{(s)}{u}_{n+p_j}, D\overset{(s)}{u}_{n+p_j}, D^2\overset{(s)}{u}_{n+p_j}, \ldots, D^{r-1}\overset{(s)}{u}_{n+p_j}\right)D^{r-1}$$
$$\vdots$$
$$+ \mathcal{L}_2\left(t_{n+P_j}, \hat{x}_i) + \mathcal{N}_2(t_{n+p_j}, \hat{x}_i, \overset{(s)}{u}_{n+p_j}, D\overset{(s)}{u}_{n+p_j}, D^2\overset{(s)}{u}_{n+p_j}\right)D^2$$
$$+ \mathcal{L}_1\left(t_{n+P_j}, \hat{x}_i) + \mathcal{N}_1(t_{n+p_j}, \hat{x}_i, \overset{(s)}{u}_{n+p_j}, D\overset{(s)}{u}_{n+p_j}\right)D + \mathcal{L}_0(t_{n+P_j}, \hat{x}_i) + \mathcal{N}_0\left(t_{n+p_j}, \hat{x}_i, \overset{(s)}{u}_{n+p_j}\right).$$

The dot denotes the derivative with respect to time t. Subject to the initial condition

$$U(0, \hat{x}_i) = u_0(\hat{x}_i), \tag{20}$$

and boundary conditions

$$\sigma_{11}U(t_{n+p_j}, \hat{x}_{N_x}) + \sigma_{12}\overline{D}U(t_{n+p_j}, \hat{x}_{N_x}) = u_a(t_{n+p_j}), \tag{21}$$

$$\sigma_{21}U(t_{n+p_j}, \hat{x}_0) + \sigma_{22}\overline{D}U(t_{n+p_j}, \hat{x}_0) = u_b(t_{n+p_j}). \tag{22}$$

To apply the optimized block hybrid method with Lagrange interpolation polynomials to the right-hand side of Equation (19), we write the time derivative as

$$\dot{U}(t_n + \tau h, \hat{x_i}) = \sum_{\omega=0}^{m} U(t_{n+p_j}, \hat{x_i}) L_\omega(\tau), \tag{23}$$

where

$$L_\omega(\tau) = \prod_{\substack{j=0 \\ j\neq\omega}}^{m} \frac{\tau - p_\omega}{p_j - p_\omega}. \tag{24}$$

Substituting Equation (23) into Equation (19) and integrating from 0 to p_j, we obtain

$$U_{n+p_j} = U_n + h \sum_{\omega=0}^{m} \alpha_{j\omega}\ f_{n+p_j} \tag{25}$$

$$= U_n + h\ \alpha_{j0}\ F_n + h \sum_{\omega=1}^{m} \alpha_{j\omega}\ \left(L_{n+p_\omega}\ U_{n+p_\omega} + H_{n+p_\omega}\right), \tag{26}$$

where the coefficients $\alpha_{j\omega}$ and α_{j0} are known. Solving Equation (26) in terms of U_{n+p_ω}, we get

$$U_{n+p_j} - h \sum_{\omega=1}^{m} \alpha_{j\omega}\ \ L_{n+p_\omega}\ U_{n+p_\omega} = U_n + h\alpha_{j,0}\ F_n + h \sum_{\omega=1}^{m} \alpha_{j,\omega}\ H_{n+p_\omega}. \tag{27}$$

Equation (27) can be written in expanded matrix form as

$$\begin{bmatrix} I & O & \cdots & O \\ O & I & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & I \end{bmatrix} \begin{bmatrix} U_{n+p_1} \\ U_{n+p_2} \\ \vdots \\ U_{n+p_m} \end{bmatrix} - h \begin{bmatrix} \alpha_{11}I & \alpha_{12}I & \cdots & \alpha_{1,m}I \\ \alpha_{21}I & \alpha_{22}I & \cdots & \alpha_{2m}I \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1}I & \alpha_{m2}I & \cdots & \alpha_{mm}I \end{bmatrix} \begin{bmatrix} L_{n+p_1} & O & \cdots & O \\ O & L_{n+p_2} & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & L_{n+p_m} \end{bmatrix} \begin{bmatrix} U_{n+p_1} \\ U_{n+p_2} \\ \vdots \\ U_{n+p_m} \end{bmatrix}$$
$$= \begin{bmatrix} U_n \\ U_n \\ \vdots \\ U_n \end{bmatrix} + h \begin{bmatrix} \alpha_{10}I & O & \cdots & O \\ O & \alpha_{20}I & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & \alpha_{m0}I \end{bmatrix} \begin{bmatrix} F_n \\ F_n \\ \vdots \\ F_n \end{bmatrix} + h \begin{bmatrix} \alpha_{11}I & \alpha_{12}I & \cdots & \alpha_{1m}I \\ \alpha_{21}I & \alpha_{22}I & \cdots & \alpha_{2m}I \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1}I & \alpha_{m2}I & \cdots & \alpha_{mm}I \end{bmatrix} \begin{bmatrix} H_{n+p_1} \\ H_{n+p_2} \\ \vdots \\ H_{n+p_m} \end{bmatrix}. \tag{28}$$

The matrix form Equation (28) can be written in terms of Kronecker products as

$$\left[\mathbf{I} - h(A \otimes I)\mathbf{L}_{n+p}\right]\mathbf{U}_{n+p} = \mathbf{U}_n + h(A_0 \otimes I)\mathbf{F}_n + h(A \otimes I)\mathbf{H}_{n+p}, \tag{29}$$

where $\mathbf{I}$ and I are the identity matrices of size $m(N_x+1) \times m(N_x+1)$ and $(N_x+1)\times(N_x+1)$ respectively. O is a zero matrix of size $(N_x+1)\times(N_x+1)$. A and A_0 matrices of size $(m \times m)$ are defined as follows:

$$\mathbf{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1m} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mm} \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} \alpha_{10} & 0 & \cdots & 0 \\ 0 & \alpha_{20} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{m,0} \end{bmatrix},$$

and $\mathbf{U}_n$, $\mathbf{G}_{n+p}$, and $\mathbf{F}_n$ are vectors of size $m(N_x+1)\times 1$ are defined as follows:

$$\mathrm{U}_n=\begin{bmatrix}U_n\\U_n\\\vdots\\U_n\end{bmatrix},\quad \mathbf{H}_{n+p}=\begin{bmatrix}H(t_{n+p_1},\hat{x}_0)\\H(t_{n+p_1},\hat{x}_1)\\\vdots\\H(t_{n+p_m},\hat{x}_{N_x})\end{bmatrix},\quad \mathbf{F}_n=\begin{bmatrix}F_n\\F_n\\\vdots\\F_n\end{bmatrix}.$$

Equation (29) may be reduced and written in the form

$$\boldsymbol{\Delta}\mathbf{U}_{n+p}=\mathbf{K}, \tag{30}$$

where

$$\boldsymbol{\Delta}=\mathbf{I}-h(A\otimes I)\mathbf{L}_{n+p},$$
$$\mathbf{K}=\mathbf{U}_n+h(A_0\otimes I)\mathbf{F}_n+h(A\otimes I)\mathbf{H}_{n+p},$$

and the size of the matrix $\boldsymbol{\Delta}$ is $m(N_x+1)\times m(N_x+1)$ and $\mathbf{K}$ is size $m(N_x+1)\times 1$. Hence, the approximate solutions may be found by finding the inverse of $\boldsymbol{\Delta}$ in Equation (30). Thus, we have

$$\mathbf{U}_{n+p}=\mathbf{inv}(\boldsymbol{\Delta})\mathbf{K}, \tag{31}$$

where **inv** is the inverse of the matrix and $\boldsymbol{\Delta}$ is nonsingular matrix.

3 | Theoretical Analysis

In this section, we present analysis of the convergence and stability properties of the proposed OBHM-SSIM. The error is decomposed into the spatial discretization error and temporal discretization error.

3.1 | Spatial Discretization Error and Spectral Convergence

The spatial discretization is performed using the Chebyshev spectral collocation method. The convergence of this approach depends on the smoothness of the solution. We formalize this well-known property in the following proposition.

Proposition 1. *Let $u(t,x)$ denote the exact solution to Equation (1), and let $U_{N_x}(t,\hat{x}_i)$ be its degree-N_x polynomial interpolate at the CGL nodes, which form the basis for the semi-discretized system (19).*

- *If, for a fixed time t_n, the solution $u(t_n,\hat{x}_i)\in C^\infty([-1,1])$, the interpolation error decays faster than any algebraic power of N_x; that is, for any integer $k>0$, there exists a constant C_k, independent of N_x, such that*

$$\|u(t_n,\hat{x}_i)-U_{N_x}(t_n,\hat{x}_i)\|_\infty\le C_k N_x^{-k},\quad \text{as } N_x\to\infty. \tag{32}$$

- *If $u(t,\cdot)$ is analytic in an open region of the complex plane containing the interval $[-1,1]$, the convergence is exponential. Specifically, there exist constants $C>0$ and $\rho>1$ such that*

$$\|u(t,\cdot)-U_{N_x}(t,\cdot)\|_\infty\le C\rho^{-N_x}. \tag{33}$$

For the proof of Proposition 1, see [19, 49]. The algebraic convergence for C^∞ functions and exponential convergence for analytic functions are well-established results for Chebyshev interpolation [50]. Although the approximation of derivatives in Equation (19) can introduce a polynomial factor in N_x, it does not affect the overall spectral rate of convergence for sufficiently smooth solutions [25].

3.2 | Temporal Discretization Error and Stability Analysis

3.2.1 | The Local Truncation Error

To determine the accuracy of the OBHM-SSIM time-stepping scheme, we evaluate the local truncation error (LTE). We introduce the linear operator $\mathfrak{L}$, which represents the difference between the exact solution and its numerical approximation. Using Equation (25), the LTE is analyzed through the linear operator $\mathfrak{L}$, defined as

$$\mathfrak{L}_l[U(t_n,\hat{x}_i);h] = U(t_n+hp_j,\hat{x}_i) - U(t_n,\hat{x}_i) - h\sum_{\omega=0}^{m}\alpha_{j\omega}\dot{U}(t_n+hp_j,\hat{x}_i). \tag{34}$$

We assume that $U(t,\hat{x}_i)$ is sufficiently differentiable. Expanding the terms $U(t_n+hp_i,\hat{x}_i)$ and $\dot{U}(t_n+hp_i,\hat{x}_i)$ by using a Taylor series about t_n, we obtain

$$\mathfrak{L}_l[U(t_n,\hat{x}_i);h] = C_{l,0}U(t_n,\hat{x}_i) + C_{l,1}h\dot{U}(t_n,\hat{x}_i) + C_{l,2}h^2\ddot{U}(t_n,\hat{x}_i) + C_{l,3}h^3\dddot{U}(t_n,\hat{x}_i) + \cdots + C_{l,q}h^qU^{(q)}(t_n,\hat{x}_i), \tag{35}$$

where $C_{l,0}, C_{l,1}, \ldots, C_{l,q}$ are constants. The method is said to be of order q if

$$\hat{C}_0 = \hat{C}_1 = \cdots = \hat{C}_{q+1} = 0, \quad \text{and} \quad \hat{C}_{q+2} \neq 0, \tag{36}$$

where $\hat{C}_q = [C_{1,q}, C_{2,q}, \ldots, C_{m,q}]^T$. The vector $\hat{C}_{q+2} = [C_{1,q+2}, C_{2,q+2}, \ldots, C_{m,q+2}]^T$ is the error constant of the method [51]. In Equation (34), applying the Taylor series about t_n leads to

$$\begin{aligned}\mathfrak{L}_l[U(t_n,\hat{x}_i);h] = &\left(U(t_n,\hat{x}_i) + hp_j\dot{U}(t_n,\hat{x}_i) + \frac{(hp_j)^2}{2!}\ddot{U}(t_n,\hat{x}_i) + \frac{(hp_j)^3}{3!}\dddot{U}(t_n,\hat{x}_i) + \cdots\right) - U(t_n,\hat{x}_i)\\ &- h\sum_{\omega=0}^{m}\alpha_{j\omega}\left(\dot{U}(t_n,\hat{x}_i) + hp_j\ddot{U}(t_n,\hat{x}_i) + \frac{(hp_j)^2}{2!}\dddot{U}(t_n,\hat{x}_i) + \cdots\right)\end{aligned} \tag{37}$$

$$= \sum_{k=1}^{K}\frac{h^kp_j^k}{k!}U^{(k)}(t_n,\hat{x}_i) - \sum_{k=1}^{K}\frac{kh^k}{k!}\sum_{\omega=0}^{m}\alpha_{j\omega}p_j^{k-1}U^{(k)}(t_n,\hat{x}_i) + \mathcal{O}(h^{K+1}), \tag{38}$$

where K is a positive integer. Expanding Equation (38) gives

$$\begin{aligned}\mathfrak{L}_l[U(t_n,\hat{x}_i);h] = &\sum_{k=1}^{m+1}\frac{h^kp_j^k}{k!}U^{(k)}(t_n,\hat{x}_i) + \sum_{k=m+2}^{K}\frac{h^kp_j^k}{k!}U^{(k)}(t_n,\hat{x}_i)\\ &- \sum_{k=1}^{m+1}\frac{kh^k}{k!}\sum_{\omega=0}^{m}\alpha_{j\omega}p_j^{k-1}U^{(k)}(t_n,\hat{x}_i) - \sum_{k=m+2}^{K}\frac{kh^k}{k!}\sum_{\omega=0}^{m}\alpha_{j\omega}p_j^{k-1}U^{(k)}(t_n,\hat{x}_i) + \mathcal{O}(h^{k+1}).\end{aligned} \tag{39}$$

Following [52], Equation (39) may be written as

$$\begin{aligned}\mathfrak{L}_l[U(t_n,\hat{x}_i);h] = &\sum_{k=1}^{m+1}\frac{h^k}{k!}\left[p_j^k - k\sum_{\omega=0}^{m}\alpha_{j\omega}p_j^{k-1}\right]U^{(k)}(t_n,\hat{x}_i)\\ &+ \sum_{k=m+2}^{K}\frac{h^k}{k!}\left[p_j^k - k\sum_{\omega=0}^{m}\alpha_{j\omega}p_j^{k-1}\right]U^{(k)}(t_n,\hat{x}_i) + \mathcal{O}(h^{K+1})\end{aligned} \tag{40}$$

$$= \frac{h^{m+2}}{(m+2)!}\left[p_j^{m+2} - (m+2)\sum_{\omega=0}^{m}\alpha_{j\omega}p_j^{m+1}\right]U^{(m+2)}(t_n,\hat{x}_i) + \mathcal{O}(h^{m+3}). \tag{41}$$

From Equation (41), we deduce that the error constant vector is defined as

$$C_{l,m+2} = \frac{h^{m+2}}{(m+2)!}\left[p_j^{m+2} - (m+2)\sum_{\omega=0}^{m}\alpha_{j\omega}p_j^{m+1}\right]U^{(m+2)}(t_n,\hat{x}_i).$$

From Equation (41), the leading term in the LTE is proportional to $\mathcal{O}(h^{m+2})$. A method has order q if the LTE satisfies

$$\mathfrak{L}_l[U(t_n,\hat{x}_i);h] = \mathcal{O}(h^{q+1}). \tag{42}$$

The analysis of the LTE in Equation (41) shows that its leading term is of order $\mathcal{O}(h^{m+2})$. By the standard definition of accuracy, where the LTE for a method of order q is $\mathcal{O}(h^{q+1})$, as shown in (42), we deduce that the order of the OBHM is

TABLE 2 | Truncation errors for different m for the equally spaced and optimal intra-step points.

m	Equally spaced intra-step points	Optimal intra-step points
2	$\frac{1}{384}h^4U^{(4)}$, $\frac{-1}{2880}h^5U^{(5)}$	$\frac{1}{384}h^4U^{(4)}$, $\frac{-1}{2880}h^5U^{(5)}$
3	$\frac{-19}{174,960}h^5U^{(5)}$, $\frac{-1}{21870}h^5U^{(5)}$, $\frac{-1}{6480}h^5U^{(5)}$	$\frac{-1}{6000\sqrt{5}}h^5U^{(5)}$, $\frac{1}{6000\sqrt{5}}h^5U^{(5)}$, $\frac{-1}{1,512,000}h^7U^{(7)}$
4	$\frac{3}{655,360}h^6U^{(6)}$, $\frac{1}{368,640}h^6U^{(6)}$, $\frac{3}{655,360}h^6U^{(6)}$ $\frac{-1}{1,935,360}h^7U^{(7)}$	$\frac{1}{493,920}h^6U^{(6)}$, $\frac{-1}{322,560}h^6U^{(6)}$, $\frac{1}{493,920}h^6U^{(6)}$ $\frac{-1}{1,422,489,600}h^9U^{(9)}$
5	$\frac{-863}{4,725,000,000}h^7U^{(7)}$, $\frac{-37}{295,312,500}h^7U^{(7)}$ $\frac{-29}{175,000,000}h^7U^{(7)}$, $\frac{-8}{73,828,125}h^7U^{(7)}$ $\frac{-11}{37,800,000}h^7U^{(7)}$	$\frac{-\sqrt{147-30\sqrt{7}}}{160,030,080}h^7U^{(7)}$, $\frac{\sqrt{147+30\sqrt{7}}}{160,030,080}h^7U^{(7)}$ $\frac{-\sqrt{147+30\sqrt{7}}}{160,030,080}h^7U^{(7)}$, $\frac{\sqrt{147-30\sqrt{7}}}{160,030,080}h^7U^{(7)}$ $\frac{-1}{2,112,397,056,000}h^{11}U^{(11)}$

TABLE 3 | Stability function $\mathcal{G}(z)$ for different values of m.

m	Stability function
2	$\frac{z^2+6z+12}{z^2-6z+12}$
3	$\frac{50,137z^3+600,774z^2+3,001,500z+6,000,000}{3(16629z^3-199,758z^2+999,500z-2,000,000)}$
4	$\frac{35,811z^4+715,366z^3+6,434,232z^2+30,012,000z+60,000,000}{35561z^4-712866z^3+6,422,232z^2-29,988,000z+60,000,000}$
5	$\frac{59,962.08409z^5+1,793,536.10818z^4+25,066,670.82z^3+200,303,340z^2+900,600,000z+1,800,000,000}{3(19,709.53803z^5-592,845.09606z^4+8,312,223.06z^3-66,567,780z^2+299,800,000z-600,000,000)}$

$q = m + 1$. Consequently, the method's consistency is assured. As established in [53, 54], a block method is consistent if its order $q \geq 1$. For the present scheme, with $m \geq 2$, the resulting order $q \geq 3$ comfortably satisfies this requirement. Table 2 presents the truncation error and order for $m = 2, 3, 4, 5$. From the table, it is observed that the optimal intra-step points yield a smaller LTE compared to the equally spaced intra-step points for $m = 3, 4, 5$.

3.2.2 | Absolute Stability

To examine the absolute stability, we implemented the method on the scalar linear test equation, as outlined in [55]. Following this approach, we define

$$\dot{U} = \lambda U, \quad \lambda \in \mathbb{C}. \tag{43}$$

Substituting Equation (43) into Equation (25), yields

$$U_{n+p_j} = U_n + \lambda h \sum_{\omega=0}^{m} \alpha_{j\omega} U_{n+p_j}, \quad i = 1, 2, \ldots, m. \tag{44}$$

Equation (44) can be reduced to the matrix form as

$$\mathbf{U}_{n+p} = \mathcal{G}(z)\mathbf{U}_n, \quad \text{where} \quad \mathcal{G}(z) = (I + \mathbf{A}z)(I - \mathbf{A_0}z)^{-1}, \quad \text{and} \quad z = \lambda h, \tag{45}$$

where $\mathcal{G}$ is the stability function. Table 3 shows the stability functions for the optimized block hybrid method for different values, specifically $m = 2, 3, 4, 5$.

Definition 1. Following [56], a method is A-stable if its region of absolute stability

$$R = \{z \in \mathbb{C} : |\mathcal{G}(z)| \leq 1\},$$

contains the entire left-half complex plane $\mathbb{C}^- = \{z \in \mathbb{C} : \mathrm{Re}(z) \leq 0\}$. A complete proof requires satisfying the following two conditions

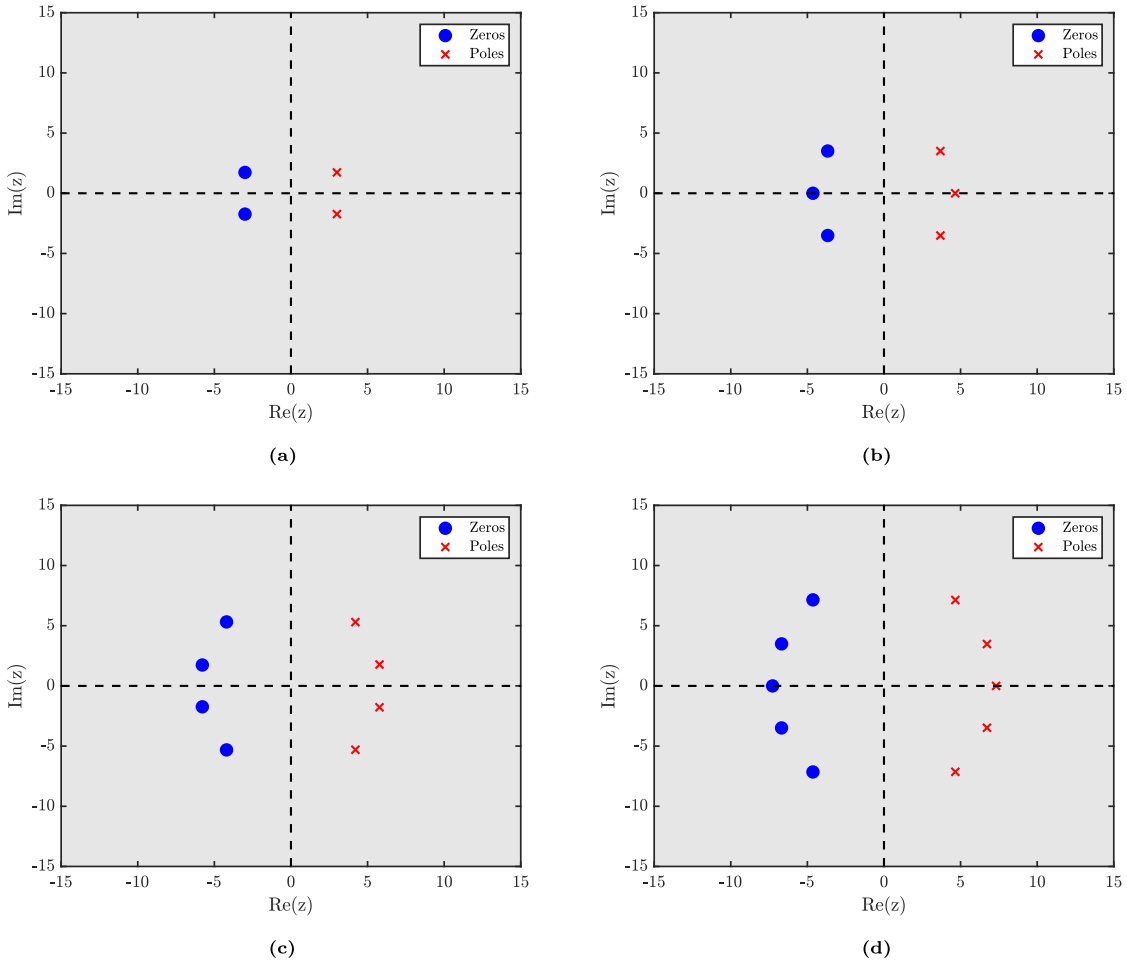

FIGURE 3 | The poles and zeros of the stability function for various values of m. (a) $m = 2$, (b) $m = 3$, (c) $m = 4$, (d) $m = 5$.

- The stability function $\mathcal{G}(z)$ must have no poles in the open left-half plane (as illustrated in Figure 3).
- It must be shown that $|\mathcal{G}(z)| \leq 1$ for all $z \in \mathbb{C}$.

The absolute stability region shown in Figure 4 confirms this result, as it entirely covers the left-half plane. Hence, the method for $m = 2, 3, 4, 5$ is A-stable.

Definition 2. Following [57], a method is L-stable if it satisfies both

- Its region of absolute stability contains the entire left half-plane.
- Its stability function $\mathcal{G}(z)$ satisfies $\lim_{z\to\infty} |\mathcal{G}(z)| = 0$.

From Table 3, the asymptotic behavior of each stability function as $z \to \infty$ is evaluated. It is observed that none of the stability functions satisfy $\lim_{z\to\infty} |\mathcal{G}(z)| = 0$, indicating that the method is not L-stable for any of the cases $m = 2, 3, 4, 5$.

3.2.3 | Order Stars

To gain insight beyond the region of absolute stability, we employ order star theory [58]. This framework characterizes a method's qualitative accuracy by comparing its stability function $\mathcal{G}(z) = \mathcal{G}_1(z)/\mathcal{G}_2(z)$ to the ideal behavior of e^z [59]. The comparison defines three regions [60]

$$R_+ = \left\{ z \in \mathbb{C} : \left| \frac{\mathcal{G}(z)}{e^z} \right| > 1 \right\}, \quad R_0 = \left\{ z \in \mathbb{C} : \left| \frac{\mathcal{G}(z)}{e^z} \right| = 1 \right\}, \quad R_- = \left\{ z \in \mathbb{C} : \left| \frac{\mathcal{G}(z)}{e^z} \right| < 1 \right\}.$$

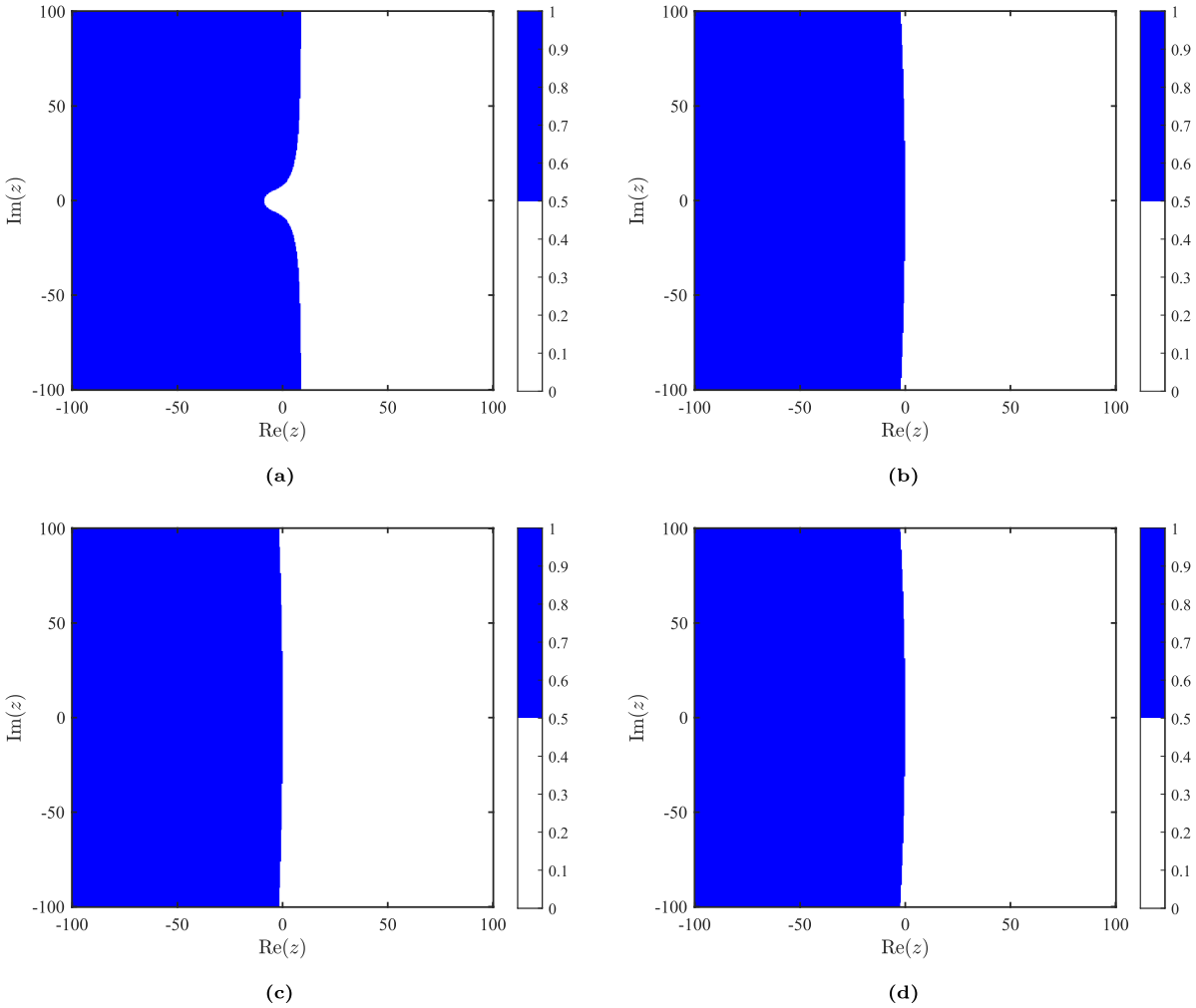

FIGURE 4 | The absolute stability regions for various values of m. (a) $m = 2$, (b) $m = 3$, (c) $m = 4$, (d) $m = 5$.

Key features of this map are the method's zeros (roots of $\mathcal{G}_1(z)$) and poles (roots of $\mathcal{G}_2(z)$). The zeros appear as points where stable regions contract, indicating strong damping properties, while the poles act as sources of instability where unstable regions diverge. Figure 5 shows the complete order star for the method for $m = 2, 3, 4, 5$.

3.2.4 | Zero Stability

Zero stability ensures that the numerical solution remains bounded as the number of steps increases [61]. For the block hybrid method, writing Equation (25) in matrix form and letting $h \to 0$, the system reduces to

$$I\mathbf{U}_{n+p} - \hat{\mathbf{A}}\mathbf{U}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} U_{n+p_1} \\ U_{n+p_2} \\ \vdots \\ U_{n+p_m} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} U_n \\ U_n \\ \vdots \\ U_n \end{bmatrix}. \tag{46}$$

Equation (46) gives the following characteristic equation

$$\rho(\lambda) = \det(\lambda I - \hat{A}) = \lambda^m(\lambda - 1). \tag{47}$$

The roots of this characteristic equation, known as the characteristic roots, must satisfy $|\rho(\lambda)| \leq 1$. Furthermore, any roots with modulus one must have multiplicity one. This ensures that the method is zero-stable, which is a prerequisite for convergence. The method developed in this study is zero-stable. Since the method exhibits both zero-stability and consistency, Dahlquist's theorem implies that it is convergent [51].

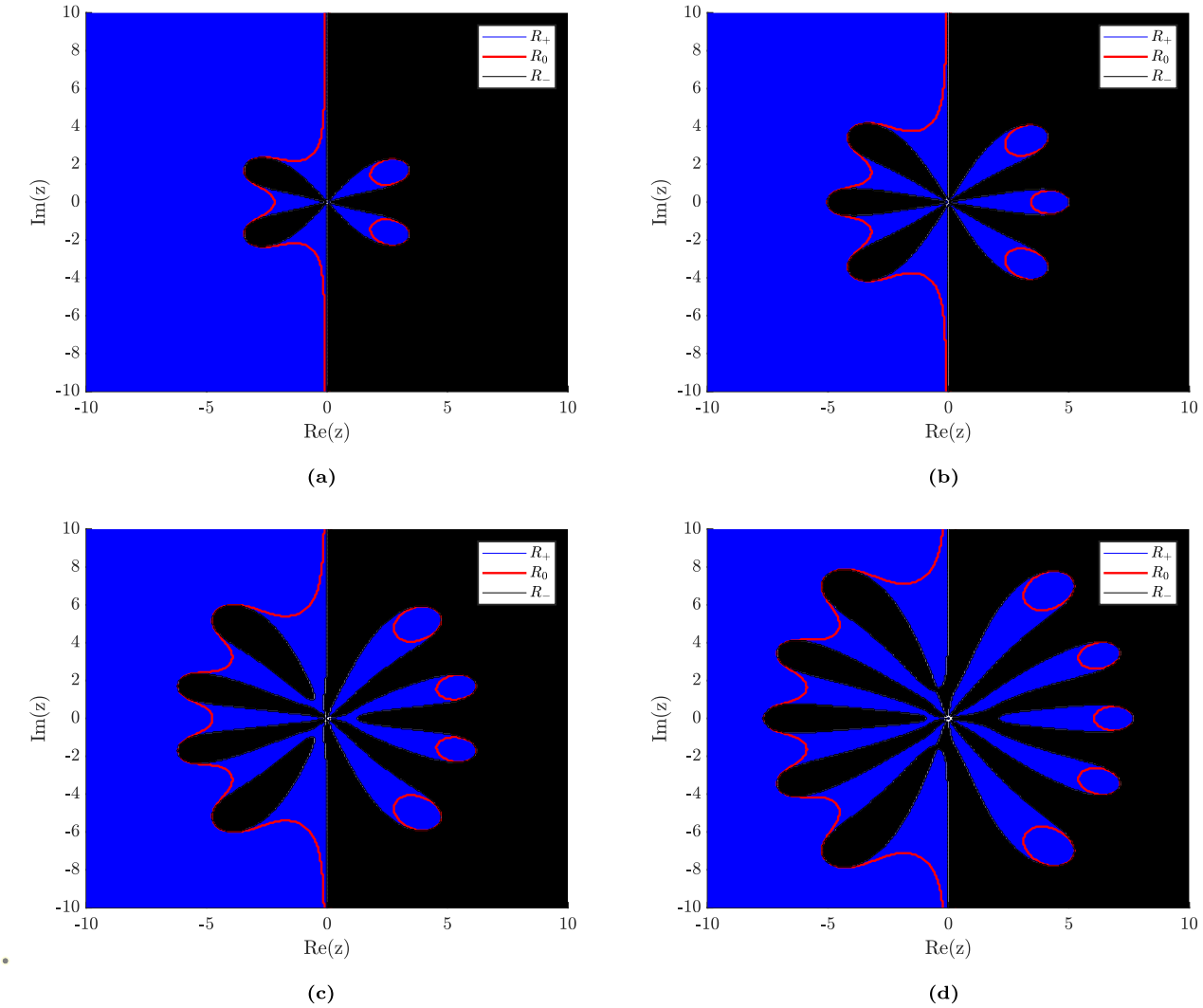

FIGURE 5 | The plot of the order stars regions for various values of m. (a) $m = 2$, (b) $m = 3$, (c) $m = 4$, (d) $m = 5$.

4 | Error Estimation and Control Strategy

To evaluate the convergence of the OBHM-SSIM method, we let u and $\mathbf{U}$ denote the exact and approximate solutions, respectively, and compute the absolute error (AE) and the error norm (maximum error L_∞) between the exact and numerical solutions as

$$AE = \left|u - \mathbf{U}\right|, \quad L_\infty = \|u - \mathbf{U}\|_\infty. \tag{48}$$

Later, we used ML_∞ to represent the largest error norm. Furthermore, we calculated the numerical absolute error estimate (AEE) between two successive iterations, defined as

$$AEE = \left\| \mathbf{U}_{n+p}^{(s+1)} - \mathbf{U}_{n+p}^{(s)} \right\|_\infty. \tag{49}$$

To terminate the iterative procedure in the OBHM-SSIM method, we applied the following criteria: if the error AEE is less than the user-defined tolerance (Tol), the current iteration's solutions are considered satisfactory, and the OBHM-SSIM proceeds to the next block. If AEE is greater than or equal to Tol, the iteration count increases and the OBHM-SSIM continues within the same block. The method iterates within each block until the absolute error estimate falls below the user-defined tolerance. Once the error converges within an acceptable criterion, the process moves to the next block or terminates the computation. The OBHM-SSIM algorithm is summarized in Algorithm 1.

ALGORITHM 1 | The OBHM-SSIM algorithm.

Input: $\frac{\partial u}{\partial t} = f(\mathbf{u})$, Spatial domain $[a, b]$, Temporal domain $[0, t_F]$, Tolerance Tol, Optimal points p_j, Step-size h, Number of the collocation point N_x, Initial and boundary conditions.

Output: Approximate solution $u(t, x)$.

Step 1: Spatial discretization

- Define Chebyshev-Gauss-Lobatto points: $\hat{x}_i = \cos\left(\frac{\pi i}{N_x}\right), i = 0, 1, \ldots, N_x$.
- Transform $x \in [a, b]$ to $[-1, 1]$ using equation (4).

Step 2: Temporal discretization

- Divide $t \in [0, t_F]$ into N blocks.
- Define step-size $h = t_{n+1} - t_n$ and $t_{n+p_j} = t_n + p_j h$.

Step 3: Linearization

- Linearize f using equation (18).

Step 4: Spectral collocation method

- Approximate spatial derivatives using differentiation matrix D.

Step 5: Optimal block hybrid method

- Write the time derivative as in equation (23) and integrate from 0 to p_j to obtain U_{n+p_j}.
- Express the system in matrix form equation (29).

Step 6: Initialize iteration

- Initialize iteration $s = 1$.

Step 7: Solve the system

- Solve the system of equations obtained from collocation equation (31).

Step 8: Check convergence

- Compute absolute error estimate between consecutive iterations using equation (49).
- If $AEE <$ Tol, proceed to the next time step.
- Otherwise, repeat Steps 5–7 until convergence.

Step 9: Update the solution

- Update U_{n+p_j} and proceed to the next time block.

Step 10: Repeat until final time

- Iterate over all time blocks until the final time t_F is reached.

Step 11: Output results

- Return the numerical solution $u(t, x)$ at all spatial and temporal points.

5 | Numerical Results and Discussions

In this section, we present the numerical results obtained by implementing the proposed OBHM-SSIM method on the N-EPDEs. Throughout the investigation, we used the optimized block hybrid method with $m = 5$, and the algorithm was implemented using MATLAB R2021b.

5.1 | Example 1

Consider the Stokes' second problem

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, b], \quad t \in [0, t_F]. \tag{50}$$

The analytical solution of the Stokes' second problem is given by [62]

$$u(t, x) = U_0\, e^{-\sqrt{\frac{\lambda}{2\nu}}x}\, \cos\left(\lambda t - \sqrt{\frac{\lambda}{2\nu}}x\right),$$

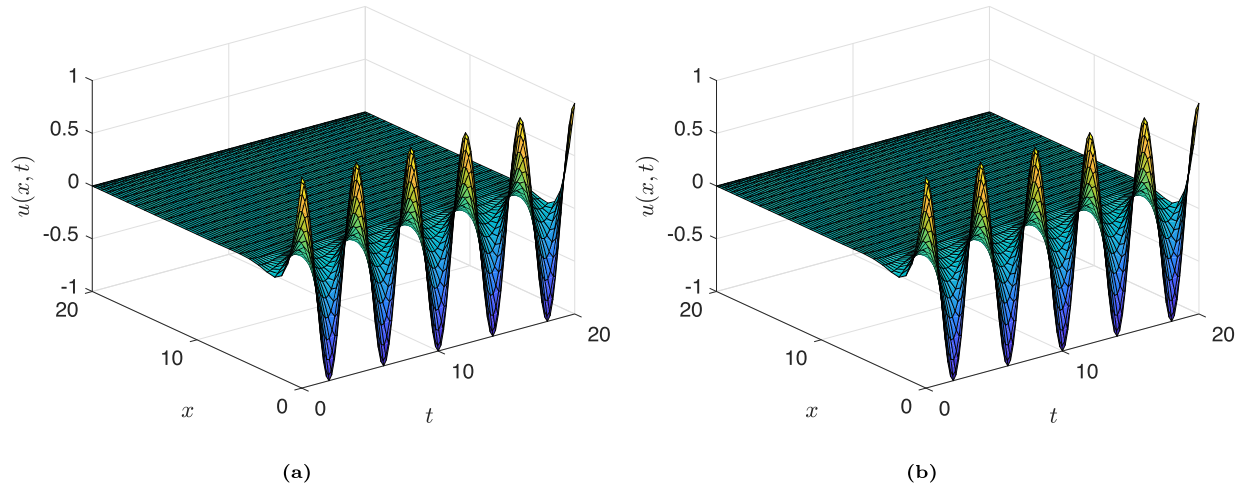

FIGURE 6 | Numerical and exact solutions for $\lambda = \frac{\pi}{2}, \nu = 1, U_0 = 1, Tol = 10^{-14}, t_F = 20, b = 20, N_x = 30$, and $h = 0.2$ in Example 1. (a) Exact solution (b) Numerical solution.

where U_0 is the initial velocity and λ is the oscillating frequency. We solve Equation (50), subject to the initial and boundary conditions

$$u(0,x) = U_0\, e^{-\sqrt{\frac{\lambda}{2\nu}}x}\, \cos\left(\sqrt{\frac{\lambda}{2\nu}}x\right),$$

$$u(t,0) = U_0\, \cos(\lambda t), \quad u(t,b) = U_0\, e^{-\sqrt{\frac{\lambda}{2\nu}}b}\, \cos\left(\lambda t - \sqrt{\frac{\lambda}{2\nu}}b\right).$$

Implementing the OBHM-SSIM, we define

$$\overset{(s+1)}{f} = \nu \overset{(s+1)}{\frac{\partial^2 u}{\partial x^2}},$$

where $\mathscr{L}_2 = 1,\ \mathscr{L}_1 = 0,\ \mathscr{L}_0 = 0$, and $\mathscr{H} = 0$.

Figure 6 presents a comparison between the exact and numerical solutions for the case where $\lambda = \frac{\pi}{2}, \nu = 1, U_0 = 1, Tol = 10^{-14}, t_F = 20, b = 20, N_x = 30$, and $h = 0.2$ for Example 1. Figure 21a shows the numerical solution obtained under the same conditions as the exact solution in Figure 21b. The numerical results exhibit a high level of agreement with the exact solution, accurately capturing the periodic oscillations and overall wave behavior. These results confirm the capability of the OBHM-SSIM method to produce highly accurate approximate solutions.

Figure 7 presents the absolute error behavior for different values of the spatial discretization points N_x and time step-sizes h when $Tol = 10^{-14}, t_F = 20$, and $b = 20$ for Example 1. Figure 7a shows the absolute error for three values of N_x (10, 20, and 30) with a fixed time step-size $h = 0.2$. The results indicate that increasing N_x significantly reduces the absolute error: for $N_x = 10$, the error is around $\mathcal{O}(10^{-5})$; for $N_x = 20$, it decreases to approximately $\mathcal{O}(10^{-10})$; and for $N_x = 30$, it further drops to the order of $\mathcal{O}(10^{-15})$. Figure 7b illustrates the effect of varying the time step-size h (1, 0.5, and 0.25) while keeping $N_x = 30$. The plot demonstrates that reducing h improves accuracy, with the absolute error decreasing from $\mathcal{O}(10^{-5})$ at $h = 1$ to $\mathcal{O}(10^{-15})$ at $h = 0.25$. Overall, both a smaller step-size and a larger number of spatial points enhance the accuracy and stability of the OBHM-SSIM method.

Figure 8 demonstrates the convergence behavior of the OBHM-SSIM per block for different values of the initial velocity U_0 and the parameter λ, with $\nu = 1, N_x = 30, h = 0.2, Tol = 10^{-14}, t_F = 20$, and $b = 20$ for Example 1. Figure 8a shows the iterations per block for various values of U_0, where for $U_0 = 5, 10$, and 15, the convergence is achieved in only two iterations per block. Figure 8b illustrates the effect of λ on the convergence rate. For $\lambda = 0$, the number of iterations fluctuates between one and two, while for $\lambda = \frac{\pi}{2}$ and $\lambda = \pi$, convergence is consistently achieved within two iterations. These results confirm the efficiency and robustness of the OBHM-SSIM, demonstrating its stable convergence characteristics across different initial velocities and oscillation frequency parameters.

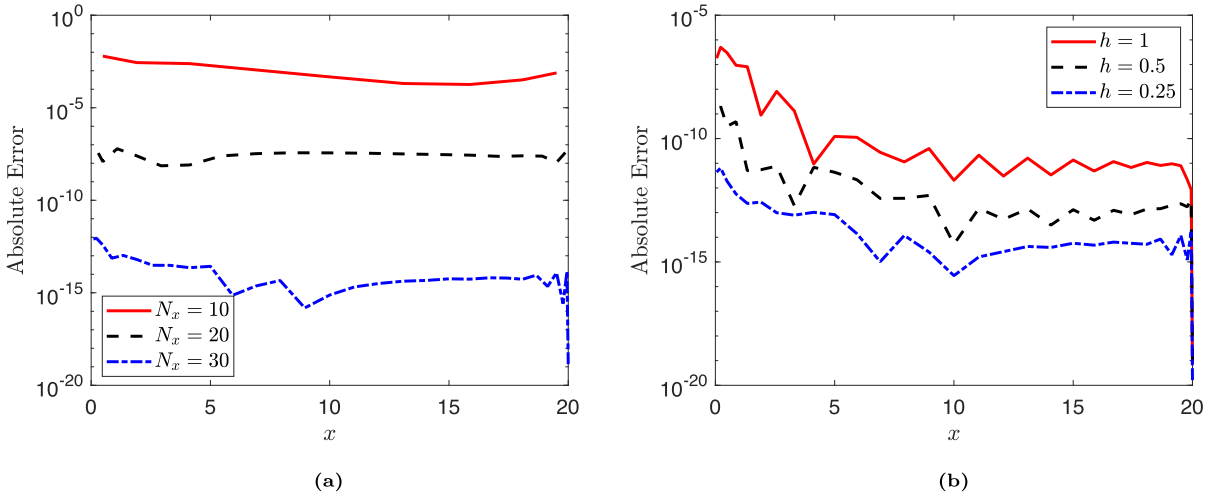

FIGURE 7 | Absolute error for different values of N_x and h with $Tol = 10^{-14}$, $t_F = 20$, and $b = 20$ for Example 1. (a) Absolute error for $h = 0.2$. (b) Absolute error for $N_x = 30$.

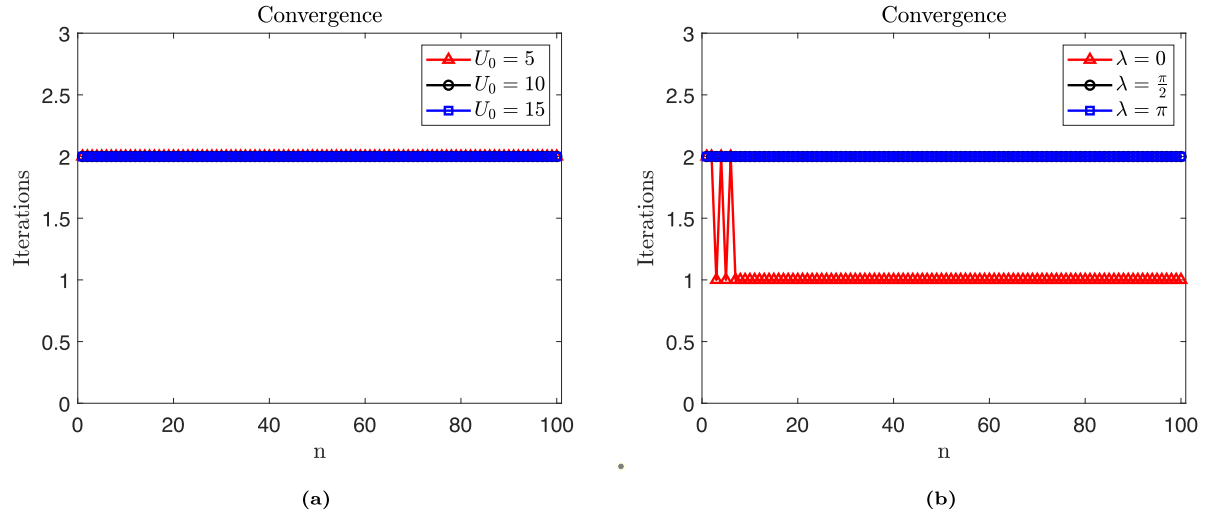

FIGURE 8 | Convergence of the OBHM-SSIM per block for different values of U_0 and λ with $\nu = 1$, $N_x = 30$, $h = 0.2$, $Tol = 10^{-14}$, $t_F = 20$, and $b = 20$ for Example 1. (a) Iterations per block for various values of U_0. (b) Iterations per block for various values of λ.

TABLE 4 | Comparison of the norm errors and CPU times using the OBHM-SSIM and the Crank-Nicolson spectral collocation method for Example 1.

Method	ν	λ	U_0	a	b	t_F	N_x	h	Tol	L_∞	CPU time in seconds
CNSCM	1	$\frac{\pi}{2}$	1	0	1	1	10	10^{-4}	—	1.271038×10^{-10}	0.0627
OBHM-SSIM								10^{-1}	10^{-4}	6.420454×10^{-13}	0.025702
CNSCM	1	π	5	0	1	2	10	10^{-4}	—	3.630568×10^{-9}	0.1193
OBHM-SSIM								10^{-1}	10^{-4}	5.8467×10^{-10}	0.027246

Table 4 presents a comparison between the proposed OBHM-SSIM and the Crank–Nicolson spectral collocation method (CNSCM) for Example 1. In CNSCM, the Crank–Nicolson scheme is employed for temporal discretization, while the spatial derivatives are approximated using Chebyshev–Gauss–Lobatto collocation points [63]. The results indicate that OBHM-SSIM consistently achieves smaller L_∞ errors while requiring significantly less CPU time. By contrast, CNSCM attains acceptable accuracy only when a much smaller time step-size h is used, which substantially increases its computational cost. Overall, OBHM-SSIM demonstrates superior efficiency by delivering higher accuracy with reduced CPU time.

5.2 | Example 2

Consider the generalized Burger-Fisher equation

$$\frac{\partial u}{\partial t} + \alpha u^{\delta}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta}), \quad x \in [0,b], \quad t \in [0,t_F], \tag{51}$$

where α, δ, and β are constant parameters [64]. Here we define f, in the proposed OBHM-SSIM to be

$$\overset{(s+1)}{f} = \frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2} - \alpha \overset{(s)}{u^{\delta}}\frac{\partial \overset{(s+1)}{u}}{\partial x} + \beta\left(1-\overset{(s)}{u^{\delta}}\right)\overset{(s+1)}{u},$$

where $\mathscr{L}_2 = 1$, $\mathscr{L}_1 = -\alpha \overset{(s)}{u^{\delta}}$, $\mathscr{L}_0 = \beta\left(1-\overset{(s)}{u^{\delta}}\right)$, and $\mathscr{H} = 0$.

- **Case 1**, when $\alpha = 0$, and $\delta = 1$, the generalized Burger-Fisher Equation (51) is reduced to a Fish-erâĂŤKolmogorov–Petrovsky–Piskunov (KPP) equation with the following initial and boundary conditions

$$u(0,x) = \left(1+e^{\sqrt{\frac{\beta}{6}}x}\right)^{-2}, \quad u(t,0) = \left(1+e^{-\frac{5\beta}{6}t}\right)^{-2}, \quad u(t,b) = \left(1+e^{\sqrt{\frac{\beta}{6}}b-\frac{5\beta}{6}t}\right)^{-2}.$$

  Thus, KPP equation has an exact solution of the form [65]

$$u(t,x) = \left(1+e^{\sqrt{\frac{\beta}{6}}x-\frac{5\beta}{6}t}\right)^{-2}.$$

  Now solving equation (51), for case 1. We present the numerical and exact solutions in Figure 9. Figure 9 shows that the graphs of the approximate and exact solutions coincide for the chosen values of $\beta = 2,3,4$. We noted that increasing β increases the rate of reaction leading to faster finish of the displacement u.

  In Figure 10, we show the absolute error, error norm, and iterations per block for the OBHM-SSIM method. Figure 10a depicts the absolute error versus the collocation points in space, while Figure 10b shows the absolute error versus the collocation points in space for different temporal step-sizes. The results provide evidence that the OBHM-SSIM

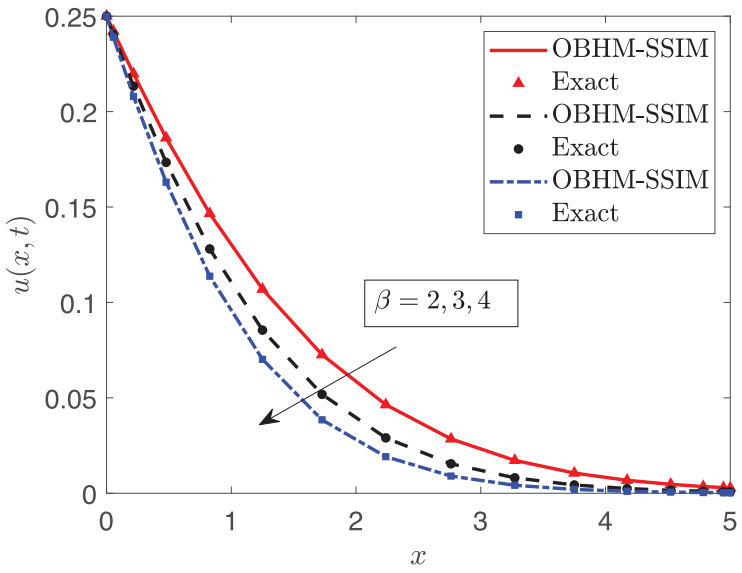

FIGURE 9 | Numerical versus exact solutions for various values of β when $Tol = 10^{-14}$, $t_F = 10$, $b = 5$, $N_x = 15$, and $h = 10^{-2}$ for Example 2 (Case 1).

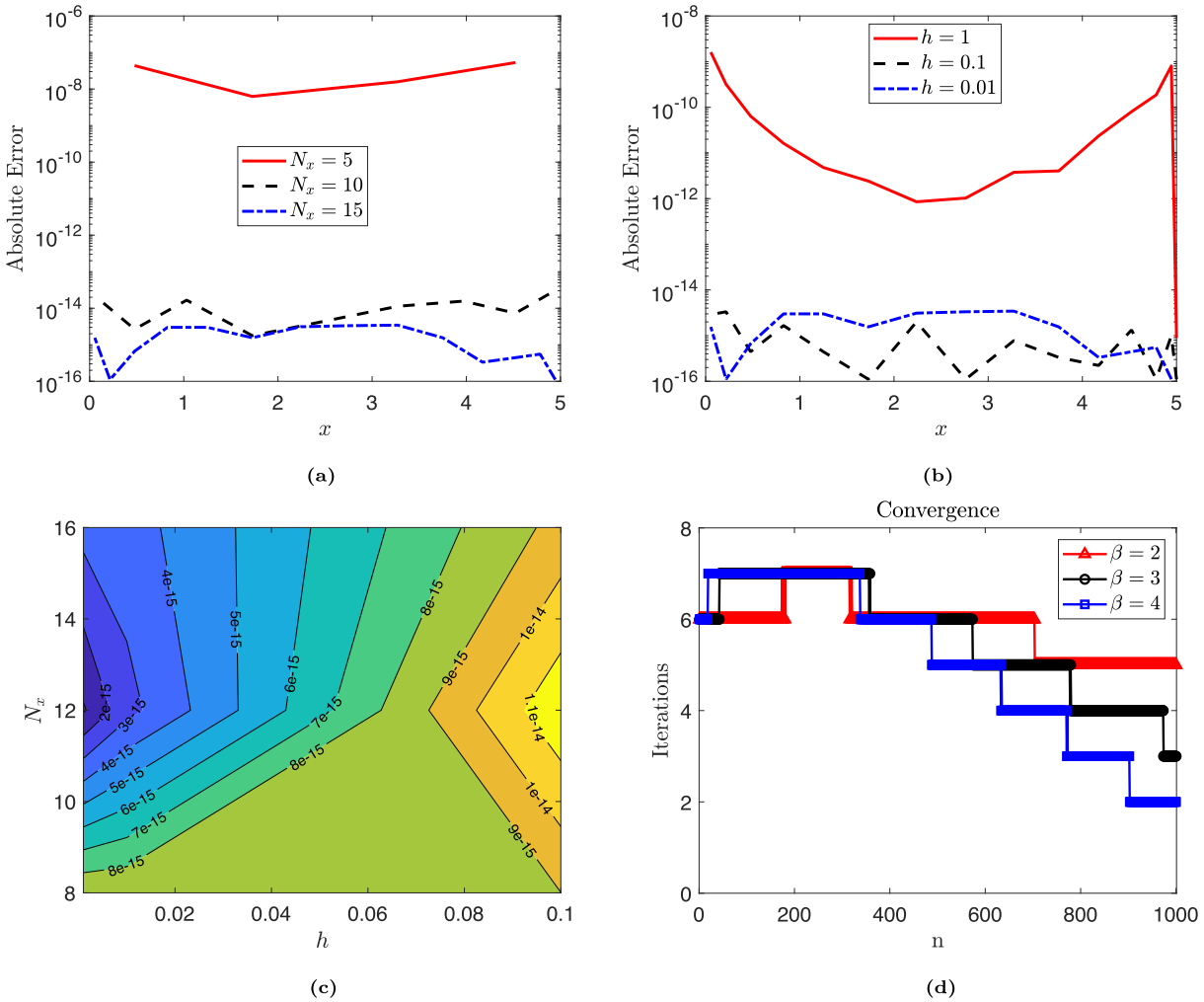

FIGURE 10 | Absolute error, error norm, and convergence of the OBHM-SSIM with $Tol = 10^{-14}$, $b = 5$, $N_x = 15$, and $h = 10^{-2}$ for Example 2 (Case 1). (a) Absolute error for $\beta = 1$ at $t_F = 10$. (b) Absolute error for $\beta = 1$ at $t_F = 10$. (c) Error norms for $\beta = 1$ and $b = 1$ at $t_F = 3$. (d) Iterations per block for various values of β at $t_F = 10$.

achieves good absolute error at $N_x = 15$ and $h = 10^{-2}$. In Figure 10c, a contour plot of the error norm is included to show how the error varies with both N_x and h. The contours indicate that the error decreases with increasing N_x and smaller h, confirming the spectral accuracy in space and stable temporal behavior of the method. Figure 10d shows the iterations per block for various values of β at $N_x = 15$ and $h = 10^{-2}$. For a larger time domain at $t_F = 10$ and different values of β, as illustrated in Figure 10d with 1000 blocks, the maximum number of iterations remains at seven for the OBHM-SSIM. This consistency in the number of iterations indicates the robustness and efficiency of the OBHM-SSIM.

Table 5 compares the error norms obtained using the OBHM-SSIM method with various spectral methods, all employing the same number of spatial collocation points [26, 27]. Motsa et al. [26] employed the BSQLM for both the spatial and temporal domains, while Magagula et al. [27] used the L-BSQLM. It is noteworthy that the L-BSQLM of Magagula et al. [27] employs a Legendre–Gauss–Lobatto points for the spatial discretization. The results indicate that by modestly increasing the spatial collocation points to $N_x = 15$, the CGL-based OBHM-SSIM achieves an accuracy of order $\mathcal{O}(10^{-13})$, significantly outperforming the Legendre-based method. As shown in Table 5, the OBHM-SSIM method achieves its largest error norm of order $\mathcal{O}(10^{-13})$ for $t_F = 10$, and $\mathcal{O}(10^{-15})$ for $t_F = 2$. For a larger time interval ($t_F = 20$), the accuracy improves with increasing N_x, while for a shorter interval ($t_F = 2$), using a smaller step-size further enhances accuracy. Table 5 demonstrates the high accuracy and computational efficiency of the OBHM-SSIM method. The reported CPU times provide additional insight into the computational cost for different values of N_x and h, further reflecting the method's efficiency.

TABLE 5 | Comparison of error norms and CPU times using OBHM-SSIM, Magagula et al. [27], and Motsa et al. [26] at selected value of t, h, and N_x for $Tol = 10^{-14}$ and $\beta = 1$ for Example 2 (Case 1).

	$b = 5, t_F = 10$				$b = 1, t_F = 2$		
		OBHM-SSIM	OBHM-SSIM			OBHM-SSIM	OBHM-SSIM
		$h = 10^{-1}$	$h = 10^{-1}$			$h = 10^{-1}$	$h = 10^{-2}$
t	L-BSQLM [27]	$N_x = 10$	$N_x = 15$	t	BSQLM [26]	$N_x = 10$	$N_x = 10$
1	7.19×10^{-10}	2.081×10^{-9}	2.284×10^{-13}	0.2	3.808×10^{-14}	2.331×10^{-15}	3.331×10^{-16}
2	6.82×10^{-10}	1.862×10^{-9}	2.103×10^{-13}	0.4	3.780×10^{-14}	1.554×10^{-15}	6.106×10^{-16}
3	2.86×10^{-10}	9.742×10^{-10}	3.086×10^{-14}	0.6	7.283×10^{-14}	3.775×10^{-15}	4.996×10^{-16}
4	5.21×10^{-11}	3.273×10^{-10}	4.330×10^{-15}	0.8	3.714×10^{-14}	2.276×10^{-15}	5.551×10^{-16}
5	3.77×10^{-12}	3.894×10^{-11}	1.776×10^{-15}	1.0	1.691×10^{-13}	3.886×10^{-15}	7.216×10^{-16}
6	2.23×10^{-12}	1.191×10^{-11}	2.776×10^{-15}	1.2	3.119×10^{-13}	3.664×10^{-15}	6.661×10^{-16}
7	1.37×10^{-13}	9.425×10^{-13}	3.220×10^{-15}	1.4	1.796×10^{-13}	2.220×10^{-15}	9.992×10^{-16}
8	5.40×10^{-14}	6.201×10^{-13}	4.219×10^{-15}	1.6	1.097×10^{-13}	2.109×10^{-15}	1.110×10^{-15}
9	1.80×10^{-14}	1.550×10^{-13}	2.109×10^{-15}	1.8	6.273×10^{-13}	3.442×10^{-15}	1.554×10^{-15}
10	4.44×10^{-15}	2.442×10^{-14}	1.887×10^{-15}	2.0	2.311×10^{-13}	1.665×10^{-15}	2.887×10^{-15}
ML_∞	7.19×10^{-10}	2.081×10^{-9}	2.284×10^{-13}	—	6.273×10^{-13}	3.886×10^{-15}	2.887×10^{-15}
CPU	—	0.318047	0.490600	—	—	0.070884	0.364717

- **Case 2,** For $\alpha = \delta = \beta = 1$ in Equation (51), we recover the combined aspects of the Burgers equation and the Fisher-KPP equation, with the following initial and boundary conditions

$$u(0,x) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1}{4}x\right), \quad u(t,0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{a}{4} - \frac{5}{8}t\right), \quad u(t,b) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{b}{4} - \frac{5}{8}t\right).$$

The Burgers equation and the Fisher-KPP equation in Case 2 has an exact solution of the following form [66]

$$u(t,x) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1}{4}x - \frac{5}{8}t\right).$$

Thus, the approximate and exact solution is presented in Figure 11. The figure demonstrates close alignment, which indicates the high accuracy of the method.

The absolute error, error norm, and iterations per block for different values of N_x are illustrated in Figure 12 for Example 2, case 2. Figure 12a depicts that the method achieves a maximum absolute error of less than $\mathcal{O}(10^{-13})$ when $N_x = 10$, and Figure 12b shows a maximum absolute error of less than $\mathcal{O}(10^{-15})$ when $N_x = 15$. These results indicate that the OBHM-SSIM method provides accurate solutions and maintains a low maximum absolute error as the number of grid points increases to $N_x = 15$. Figure 12c illustrates the variation of the error norm with respect to the spatial collocation points N_x and the temporal step-size h for $\beta = 1$ and $b = 1$ at $t_F = 3$. The contours show that the error decreases with increasing N_x and smaller h. Figure 12d shows that the maximum number of iterations required within the computational block is five for all tested values of N_x, highlighting the computational efficiency of the OBHM-SSIM.

In Table 6, we compare the error norms of the OBHM-SSIM method with various variants of the spectral method [26–28], where all use the same number of collocation points in the spatial domain, $N_x = 10$. Khumalo et al. [28] employed the BSLPM for both the spatial and temporal domains. It is evident that the OBHM-SSIM consistently achieves a higher accuracy order of $\mathcal{O}(10^{-15})$ than the spectral methods developed in [26, 28] when all utilize Chebyshev–Gauss–Lobatto grid points. In the case of $b = 5$ and $t_F = 10$, the OBHM-SSIM and the spectral method in [27] achieved the same largest error norm of order $\mathcal{O}(10^{-9})$, while using different types of grid points. It is noted that when the number of collocation points was increased to $N_x = 14$, the largest error of order $\mathcal{O}(10^{-12})$, the OBHM-SSIM outperformed the L-BSQLM method. We also noted the superior accuracy of the OBHM-SSIM compared to the BSQLM and BSLPM methods. The CPU time results in Table 6 further demonstrate the efficiency of the OBHM-SSIM method, which achieves shorter computation times than the BSLPM and BSQLM methods, confirming its superior computational performance.

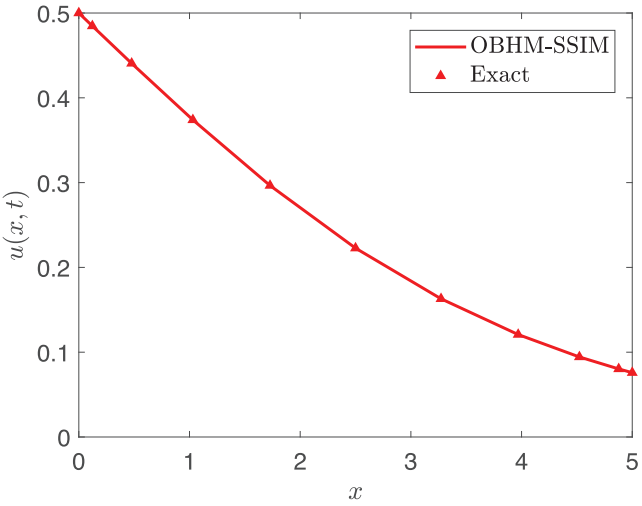

FIGURE 11 | Comparison of the numerical and exact solution of the OBHM-SSIM for Example 2 (Case 2) with $N_x = 15$, $Tol = 10^{-12}$, $t_F = 10$, $b = 5$, and $h = 10^{-2}$.

5.3 | Example 3

Consider the Burger's–Huxley equation,

$$\frac{\partial u}{\partial t} + \alpha u^{\delta}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta})(u^{\delta}-\gamma), \quad x \in [0,b], \quad t \in [0,t_F], \tag{52}$$

where α, δ, β, and γ are constant parameters [67]. Subject to initial and boundary conditions

$$u(0,x) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x}{2}\right), \quad u(t,0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{(\gamma-1)t}{2}\right), \quad u(t,b) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1}{2}[b+(\gamma-1)t]\right).$$

Using the OBHM-SSIM, we define

$$\overset{(s+1)}{f} = \frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2} - \alpha \overset{(s)}{u}{}^{\delta}\frac{\partial \overset{(s+1)}{u}}{\partial x} + \beta\left(1-\overset{(s)}{u}{}^{\delta}\right)\left(\overset{(s)}{u}{}^{\delta}-\gamma\right)\overset{(s+1)}{u},$$

where $\mathscr{L}_2 = 1$, $\mathscr{L}_1 = -\alpha \overset{(s)}{u}{}^{\delta}$, $\mathscr{L}_0 = \beta\left(1-\overset{(s)}{u}{}^{\delta}\right)\left(\overset{(s)}{u}{}^{\delta}-\gamma\right)$, and $\mathscr{H} = 0$. Equation (52) has the exact solution of the form

$$u(t,x) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{\beta}{r-\alpha}[x-ct]\right), \quad \text{where} \quad r = \sqrt{\alpha^2+8\beta}, \quad c = \frac{(\alpha-r)(2\gamma-1)+2\alpha}{4}.$$

Setting $\delta = \alpha = \beta = 1$, and $\gamma \in (0,1)$, where γ represents a threshold that influences the reaction dynamics. Figure 13 compares numerical and exact solutions for $t_F = 5$, $\gamma = 0.1$, and $N_x = 15$. It is evident that the approximate solution is almost matching the exact solution for $\gamma = 0.1$.

In Figure 14, we present the absolute error and iterations per block for Example 3. Figure 14a shows the absolute error for $N_x = 10$, achieving a maximum absolute error of less than $\mathcal{O}(10^{-13})$. When the number of collocation points increases to $N_x = 15$, the maximum absolute error remains below $\mathcal{O}(10^{-15})$, indicating enhanced accuracy, as shown in Figure 14b. Figure 14c illustrates the variation of the error norm with respect to the spatial collocation points N_x and the temporal step-size h. Figure 14d shows that the maximum number of iterations required per computational block is six within 500 blocks for $\gamma = 0.1, 0.5$, and 0.75, demonstrating the method's computational efficiency. The OBHM-SSIM method shows improvements in absolute error, particularly with increased collocation points in the spatial domain, and consistently achieves highly accurate results.

The results presented in Table 7 compare the error norms of the OBHM-SSIM method with those obtained using other spectral methods, namely the BSLPM [28] and BSQLM [26], for different combinations of t_F, N_x, and γ. For $t_F = 1$ and

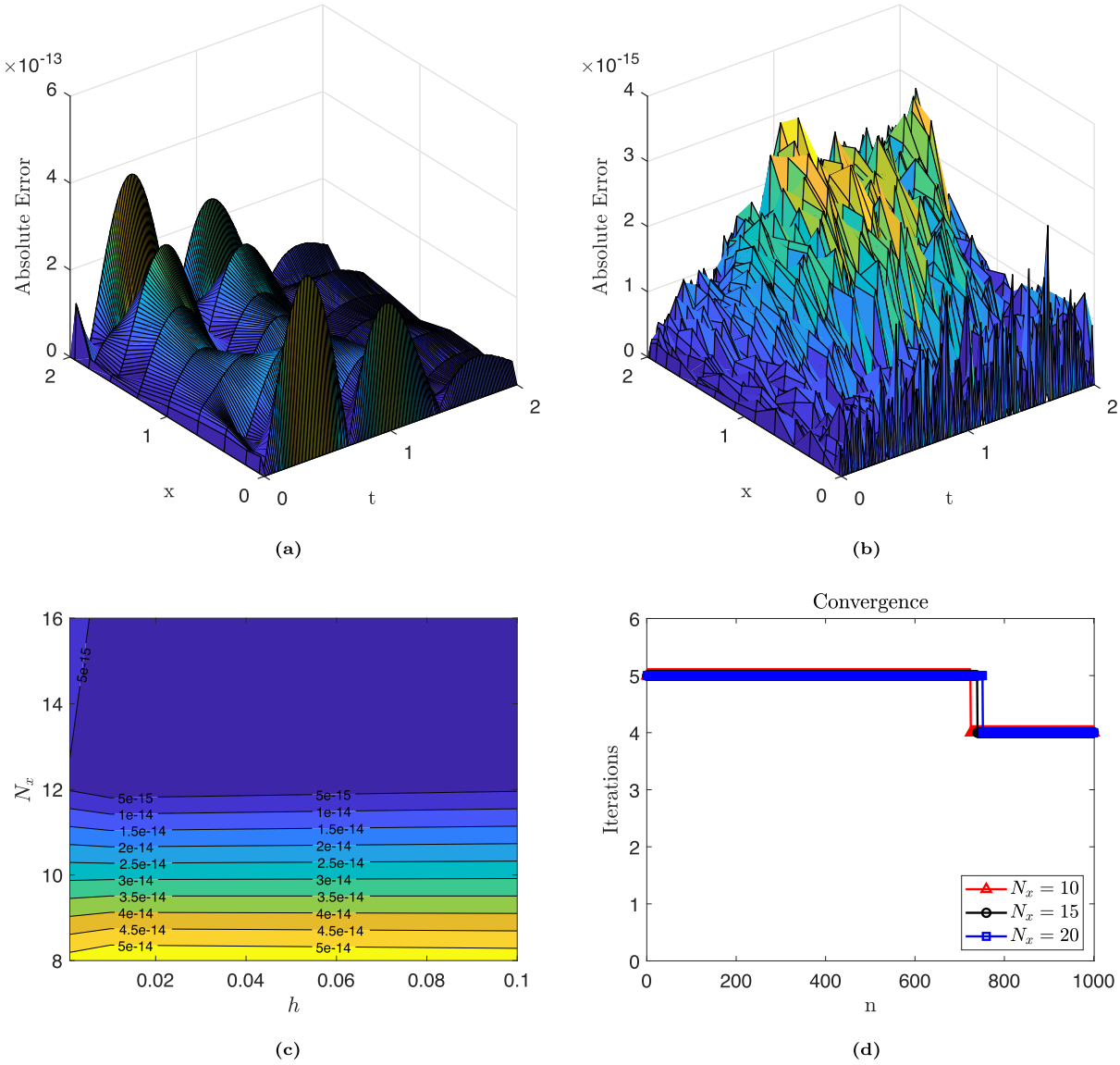

FIGURE 12 | Absolute error, error norm, and convergence of the OBHM-SSIM with $Tol = 10^{-12}$, $t_F = 2$, $b = 2$, and $h = 10^{-2}$ for Example 2 (Case 2). (a) Absolute error for $N_x = 10$, (b) Absolute error for $N_x = 15$, (c) Error norms for $Tol = 10^{-14}$, $t_F = 8$, $b = 3$, (d) Iterations per block for various N_x.

$\gamma = 0.1, 0.75$, the largest error norms produced by the OBHM-SSIM are of the order $\mathcal{O}(10^{-15})$, which are comparable to those obtained by the BSLPM and BSQLM methods. For $t_F = 2$ and $\gamma = 0.5$, the OBHM-SSIM maintains consistently lower error across all time points than those achieved by the BSQLM method. These results confirm that the OBHM-SSIM method is highly effective in producing low error norms, highlighting its robustness and accuracy. Furthermore, the consistent performance across different values of t_F and γ demonstrates the method's adaptability and precision under varying parameter values. The CPU time results in Table 7 demonstrate the superior efficiency of the OBHM-SSIM method, which requires less computation time than the BSLPM and BSQLM.

5.4 | Example 4

The FitzHugh–Nagumo equation has various applications in the fields of flame propagation, logistic population growth, neurophysiology, branching Brownian motion processes, autocatalytic chemical reactions, and nuclear reactor theory [68]. The generalized FitzHugh–Nagumo equation with time-dependent coefficients studied by [69] is

$$\frac{\partial u}{\partial t} + \nu(t)\frac{\partial u}{\partial x} = \mu(t)\frac{\partial^2 u}{\partial x^2} + \eta(t)u(1-u)(\rho - u), \quad x \in [a, b], \quad t \in [0, t_F], \tag{53}$$

TABLE 6 | Comparison of error norms and CPU times using OBHM-SSIM, Magagula et al. [27], Khumalo et al. [28], and Motsa et al. [26] at selected value of t when $h = 10^{-1}$ and $Tol = 10^{-14}$ for Example 2 (Case 2).

	$b = 5, t_F = 10$				$b = 1, t_F = 1$		
t	L-BSQLM [27]	OBHM-SSIM $N_x = 10$	OBHM-SSIM $N_x = 14$	t	BSLPM [28]	BSQLM [26]	OBHM-SSIM $N_x = 10$
1	3.68×10^{-9}	6.398×10^{-9}	5.937×10^{-12}	0.1	5.695×10^{-14}	6.143×10^{-12}	5.551×10^{-16}
2	7.78×10^{-10}	2.333×10^{-9}	2.439×10^{-12}	0.2	3.431×10^{-14}	1.013×10^{-11}	4.441×10^{-16}
3	4.55×10^{-11}	2.298×10^{-10}	1.875×10^{-13}	0.3	1.321×10^{-13}	1.512×10^{-11}	6.661×10^{-16}
4	3.14×10^{-12}	5.874×10^{-11}	9.659×10^{-15}	0.4	3.217×10^{-13}	1.702×10^{-11}	1.110×10^{-15}
5	4.31×10^{-12}	8.613×10^{-12}	1.998×10^{-15}	0.5	1.852×10^{-13}	5.736×10^{-12}	1.776×10^{-15}
6	5.53×10^{-13}	1.041×10^{-12}	2.442×10^{-15}	0.6	8.038×10^{-14}	1.626×10^{-11}	6.661×10^{-16}
7	1.37×10^{-14}	8.882×10^{-14}	1.887×10^{-15}	0.7	1.361×10^{-13}	1.364×10^{-11}	7.772×10^{-16}
8	1.20×10^{-14}	3.220×10^{-14}	2.554×10^{-15}	0.8	4.013×10^{-13}	3.852×10^{-12}	1.443×10^{-15}
9	4.66×10^{-15}	1.343×10^{-14}	1.776×10^{-15}	0.9	2.000×10^{-13}	4.727×10^{-12}	8.882×10^{-16}
10	4.22×10^{-15}	3.775×10^{-15}	1.221×10^{-15}	1.0	4.089×10^{-13}	7.261×10^{-13}	7.772×10^{-16}
ML_∞	3.68×10^{-9}	6.398×10^{-9}	5.937×10^{-12}	—	3.217×10^{-13}	1.702×10^{-11}	1.776×10^{-15}
CPU	—	0.303803	0.345517	—	0.175061	0.021564	0.046416

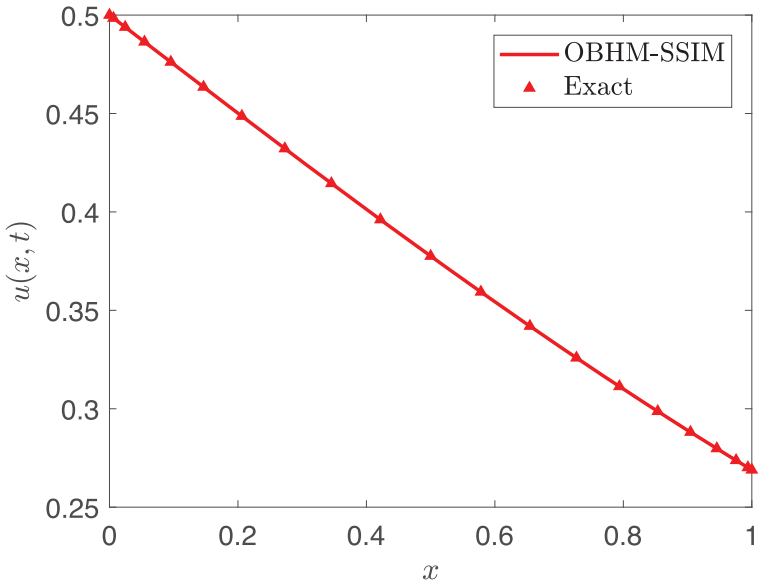

FIGURE 13 | Numerical and exact solutions when $b = 1$, $Tol = 10^{-12}$, $h = 10^{-2}$ $t_F = 5$, $N_x = 15$, at $\gamma = 0.1$ for Example 3.

where ρ is a constant parameter, subject to the initial and boundary conditions

$$u(0,x) = \frac{\rho}{2} + \frac{\rho}{2}\tanh\left(\frac{\rho}{2}x\right),$$
$$u(t,a) = \frac{\rho}{2} + \frac{\rho}{2}\tanh\left(\frac{\rho}{2}[a - (3-\rho)\sin(t)]\right), \quad u(t,b) = \frac{\rho}{2} + \frac{\rho}{2}\tanh\left(\frac{\rho}{2}[b - (3-\rho)\sin(t)]\right).$$

Applying the OBHM-SSIM method to Equation (52) yields

$$\overset{(s+1)}{f} = \mu(t)\frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2} - \nu(t)\frac{\partial \overset{(s+1)}{u}}{\partial x} + \eta(t)(1 - \overset{(s)}{u})(\rho - \overset{(s)}{u})\overset{(s+1)}{u},$$

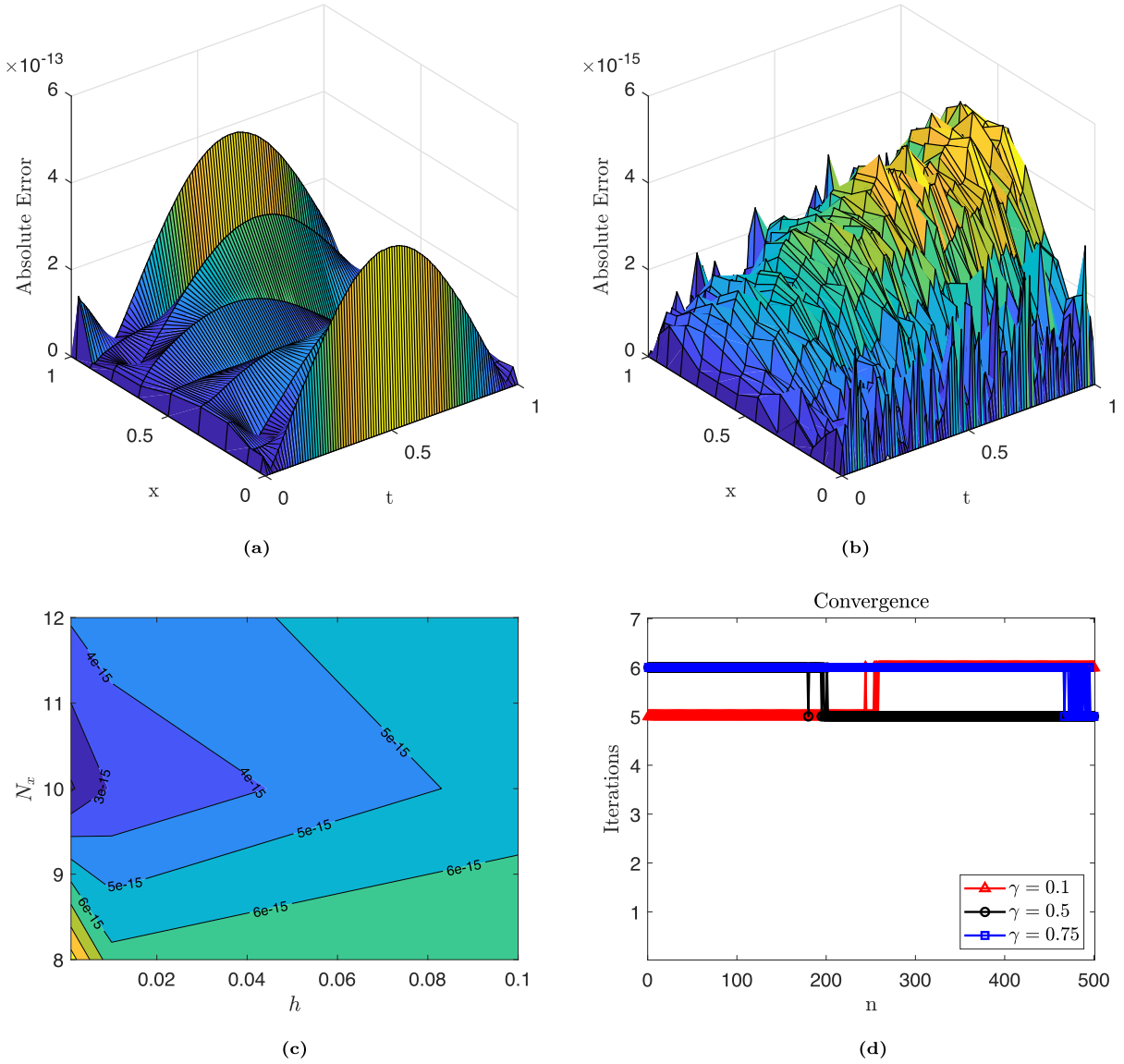

FIGURE 14 | Absolute error, error norm, and convergence of the OBHM-SSIM with $b = 1$, $Tol = 10^{-12}$, and $h = 10^{-2}$ for Example 3. (a) Absolute error for $t_F = 1$ and $N_x = 10$ at $\gamma = 0.1$. (b) Absolute error for $t_F = 1$ and $N_x = 15$ at $\gamma = 0.1$. (c) Error norm for $t_F = 8$, $b = 1$, and $Tol = 10^{-14}$ at $\gamma = 0.1$. (d) Iterations per block for $t_F = 5$ and $N_x = 15$ for various γ.

where $\mathscr{L}_2 = \mu(t)$, $\mathscr{L}_1 = -\nu(t)$, $\mathscr{L}_0 = -\eta(t)(1 - \overset{(s)}{u})(\rho - \overset{(s)}{u})$, and $\mathscr{H} = 0$. The time-dependent coefficients are defined as $\mu(t) = \nu(t) = \cos(t)$ and $\eta(t) = 2\cos(t)$. The exact solution of Equation (53), as given in [70], has the form

$$u(t,x) = \frac{\rho}{2} + \frac{\rho}{2}\tanh\Big(\frac{\rho}{2}[x - (3-\rho)\sin(t)]\Big).$$

Figure 15 presents the numerical and exact solutions for various values of ρ with $N_x = 15$. These results indicate that increasing ρ in the generalized FitzHugh–Nagumo equation enhances the displacement behavior of u.

Figure 16 presents the absolute error, error norm, and iterations per block for Example 4. Figures 16a and 16b show the absolute error when the number of collocation points in the spatial domain increases from $N_x = 15$ to $N_x = 20$. The maximum absolute error is less than $\mathcal{O}(10^{-13})$ for $N_x = 15$ and further decreases to below $\mathcal{O}(10^{-15})$ for $N_x = 20$, demonstrating that increasing N_x enhances the accuracy of the method. Figure 16c illustrates the variation of the error norm with respect to N_x and h. Figure 16d shows the iterations per block required for the OBHM-SSIM method to converge for different values of ρ, indicating that the maximum number of iterations over 100 blocks is consistently six. This stability confirms the computational efficiency and reliable convergence of the OBHM-SSIM method across various ρ values.

TABLE 7 | Comparison of error norms and CPU times using the OBHM-SSIM, BSLPM [28], and BSQLM [26] at selected values of t and various values of t_F and γ, for $b = 1$, $h = 10^{-1}$, and $Tol = 10^{-14}$ in Example 3.

	$t_F = 1, \gamma = 0.1$		$t_F = 1, \gamma = 0.75$			$t_F = 2, \gamma = 0.5$	
t	BSLPM [28]	OBHM-SSIM $N_x = 12$	BSQLM [26]	OBHM-SSIM $N_x = 12$	t	BSQLM [26]	OBHM-SSIM $N_x = 12$
0.1	9.232×10^{-14}	1.776×10^{-15}	7.822×10^{-14}	3.220×10^{-15}	0.2	1.150×10^{-12}	1.277×10^{-15}
0.2	2.790×10^{-13}	1.998×10^{-15}	1.184×10^{-13}	2.276×10^{-15}	0.4	1.638×10^{-12}	2.276×10^{-15}
0.3	2.850×10^{-13}	7.327×10^{-15}	1.049×10^{-13}	2.165×10^{-15}	0.6	1.958×10^{-12}	2.776×10^{-15}
0.4	4.696×10^{-13}	7.216×10^{-15}	9.426×10^{-14}	2.109×10^{-15}	0.8	7.002×10^{-13}	3.109×10^{-15}
0.5	3.957×10^{-13}	5.329×10^{-15}	1.510×10^{-13}	4.108×10^{-15}	1.0	1.267×10^{-12}	2.942×10^{-15}
0.6	3.748×10^{-13}	3.442×10^{-15}	2.127×10^{-13}	2.887×10^{-15}	1.2	1.710×10^{-12}	1.998×10^{-15}
0.7	4.136×10^{-13}	7.550×10^{-15}	1.230×10^{-13}	2.554×10^{-15}	1.4	5.109×10^{-13}	4.330×10^{-15}
0.8	4.852×10^{-13}	2.998×10^{-15}	1.549×10^{-13}	1.998×10^{-15}	1.6	8.203×10^{-13}	3.775×10^{-15}
0.9	1.885×10^{-13}	4.552×10^{-15}	3.063×10^{-13}	3.109×10^{-15}	1.8	8.294×10^{-13}	4.330×10^{-15}
1.0	6.568×10^{-13}	2.554×10^{-15}	2.951×10^{-13}	5.995×10^{-15}	2.0	9.726×10^{-14}	2.887×10^{-15}
ML_∞	6.568×10^{-13}	7.550×10^{-15}	2.951×10^{-13}	5.995×10^{-15}	—	1.958×10^{-12}	4.663×10^{-15}
CPU	0.155115	0.036427	0.032806	0.030770	—	0.032806	0.031448

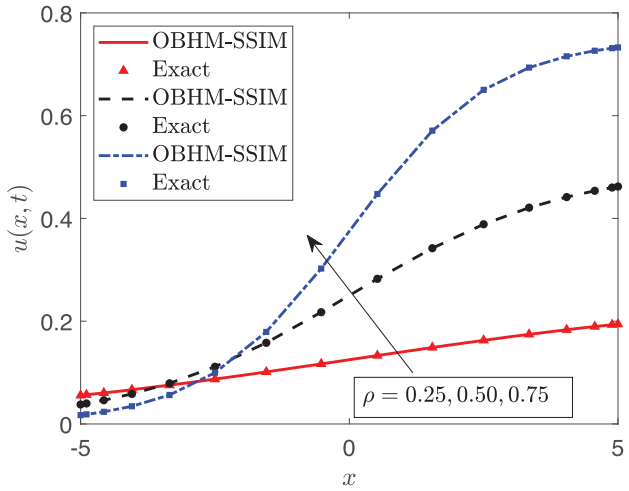

FIGURE 15 | Numerical versus exact solutions for various values of ρ when $N_x = 15$, $Tol = 10^{-14}$, $h = 10^{-2}$, $t_F = 1$, $a = -5$, and $b = 5$ for Example 4.

Table 8 shows a comparison of the maximum absolute errors between the OBHM-SSIM and the Jacobi–Gauss–Lobatto collocation method (JGLCM) [69]. Bhrawy et al. [69] used the Jacobi–Gauss–Lobatto points to approximate the spatial derivatives and reduce the equation to a system of ODEs in time to solve the FitzHugh–Nagumo equation with time-dependent coefficients numerically. As shown in Table 8, it is observed that the OBHM-SSIM achieves maximum absolute errors of the order $\mathcal{O}(10^{-14})$ and $\mathcal{O}(10^{-8})$, which are lower than those obtained by the JGLCM.

5.5 | Example 5

To implement the OBHM-SSIM in solving a nonlinear system of partial differential equations, let us consider the following system of equations

$$\frac{\partial u}{\partial t} = f_1(\mathbf{u}, \mathbf{v}, \cdots, \mathbf{w}), \quad \frac{\partial v}{\partial t} = f_2(\mathbf{u}, \mathbf{v}, \cdots, \mathbf{w}), \quad \cdots, \quad \frac{\partial w}{\partial t} = f_v(\mathbf{u}, \mathbf{v}, \cdots, \mathbf{w}), \tag{54}$$

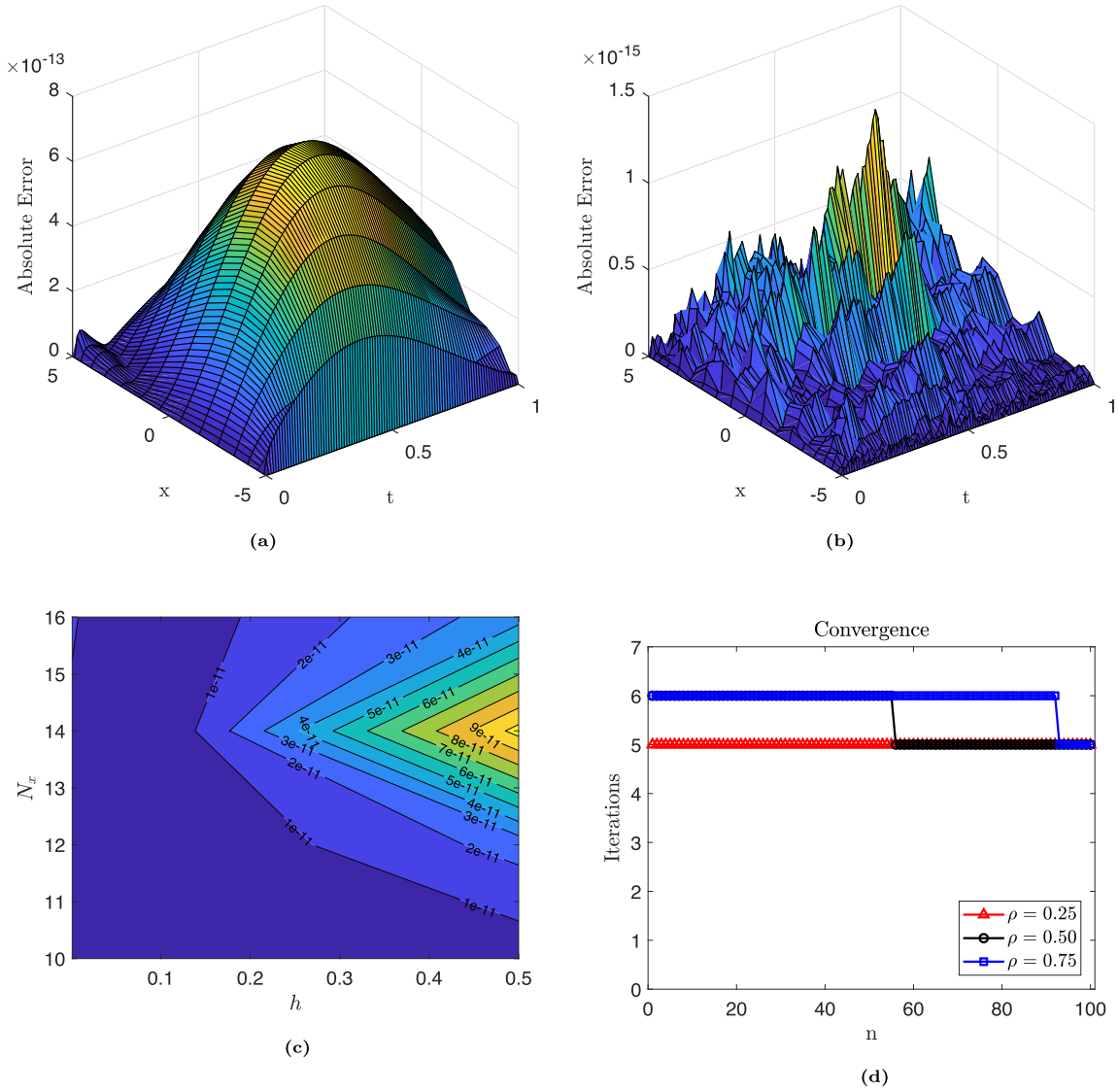

FIGURE 16 | Absolute error, error norm, and convergence of the OBHM-SSIM for $Tol = 10^{-14}$, $h = 10^{-2}$, $t_F = 1$, $a = -5$, and $b = 5$ in Example 4. (a) Absolute error for $N_x = 15$. (b) Absolute error for $N_x = 20$. (c) Error norm for $a = -5$, $b = 5$, $t_F = 2$, $Tol = 10^{-14}$, and $\rho = 0.1$. (d) Iterations per block for $N_x = 15$ and various ρ.

TABLE 8 | Comparison of maximum absolute errors and CPU times using the OBHM-SSIM and the Jacobi–Gauss–Lobatto collocation method [69] for Example 4.

Method	ρ	a	b	t_F	N_x	h	Tol	Maximum AE_u	CPU time in seconds
Bhrawy [69]	1	−1	1	1	15	—	—	4.56×10^{-8}	—
OBHM-SSIM						10^{-2}	10^{-14}	3.16414×10^{-14}	0.368102
Bhrawy [69]	0.25	−10	10	2	—	—	—	1.33×10^{-6}	—
OBHM-SSIM						10^{-2}	10^{-14}	1.16029×10^{-8}	0.443347

where

$$\mathbf{u} = \left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \ldots, \frac{\partial^r u}{\partial x^r}\right), \quad \mathbf{v} = \left(v, \frac{\partial v}{\partial x}, \frac{\partial^2 v}{\partial x^2}, \ldots, \frac{\partial^r v}{\partial x^r}\right), \quad \text{and} \quad \mathbf{w} = \left(w, \frac{\partial w}{\partial x}, \frac{\partial^2 w}{\partial x^2}, \ldots, \frac{\partial^r w}{\partial x^r}\right).$$

Subject to the initial conditions

$$u(0, x) = u_0(x), \quad v(0, x) = v_0(x), \quad \cdots, \quad w(0, x) = w_0(x), \tag{55}$$

and boundary conditions

$$\sigma_{11}u(t,a)+\sigma_{12}\left.\frac{\partial u}{\partial x}\right|_{(t,a)}=u_a(t),\quad \sigma_{13}u(t,b)+\sigma_{14}\left.\frac{\partial u}{\partial x}\right|_{(t,b)}=u_b(t),$$
$$\sigma_{21}v(t,a)+\sigma_{22}\left.\frac{\partial v}{\partial x}\right|_{(t,a)}=v_a(t),\quad \sigma_{23}v(t,b)+\sigma_{24}\left.\frac{\partial v}{\partial x}\right|_{(t,b)}=v_b(t),$$
$$\vdots$$
$$\sigma_{\nu 1}w(t,a)+\sigma_{\nu 2}\left.\frac{\partial w}{\partial x}\right|_{(t,a)}=w_a(t),\quad \sigma_{\nu 3}w(t,b)+\sigma_{\nu 4}\left.\frac{\partial w}{\partial x}\right|_{(t,b)}=w_b(t), \tag{56}$$

where $\sigma_{\nu 1}, \sigma_{\nu 2}, \sigma_{\nu 3}$, and $\sigma_{\nu 4}$ are constant parameters. In Equation (54), on the right-hand side, $f_1, f_2, \ldots, f_\nu$ are continuous nonlinear functions. Applying the OBHM-SSIM to Equations (54) to (56) and linearizing the system gives

$$\overset{(s+1)}{f_1}=\mathscr{L}_{1r}\frac{\partial^r \overset{(s+1)}{u}}{\partial x^r}+\mathscr{L}_{1r-1}\frac{\partial^{r-1} \overset{(s+1)}{u}}{\partial x^{r-1}}+\cdots+\mathscr{L}_{12}\frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2}+\mathscr{L}_{11}\frac{\partial \overset{(s+1)}{u}}{\partial x}+\mathscr{L}_{10}\overset{(s+1)}{u}+\mathscr{H}_1,$$
$$\overset{(s+1)}{f_2}=\mathscr{L}_{2r}\frac{\partial^r \overset{(s+1)}{v}}{\partial x^r}+\mathscr{L}_{2r-1}\frac{\partial^{r-1} \overset{(s+1)}{v}}{\partial x^{r-1}}+\cdots+\mathscr{L}_{22}\frac{\partial^2 \overset{(s+1)}{v}}{\partial x^2}+\mathscr{L}_{21}\frac{\partial \overset{(s+1)}{v}}{\partial x}+\mathscr{L}_{20}\overset{(s+1)}{v}+\mathscr{H}_2,$$
$$\vdots$$
$$\overset{(s+1)}{f_\nu}=\mathscr{L}_{\nu r}\frac{\partial^r \overset{(s+1)}{w}}{\partial x^r}+\mathscr{L}_{\nu r-1}\frac{\partial^{r-1} \overset{(s+1)}{w}}{\partial x^{r-1}}+\cdots+\mathscr{L}_{\nu 2}\frac{\partial^2 \overset{(s+1)}{w}}{\partial x^2}+\mathscr{L}_{\nu 1}\frac{\partial \overset{(s+1)}{w}}{\partial x}+\mathscr{L}_{\nu 0}\overset{(s+1)}{w}+\mathscr{H}_\nu. \tag{57}$$

Equation (57) is reduced to

$$\dot{\overset{(s+1)}{u}}_{n+p_j}=\overset{(s+1)}{f}_{1n+p_j}=L_{1n+p_j}\overset{(s+1)}{u}_{n+p_j}+H_{1n+p_j},$$
$$\dot{\overset{(s+1)}{V}}_{n+p_j}=\overset{(s+1)}{f}_{2n+p_j}=L_{2n+p_j}\overset{(s+1)}{V}_{n+p_j}+H_{2n+p_j},$$
$$\vdots$$
$$\dot{\overset{(s+1)}{W}}_{n+p_j}=\overset{(s+1)}{f}_{\nu n+p_j}=L_{\nu n+p_j}\overset{(s+1)}{W}_{n+p_j}+H_{\nu n+p_j}. \tag{58}$$

Furthermore, Equation (58) can be written in matrix form as Equation (28), which yields

$$\left[\mathbf{I}-h(A\otimes I)\mathbf{L}_{1n+p}\right]\overset{(s+1)}{\mathbf{U}}_{n+p}=\mathbf{U}_n+h(A_0\otimes I)\mathbf{F}_{1n}+h(A\otimes I)\mathbf{H}_{1n+p},$$
$$\left[\mathbf{I}-h(A\otimes I)\mathbf{L}_{2n+p}\right]\overset{(s+1)}{\mathbf{V}}_{n+p}=\mathbf{V}_n+h(A_0\otimes I)\mathbf{F}_{2n}+h(A\otimes I)\mathbf{H}_{2n+p},$$
$$\vdots$$
$$\left[\mathbf{I}-h(A\otimes I)\mathbf{L}_{\nu n+p}\right]\overset{(s+1)}{\mathbf{W}}_{n+p}=\mathbf{W}_n+h(A_0\otimes I)\mathbf{F}_{\nu n}+h(A\otimes I)\mathbf{H}_{\nu n+p}. \tag{59}$$

Simplifying Equation (59) gives

$$\boldsymbol{\Delta}_1\overset{(s+1)}{\mathbf{U}}_{n+p}=\mathbf{K}_1,\quad \boldsymbol{\Delta}_2\overset{(s+1)}{\mathbf{V}}_{n+p}=\mathbf{K}_2,\quad \cdots,\quad \boldsymbol{\Delta}_\nu\overset{(s+1)}{\mathbf{W}}_{n+p}=\mathbf{K}_\nu. \tag{60}$$

The solutions are then derived by solving the matrix systems iteratively. The approximate solutions are obtained by finding the inverses of $\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \ldots, \boldsymbol{\Delta}_\nu$ in Equation (59). Thus, we obtain

$$\overset{(s+1)}{\mathbf{U}}_{n+p}=\mathbf{inv}(\boldsymbol{\Delta}_1)\mathbf{K}_1,\quad \overset{(s+1)}{\mathbf{V}}_{n+p}=\mathbf{inv}(\boldsymbol{\Delta}_2)\mathbf{K}_2,\quad \cdots\quad ,\quad \overset{(s+1)}{\mathbf{W}}_{n+p}=\mathbf{inv}(\boldsymbol{\Delta}_\nu)\mathbf{K}_\nu. \tag{61}$$

Consider the system of equations investigated in [14, 15],

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}+2u\frac{\partial u}{\partial x}-u\frac{\partial v}{\partial x}-v\frac{\partial u}{\partial x}, \tag{62}$$

$$\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} + 2v\frac{\partial v}{\partial x} - u\frac{\partial v}{\partial x} - v\frac{\partial u}{\partial x}, \tag{63}$$

subject to the initial and boundary conditions:

$$u(0,x) = \sin(x), \quad u(t,a) = e^{-t}\sin(a), \quad u(t,b) = e^{-t}\sin(b),$$
$$v(0,x) = \sin(x), \quad v(t,a) = e^{-t}\sin(a), \quad v(t,b) = e^{-t}\sin(b).$$

The exact solutions of Equations (62) and (63) are given in [71] as

$$u(x,t) = e^{-t}\sin(x), \quad v(x,t) = e^{-t}\sin(x).$$

To apply the OBHM-SSIM to this system of two equations, we define

$$\overset{(s+1)}{f_1} = \frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2} + (2\overset{(s)}{u} - \overset{(s)}{v})\frac{\partial \overset{(s+1)}{u}}{\partial x} - \frac{\partial \overset{(s)}{v}}{\partial x}\overset{(s+1)}{u},$$
$$\overset{(s+1)}{f_2} = \frac{\partial^2 \overset{(s+1)}{v}}{\partial x^2} + (2\overset{(s)}{v} - \overset{(s)}{u})\frac{\partial \overset{(s+1)}{v}}{\partial x} - \frac{\partial \overset{(s)}{u}}{\partial x}\overset{(s+1)}{v},$$

where

$$\mathscr{L}_{12} = 1, \quad \mathscr{L}_{11} = 2\overset{(s)}{u} - \overset{(s)}{v}, \quad \mathscr{L}_{10} = -\frac{\partial \overset{(s)}{v}}{\partial x}, \quad \mathscr{H}_1 = 0,$$
$$\mathscr{L}_{22} = 1, \quad \mathscr{L}_{21} = 2\overset{(s)}{v} - \overset{(s)}{u}, \quad \mathscr{L}_{20} = -\frac{\partial \overset{(s)}{u}}{\partial x}, \quad \mathscr{H}_2 = 0.$$

The exact and numerical solutions of Equations (62) and (63) are shown in Figure 17. This demonstrates that the numerical solutions closely match the exact solutions for $N_x = 50$.

Figure 18 shows the absolute error, error norm, and iterations per block for Example 5. Figures 18a and 18b depict the absolute error for different values of N_x. Both figures show that the OBHM-SSIM method achieves a maximum absolute error of less than $\mathcal{O}(10^{-15})$ for an extended spatial domain $[-5, 5]$ and a time domain $[0, 5]$. This indicates that the method maintains high accuracy even over larger domains and longer time periods, further demonstrating its reliability and precision. Figure 18c presents the error norm, which remains below the order of $\mathcal{O}(10^{-15})$. The figure shows that the error decreases further with smaller step-sizes and larger values of N_x. Figure 18d illustrates the iterations per block required for convergence across various values of N_x. The results show that the OBHM-SSIM method converges rapidly, requiring

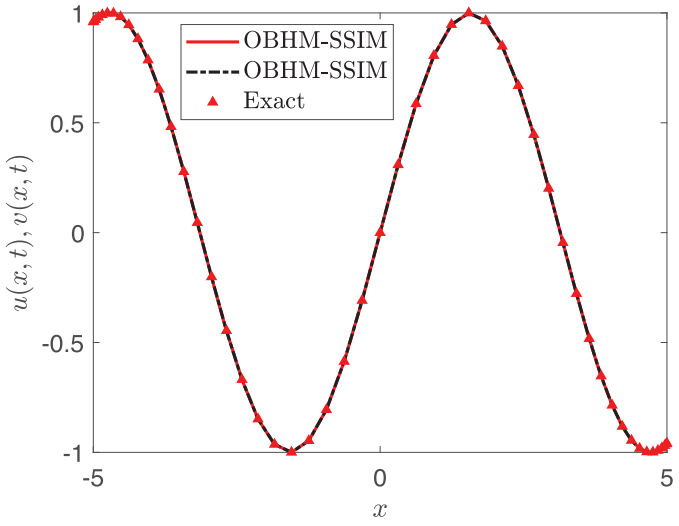

FIGURE 17 | Numerical versus exact solutions when $N_x = 50$, $Tol = 10^{-14}$, $h = 10^{-2}$, $t_F = 5$, $a = -5$, and $b = 5$ for Example 5.

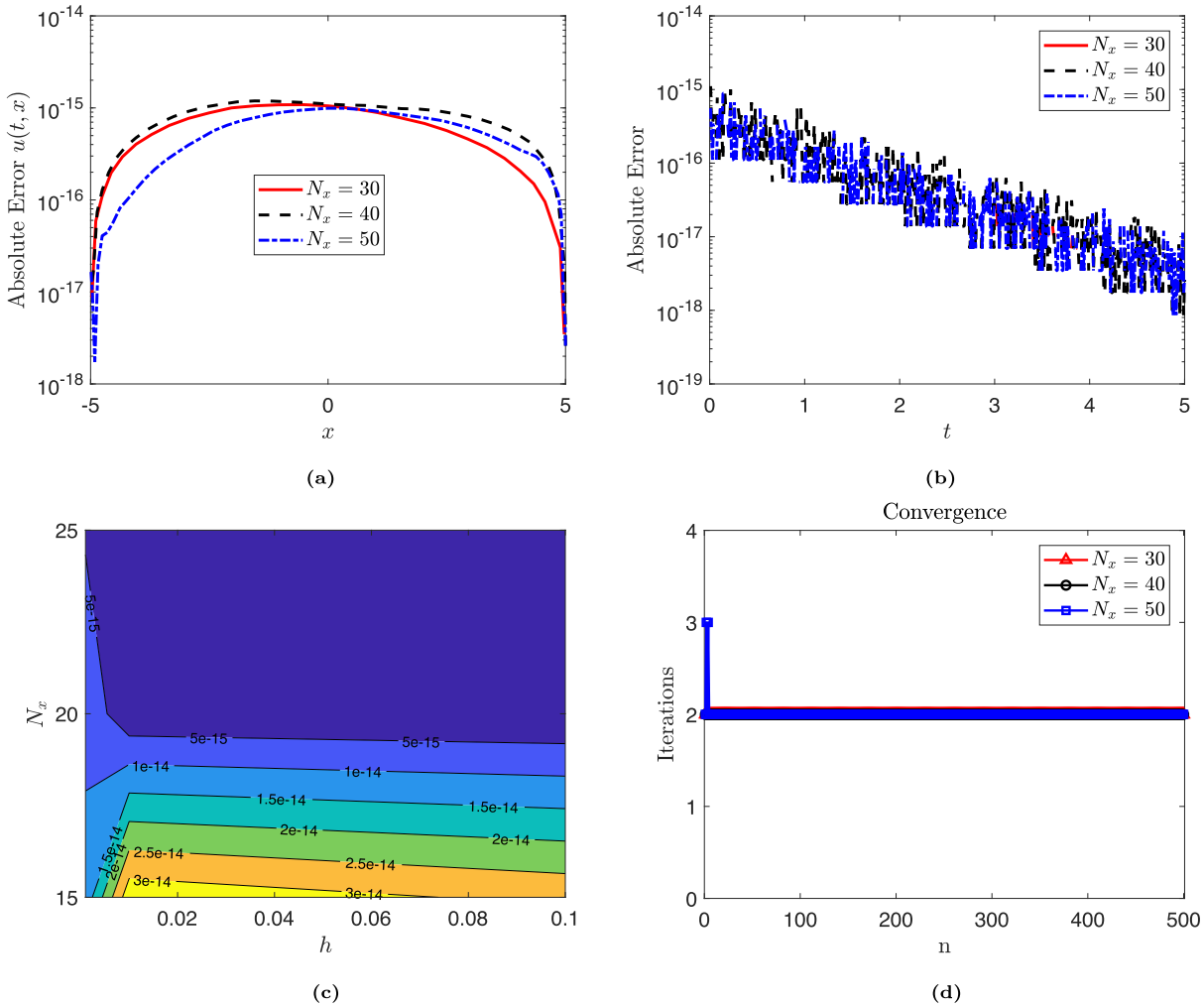

FIGURE 18 | Absolute error, error norm, and iterations per block for $Tol = 10^{-14}$, $h = 10^{-2}$, $t_F = 5$, $a = -5$, and $b = 5$ in Example 5. (a) Absolute error for various N_x. (b) Absolute error for various N_x. (c) Error norm for $a = 0$, $b = 0$, $t_F = 5$, and $Tol = 10^{-14}$. (d) Iterations per block for various N_x.

TABLE 9 | Comparison of error norms and CPU times for the OBHM-SSIM using $t_F = 1$, $h = 10^{-2}$, $N_x = 30$, and $Tol = 10^{-14}$ at selected values of t for Example 5.

t	a	b	Ahmad et al. [15]	Mittal and Arora [16]	Mohammadi and Mokhtari [17]	OBHM-SSIM
0.1	−10	10	4.81×10^{-11}	1.86×10^{-6}	4.6×10^{-6}	5.673×10^{-14}
0.5			1.49×10^{-11}	6.22×10^{-6}	4.8×10^{-6}	3.941×10^{-14}
1			2.29×10^{-8}	7.56×10^{-6}	3.7×10^{-5}	2.426×10^{-14}
CPU			—	—	243.86	0.484787

only three iterations to achieve convergence in all blocks. This rapid convergence highlights the method's computational efficiency and robust convergence properties.

Table 9 presents a comparison of the error norms for the OBHM-SSIM method against other methods [15–17]. As shown in Table 9, for all selected values of t, the OBHM-SSIM achieves an accuracy of the order $\mathcal{O}(10^{-14})$, which is superior to the methods reported in [15–17] in terms of the error norm. Furthermore, the CPU time demonstrates the computational efficiency of the OBHM-SSIM, requiring significantly less execution time compared to the method in [17].

5.6 | Example 6

To extend the OBHM-SSIM method to solve (2 + 1)-dimensional nonlinear partial differential equations, we consider the general form

$$\frac{\partial u}{\partial t} = f(\mathbf{u_2}), \quad \mathbf{u_2} = \left(u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \ldots, \frac{\partial^r u}{\partial x^r}, \frac{\partial^r u}{\partial y^r}\right), \quad a < x < b, \quad c < y < d, \quad 0 < t < t_F, \tag{64}$$

where f represents a nonlinear operator. The governing equation is subject to the initial condition

$$u(0, x, y) = u_0(x, y), \tag{65}$$

and the boundary conditions

$$\begin{aligned} \sigma_{11}u(t, a, y) + \sigma_{12}\left.\frac{\partial u}{\partial x}\right|_{(t,a,y)} &= u_a(t, y), \quad \sigma_{13}u(t, b, y) + \sigma_{14}\left.\frac{\partial u}{\partial x}\right|_{(t,b,y)} = u_b(t, y), \\ \sigma_{21}u(t, x, c) + \sigma_{22}\left.\frac{\partial u}{\partial y}\right|_{(t,x,c)} &= u_c(t, x), \quad \sigma_{23}u(t, x, d) + \sigma_{24}\left.\frac{\partial u}{\partial y}\right|_{(t,x,d)} = u_d(t, x), \end{aligned} \tag{66}$$

where $\sigma_{l1}, \sigma_{l2}, \sigma_{l3}$, and σ_{l4} are constant parameters and $l = 1, 2$. The temporal discretization remains identical to that of the (1 + 1)-dimensional case, as defined in Equations (5) to (7), while the spatial discretization is extended to a two-dimensional domain.

The spatial domains $x \in [a, b]$ and $y \in [c, d]$ are discretized into $N_x + 1$ and $N_y + 1$ Chebyshev–Gauss–Lobatto points, respectively. The collocation points are expressed as

$$\hat{x}_{i_x} = \cos\frac{\pi\, i_x}{N_x}, \quad \hat{y}_{i_y} = \cos\frac{\pi\, i_y}{N_y}, \tag{67}$$

and the corresponding physical coordinates are obtained using the transformations

$$x = \frac{(b-a)}{2}\hat{x}_{i_x} + \frac{(b+a)}{2}, \quad y = \frac{(d-c)}{2}\hat{y}_{i_y} + \frac{(d+c)}{2}. \tag{68}$$

These are extended to two-dimensional operators using the tensor product formulation $\mathcal{D}_x = D_x \otimes I_{N_y+1}$ and $\mathcal{D}_y = I_{N_x+1} \otimes D_y$. Approximating u at each time block n by a tensor-product Lagrange polynomial gives

$$u(t, x, y) \approx \sum_{j=0}^{m}\sum_{i_x=0}^{N_x}\sum_{i_y=0}^{N_y} u(t_{n+p_j}, \hat{x}_{i_x}, \hat{y}_{i_y}) L_{i_x}(x) L_{i_y}(y) L_n(t), \tag{69}$$

where $L_{i_x}(x)$ and $L_{i_y}(y)$ denote the characteristic Lagrange basis functions associated with the Chebyshev–Gauss–Lobatto points. From Equation (69), the approximate first-order derivatives are expressed as

$$\frac{\partial u}{\partial x} \approx \mathcal{D}_x U, \qquad \frac{\partial u}{\partial y} \approx \mathcal{D}_y U, \tag{70}$$

Similarly, higher-order derivatives are computed by powers of these matrices. To solve Equation (64), the (2 + 1)-dimensional nonlinear evolution equation is first linearized using the simple iteration method (SIM), resulting in the following iterative form:

$$\begin{aligned} \overset{(s+1)}{f} &= \mathscr{L}_{r_x}\overset{(s+1)}{\frac{\partial^r u}{\partial x^r}} + \mathscr{L}_{r_y}\overset{(s+1)}{\frac{\partial^r u}{\partial y^r}} + \mathscr{L}_{r-1_x}\overset{(s+1)}{\frac{\partial^{r-1} u}{\partial x^{r-1}}} + \mathscr{L}_{r-1_y}\overset{(s+1)}{\frac{\partial^{r-1} u}{\partial y^{r-1}}} + \ldots \\ &\quad + \mathscr{L}_{2_x}\overset{(s+1)}{\frac{\partial^2 u}{\partial x^2}} + \mathscr{L}_{2_y}\overset{(s+1)}{\frac{\partial^2 u}{\partial y^2}} + \mathscr{L}_{1_x}\overset{(s+1)}{\frac{\partial u}{\partial x}} + \mathscr{L}_{1_y}\overset{(s+1)}{\frac{\partial u}{\partial y}} + \mathscr{L}_{0_{xy}}\overset{(s+1)}{u} + \mathscr{H}(t, x, y), \end{aligned} \tag{71}$$

where the coefficients $\mathscr{L}_{r_x}, \mathscr{L}_{r_y}, \ldots, \mathscr{L}_{0_{xy}}$ are obtained from the previous iteration. Equation (71) can thus be written in compact form as

$$\overset{(s+1)}{\dot{u}_{n+p_j}} = \overset{(s+1)}{f_{n+p_j}} = L_{n+p_j}\overset{(s+1)}{u_{n+p_j}} + H_{n+p_j}, \tag{72}$$

Furthermore, the fully discretized OBHM-SSIM system takes the matrix form

$$\left[\mathbf{I} - h(A \otimes I)\mathbf{L}_{n+p}\right]\overset{(s+1)}{\mathbf{U}}_{n+p} = \mathbf{U}_n + h(A_0 \otimes I)\mathbf{F}_n + h(A \otimes I)\mathbf{H}_{n+p}, \tag{73}$$

where I is an identity matrix of size $(N_x+1)(N_y+1)\times(N_x+1)(N_y+1)$, $\mathbf{I}$ is of size $m(N_x+1)(N_y+1)\times m(N_x+1)(N_y+1)$. Simplifying Equation (73) yields

$$\mathbf{\Delta}\overset{(s+1)}{\mathbf{U}}_{n+p} = \mathbf{K}. \tag{74}$$

The approximate solution is then obtained from

$$\overset{(s+1)}{\mathbf{U}}_{n+p} = \mathbf{inv}(\mathbf{\Delta})\mathbf{K}. \tag{75}$$

Consider the two-dimensional nonlinear Burgers' equation

$$\frac{\partial u}{\partial t} = \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - u\frac{\partial u}{\partial x} - u\frac{\partial u}{\partial y}, \quad 0 < x < b, \quad 0 < y < d, \quad 0 < t < t_F, \tag{76}$$

where Re denotes the Reynolds number. The initial condition is given by

$$u(0, x, y) = \frac{1}{1 + e^{Re(x+y)/2}},$$

and the boundary conditions are

$$u(t, 0, y) = \frac{1}{1 + e^{Re(y-t)/2}}, \quad u(t, b, y) = \frac{1}{1 + e^{Re(2+y-t)/2}},$$
$$u(t, x, 0) = \frac{1}{1 + e^{Re(x-t)/2}}, \quad u(t, x, d) = \frac{1}{1 + e^{Re(x+2-t)/2}}.$$

The exact solution of the equation, as given in [72], is

$$u(t, x, y) = \frac{1}{1 + e^{Re(x+y-t)/2}}.$$

To implement the OBHM-SSIM, from Equation (76), we define

$$\overset{(s+1)}{f} = \frac{1}{Re}\frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2} + \frac{1}{Re}\frac{\partial^2 \overset{(s+1)}{u}}{\partial y^2} - \overset{(s)}{u}\frac{\partial \overset{(s+1)}{u}}{\partial x} - \overset{(s)}{u}\frac{\partial \overset{(s+1)}{u}}{\partial y},$$

where $\mathscr{L}_{2_x} = 1/Re$, $\mathscr{L}_{2_y} = 1/Re$, $\mathscr{L}_{1_x} = -\overset{(s)}{u}$, $\mathscr{L}_{1_y} = -\overset{(s)}{u}$, $\mathscr{L}_0 = 0$, and $\mathscr{H}_{xy} = 0$.

Figure 19 demonstrates the excellent agreement between the numerical and exact solutions for Example 5, confirming the precision, stability, and reliability of the OBHM-SSIM in solving the two-dimensional Burgers' equation.

Table 10 presents a comparison of the maximum absolute errors and computational times obtained using the proposed OBHM-SSIM and the MDMV-SQLM [73] for Example 6. The OBHM-SSIM exhibits superior accuracy, yielding smaller maximum absolute errors than the MDMV-SQLM for all considered Reynolds numbers. In addition, the computational time is substantially reduced, confirming the method's efficiency and rapid convergence. Overall, the results demonstrate the effectiveness and robustness of the OBHM-SSIM in solving nonlinear two-dimensional problems with high accuracy and reduced computational cost.

Figure 20 presents the absolute error, error norm, and iteration profiles for Example 6. As shown in Figure 20a, the absolute error distribution for $Re = 1$ at $h = 10^{-1}$ remains very small, on the order of $\mathcal{O}(10^{-15})$, confirming the high spatial accuracy of the proposed method. The temporal evolution of the error norm for $Re = 1,\ 2,$ and 3, shown in Figure 20b, decreases steadily with time to approximately $\mathcal{O}(10^{-8})$, $\mathcal{O}(10^{-9})$, and $\mathcal{O}(10^{-12})$, respectively, demonstrating the stability and convergence of the scheme. Figure 20c shows that increasing N_x and N_y while decreasing h at $Re = 1$ further improves accuracy, highlighting the method's consistency. Figure 20d indicates that the number of iterations required for convergence remains nearly constant, with a maximum of eight across all Re, confirming the robustness and computational efficiency of the OBHM-SSIM method.

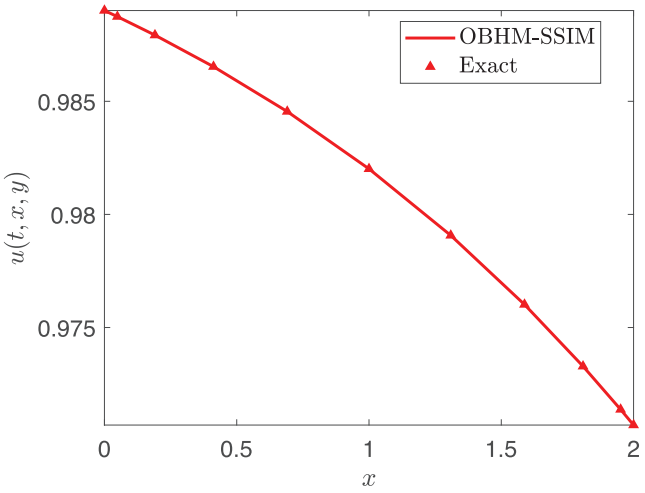

FIGURE 19 | Numerical versus exact solutions for Example 6 when $N_x = 8$, $N_y = 8$, $Tol = 10^{-12}$, $h = 10^{-1}$, $t_F = 20$, $a = 0$, $b = 2$, $c = 0$, and $d = 2$.

TABLE 10 | Comparison of maximum absolute errors and CPU times using the OBHM-SSIM and the MDMV-SQLM [73] for Example 6.

Method	Re	a	b	c	d	t_F	N_x	N_y	h	Tol	Maximum AE_u	CPU time
Mkhatshwa et al. [73]	1	0	2	0	2	10	15	15	—	—	1.421×10^{-14}	16.054362
OBHM-SSIM							9	9	10^{-1}	10^{-12}	2.708944×10^{-14}	3.662028
Mkhatshwa et al. [73]	10	0	2	0	2	16	15	15	—	—	2.032×10^{-14}	16.609866
OBHM-SSIM							9	9	10^{-1}	10^{-12}	3.663736×10^{-15}	4.012252
Mkhatshwa et al. [73]	20	0	2	0	2	20	15	15	—	—	7.327×10^{-15}	16.595988
OBHM-SSIM							9	9	10^{-1}	10^{-12}	2.331468×10^{-15}	4.539924

5.7 | Example 7

Consider the cubic Klein–Gordon equation

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - u + 2u^3, \quad a < x < b, \quad c < y < d, \quad 0 < t < t_F, \tag{77}$$

where the initial conditions are given by

$$u(0, x, y) = \operatorname{sech}(x + y), \quad \text{and} \quad u_t(0, x, y) = \operatorname{sech}(x + y)\tanh(x + y),$$

subject to the boundary conditions

$$\begin{aligned}
u(t, a, y) &= \operatorname{sech}(a + y - t), \quad & u_t(t, a, y) &= \operatorname{sech}(a + y - t)\tanh(a + y - t),\\
u(t, b, y) &= \operatorname{sech}(b + y - t), \quad & u_t(t, b, y) &= \operatorname{sech}(b + y - t)\tanh(b + y - t),\\
u(t, x, c) &= \operatorname{sech}(x + c - t), \quad & u_t(t, x, c) &= \operatorname{sech}(x + c - t)\tanh(x + c - t),\\
u(t, x, d) &= \operatorname{sech}(x + d - t), \quad & u_t(t, x, d) &= \operatorname{sech}(x + d - t)\tanh(x + d - t).
\end{aligned}$$

The exact solution of the equation, as given in [74], is

$$u(t, x, y) = \operatorname{sech}(x + y - t).$$

Letting $v = \frac{\partial u}{\partial t}$, the equation can be rewritten as the system

$$\frac{\partial u}{\partial t} = v, \quad \frac{\partial v}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - u + 2u^3. \tag{78}$$

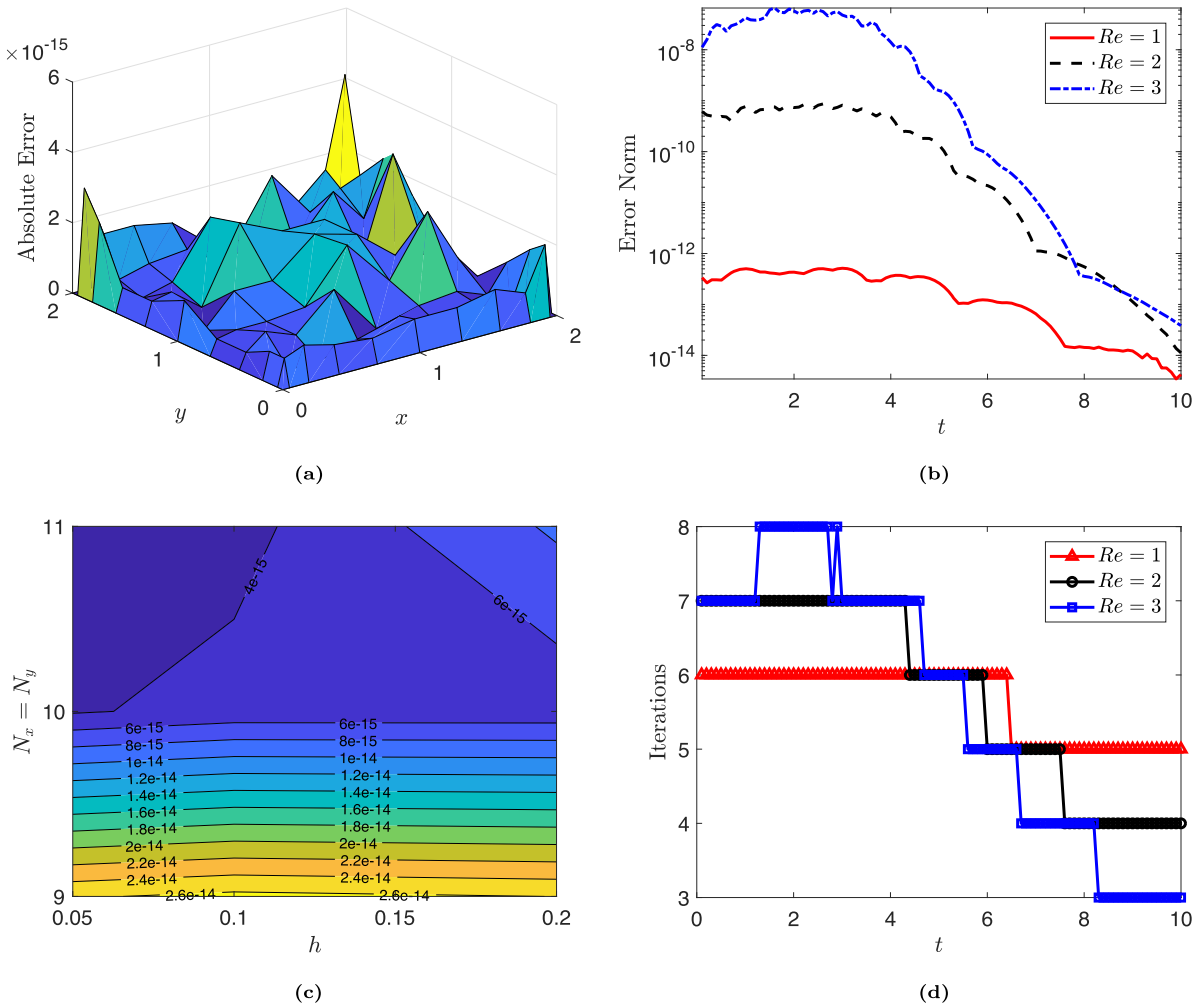

FIGURE 20 | Absolute error, error norm, and iteration for Example 6 when $Tol = 10^{-12}$, $h = 10^{-1}$, $t_F = 10$, $a = 0$, $b = 2$, $c = 0$, $d = 2$, $b = 5$, $N_x = 10$, and $N_y = 10$. (a) Absolute error for $Re = 1$ at $h = 10^{-1}$. (b) Error norm for varying Re at $h = 10^{-1}$. (c) Error norm for varying N_x, N_y, and h at $Re = 1$. (d) Iterations versus time for different values of Re.

To apply the OBHM-SSIM to this $(2+1)$-dimensional system of two equations, we define

$$\overset{(s+1)}{f_1} = \overset{(s+1)}{v},$$

$$\overset{(s+1)}{f_2} = \frac{\partial^2 \overset{(s+1)}{u}}{\partial x^2} + \frac{\partial^2 \overset{(s+1)}{u}}{\partial y^2} + \left(-1 + 2s)u^2\right)\overset{(s+1)}{u},$$

where

$$\mathscr{L}_{12_u} = 0, \quad \mathscr{L}_{12_v} = 0, \quad \mathscr{L}_{11_u} = 0, \quad \mathscr{L}_{11_v} = 0, \quad \mathscr{L}_{10_u} = 0, \quad \mathscr{L}_{10_v} = 1, \quad \mathscr{H}_{1_{xy}} = 0,$$
$$\mathscr{L}_{22_u} = 1, \quad \mathscr{L}_{22_v} = 0, \quad \mathscr{L}_{21_u} = 1, \quad \mathscr{L}_{21_v} = 0, \quad \mathscr{L}_{20_u} = -1 + 2s)u^2, \quad \mathscr{L}_{20_v} = 0, \quad \mathscr{H}_{2_{xy}} = 0.$$

Figure 21 compares the numerical and exact solutions of Example 7 for the variables u and v. As shown in Figures 21a and 21b, the numerical results obtained using the OBHM-SSIM method are in excellent agreement with the exact solutions across the entire domain, confirming the high accuracy and of the method in solving the two-dimensional cubic Klein–Gordon equation.

Figure 22 illustrates the absolute error and error norm distributions for Example 7. Figures 22a and 22b show the absolute error surfaces for $u(t, x, y)$ and $v(t, x, y)$, respectively. In both cases, the maximum absolute error remains small, of the order of $\mathcal{O}(10^{-15})$ for u and $\mathcal{O}(10^{-13})$ for v, indicating the high spatial precision of the OBHM-SSIM method. Figures 22c and

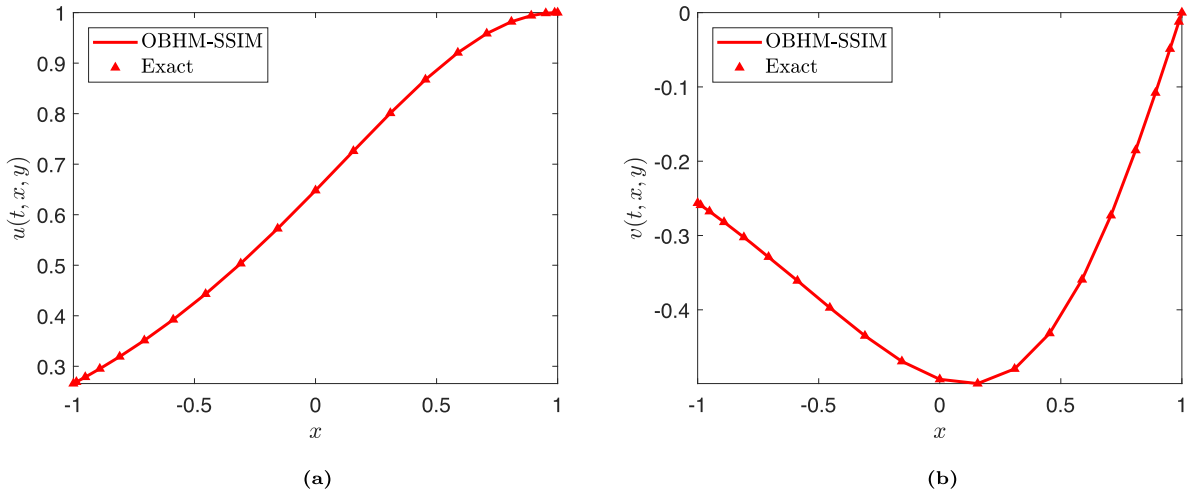

FIGURE 21 | Numerical and exact solutions for Example 7 when $a = -1$, $b = 1$, $c = -1$, $d = 1$, $Tol = 10^{-12}$, $t_F = 1$, $N_x = 20$, $N_y = 20$, and $h = 10^{-2}$. (a) u, (b) v.

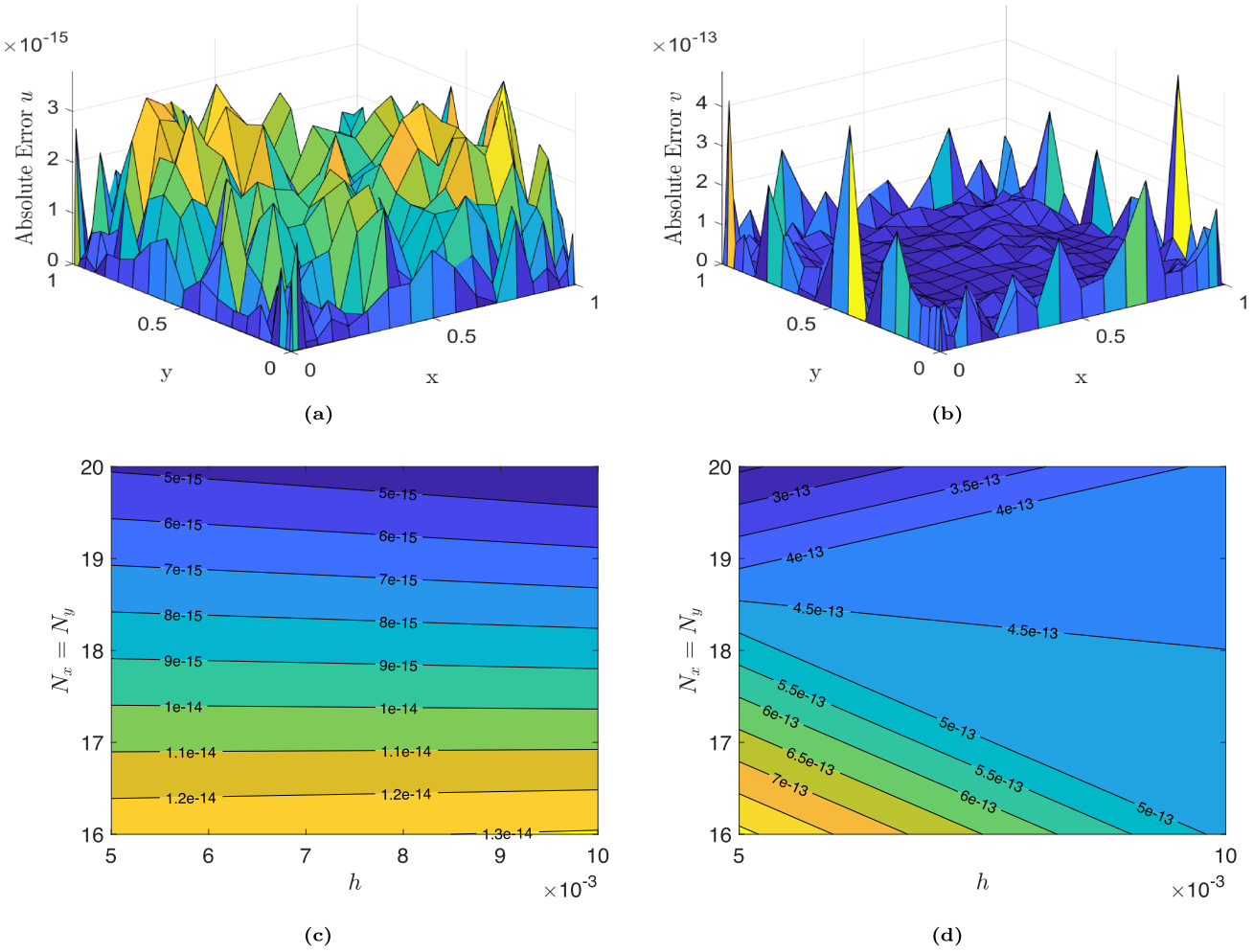

FIGURE 22 | Absolute error and error norm for Example 7 when $a = 0$, $b = 1$, $c = 0$, $d = 1$, $Tol = 10^{-12}$, $t_F = 1$, $N_x = 20$, $N_y = 20$, and $h = 10^{-2}$. (a) Absolute error for $u(t, x, y)$. (b) Absolute error for $v(t, x, y)$. (c) Error norm for $u(t, x, y)$. (d) Error norm for $v(t, x, y)$.

22d present the error norm variations with respect to $N_x = N_y$ and h, showing that increasing the collocation points N_x and N_y while using smaller step-sizes h leads to a consistent reduction in error. These results confirm the accuracy of the OBHM-SSIM when applied to the two-dimensional cubic Klein–Gordon equation.

6 | Conclusion

In this study, we presented and investigated a new optimized block hybrid spectral simple iteration method (OBHM-SSIM) for solving nonlinear evolution equations. The accuracy and efficiency of the method were evaluated through comparisons with established numerical techniques. The simple iteration scheme was employed to linearize the nonlinear terms, and the method was applied to several benchmark models, including Stokes' second problem, Burgers–Fisher, Burgers–Huxley, FitzHugh–Nagumo, and coupled Burgers' equations. The results demonstrated that the OBHM-SSIM provides highly accurate and stable solutions with fewer collocation points. The spectral collocation method ensures exponential spatial accuracy, while the optimized block hybrid approach maintains A-stability and allows for larger time domains without compromising precision. The OBHM-SSIM was further extended to two-dimensional problems, including the nonlinear Burgers' equation and the cubic Klein–Gordon equation. The numerical experiments confirmed the flexibility and robustness of the proposed method in handling multidimensional nonlinear systems with high precision and efficiency. This study contributes to the advancement of efficient high-order numerical algorithms for nonlinear dynamical systems and fluid-flow problems. The key findings of this study are summarized as follows

- ⋆ The OBHM-SSIM exhibited rapid convergence, typically requiring only a few iterations.
- ⋆ The SIM approach provided a simple and effective procedure for linearizing nonlinear equations.
- ⋆ The OBHM-SSIM method achieved higher efficiency with smaller time step-size and greater accuracy with 10–50 collocation points in large domains.

The proposed OBHM–SSIM can be extended to solve nonlinear three-dimensional partial differential equations as well as time-fractional partial differential equations. Also, future work will focus on

- ⋆ Investigating the use of different linearization methods to further improve the accuracy.
- ⋆ Employing rational optimal grid points to enhance stability, accuracy, and reduce errors [75].
- ⋆ Implementing overlapping multi-domain discretization in both spatial and temporal domains [76, 77].
- ⋆ Extending the spectral collocation method to Legendre–Gauss–Lobatto points, which have been shown to improve accuracy [27].
- ⋆ Incorporating adaptive step-size control in time and space to improve efficiency and accuracy for solutions with varying behavior across the domain [42, 78].

Nomenclature

AE	absolute error
AEE	absolute error estimate
BHM	block hybrid method
CGL	Chebyshev–Gauss–Lobatto
CPU	computational time in seconds
DQM	differential quadrature method
L_∞	error norm
LPM	linear partition method
LTE	local truncation error
ML_∞	the largest error norm
N-EPDEs	nonlinear evolution partial differential equations

N-ODEs	nonlinear ordinary differential equations
N-PDEs	nonlinear partial differential equations
OBHM	optimized block hybrid method
QLM	Quasi-linearization method
r	the order of space derivative
s	the iteration index
SIM	simple iteration method
SSIM	spectral simple iteration method
Tol	user-defined tolerance

Acknowledgments

The authors are grateful to the University of Pretoria, the University of KwaZulu–Natal, and University of Zululand. We are grateful to the editor and reviewers for their valuable suggestions, which have greatly improved this manuscript.

Disclosure

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available within the article.

References

1. S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering Governing Equations From Data by Sparse Identification of Nonlinear Dynamical Systems," *National Academy of Sciences of The United States of America* 113, no. 15 (2016): 3932–3937.

2. R. C. Hilborn, *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers* (Oxford University Press, 2000).

3. J. D. Logan, *An Introduction to Nonlinear Partial Differential Equations*, vol. 89 (John Wiley & Sons, 2008).

4. R. H. Enns and G. C. McGuire, *Nonlinear Physics With Mathematica for Scientists and Engineers* (Springer Science & Business Media, 2012).

5. B. P. Ingalls, *Mathematical Modeling in Systems Biology: An Introduction* (MIT Press, 2013).

6. C. D. Brummitt and J. Sprott, "A Search for the Simplest Chaotic Partial Differential Equation," *Physics Letters A* 373, no. 31 (2009): 2717–2721.

7. W. F. Ames, *Numerical Methods for Partial Differential Equations* (Academic Press, 2014).

8. B. Zheng, "Ordinary Differential Equation and Its Application," *Highlights in Science, Engineering and Technology* 72 (2023): 645–651.

9. A. El-Ajou, O. A. Arqub, and S. Momani, "Approximate Analytical Solution of the Nonlinear Fractional KdV–Burgers Equation: A New Iterative Algorithm," *Journal of Computational Physics* 293 (2015): 81–95.

10. E. U. Haq, T. Abbas, Q. M. U. Hassan, and B. Ahmad, "Analysis of Recent Analytical Techniques on the KdVB Equation," *Journal of Science and Arts* 21 (2021): 1143–1152.

11. C. Koroglu, "Exact and Nonstandard Finite Difference Schemes for the Generalized KdV–Burgers Equation," *Advances in Difference Equations* 2020, no. 1 (2020): 134.

12. J. F. Epperson, *An Introduction to Numerical Methods and Analysis* (John Wiley & Sons, 2013).

13. G. Meinardus, *Approximation of Functions: Theory and Numerical Methods*, vol. 13 (Springer Science & Business Media, 2012).

14. F. Shi, H. Zheng, Y. Cao, J. Li, and R. Zhao, "A Fast Numerical Method for Solving Coupled Burgers' Equations," *Numerical Methods for Partial Differential Equations* 33, no. 6 (2017): 1823–1838.

15. H. Ahmad, T. A. Khan, and C. Cesarano, "Numerical Solutions of Coupled Burgers Equations," *Axioms* 8, no. 4 (2019): 119.

16. R. Mittal and G. Arora, "Numerical Solution of the Coupled Viscous Burgers Equation," *Communications in Nonlinear Science and Numerical Simulation* 16, no. 3 (2011): 1304–1313.

17. M. Mohammadi and R. Mokhtari, "A Reproducing Kernel Method for Solving a Class of Nonlinear Systems of PDEs," *Mathematical Modelling and Analysis* 19, no. 2 (2014): 180–198.

18. S. A. Orszag, "Spectral Methods for Problems in Complex Geometrics," in *Numerical Methods for Partial Differential Equations* (Elsevier, 1979), 273–305.

19. L. N. Trefethen, *Spectral Methods in MATLAB* (SIAM, 2000).

20. J. Shen, T. Tang, and L. L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, vol. 41 (Springer Science & Business Media, 2011).

21. C. Pozrikidis, *Finite and Spectral Element Methods Using MATLAB* (Chapman Hall/CRC, 2005).

22. B. Meuris, S. Qadeer, and P. Stinis, "Machine-Learning-Based Spectral Methods for Partial Differential Equations," *Scientific Reports* 13, no. 1 (2023): 1739.

23. O. Otegbeye and S. Motsa, "A Paired Spectral-Finite Difference Approach for Solving Boundary Layer Flow Problems," *Afrika Matematika* 30 (2019): 433–458.

24. V. Mpendulo Magagula, S. S. Motsa, and P. Sibanda, "A Comparison of Bivariate Pseudospectral Methods for Nonlinear Systems of Steady Nonsimilar Boundary Layer Partial Differential Equations," *Computational and Mathematical Methods* 2, no. 6 (2020): e1125.

25. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics* (Springer Science & Business Media, 2007).

26. S. Motsa, V. Magagula, P. Sibanda, et al., "A Bivariate Chebyshev Spectral Collocation Quasilinearization Method for Nonlinear Evolution Parabolic Equations," *Scientific World Journal* 2014 (2014): 1–13.

27. V. Magagula, S. Motsa, and P. Sibanda, "A Multi-Domain Bivariate Pseudospectral Method for Evolution Equations," *International Journal of Computational Methods* 14, no. 4 (2017): 1750041.

28. F. B. Khumalo, S. S. Motsa, and V. M. Magagula, "A Bivariate Spectral Linear Partition Method for Solving Nonlinear Evolution Equations," *Mathematical Methods in the Applied Sciences* 47, no. 7 (2023): 6607–6621.

29. H. El-Hawary and S. Mahmoud, "On Some 4-Point Spline Collocation Methods for Solving Second-Order Initial Value Problems," *Applied Numerical Mathematics* 38, no. 1–2 (2001): 223–236.

30. M. Pheko, *Block-Hybrid Method for Burgers-Huxley Equation*Master's Thesis (North-West University, Mafikeng Campus, South Africa, 2023).

31. R. Abdelrahim, "Four Step Hybrid Block Method for the Direct Solution of Fourth Order Ordinary Differential Equations," *International Journal of Nonlinear Analysis and Applications* 12, no. 1 (2021): 215–229.

32. M. Farhan, Z. Omar, F. Mebarek-Oudina, et al., "Implementation of the One-Step One-Hybrid Block Method on the Nonlinear Equation of a Circular Sector Oscillator," *Computational Mathematics and Modeling* 31 (2020): 116–132.

33. L. F. Shampine and H. Watts, "Block Implicit One-Step Methods," *Mathematics of Computation* 23, no. 108 (1969): 731–740.

34. S. A. Ahmedai Abd Allah, P. Sibanda, S. P. Goqo, U. O. Rufai, H. Sithole Mthethwa, and O. A. Noreldin, "A Block Hybrid Method With Equally Spaced Grid Points for Third-Order Initial Value Problems," *AppliedMath* 4, no. 1 (2024): 320–347.

35. H. Ramos, "An Optimized Two-Step Hybrid Block Method for Solving First-Order Initial-Value Problems in ODEs," *Journal of Differential Geometry-Dynamical Systems* 19 (2017): 107–118.

36. R. Singla, G. Singh, H. Ramos, et al., "A Family Of-Stable Optimized Hybrid Block Methods for Integrating Stiff Differential Systems," *Mathematical Problems in Engineering* 2022 (2022): 1–18.

37. M. G. Orakwelu, *Generalised Implicit Block Hybrid Algorithms for Initial Value Problems*PhD thesis (University of KwaZulu-Natal, Peitermaritzburg, South Africa, 2019).

38. A. Kaur and V. Kanwar, *International Journal of Applied and Computational Mathematics*, vol. 8 (Springer, 2022), 1–19.

39. H. Ramos, A. Kaur, and V. Kanwar, "Using a Cubic B-Spline Method in Conjunction With a One-Step Optimized Hybrid Block Approach to Solve Nonlinear Partial Differential Equations," *Computational and Applied Mathematics* 41, no. 1 (2022): 34.

40. J. H. He, "A Review on Some New Recently Developed Nonlinear Analytical Techniques," *International Journal of Nonlinear Sciences and Numerical Simulation* 1, no. 1 (2000): 51–70.

41. S. S. Motsa, "A New Spectral Local Linearization Method for Nonlinear Boundary Layer Flow Problems," *Journal of Applied Mathematics* 2013, no. 1 (2013): 423628.

42. S. Ahmedai, P. Sibanda, S. Goqo, and O. Noreldin, "A Comparative Study of Linearization Techniques With Adaptive Block Hybrid Method for Solving First-Order Initial Value Problems," *International Journal of Differential Equations* 2025, no. 1 (2025): 8813408.

43. J. I. Ramos, "Linearization Methods in Classical and Quantum Mechanics," *Computer Physics Communications* 153, no. 2 (2003): 199–208.

44. J. M. Ortega, "The Newton-Kantorovich Theorem," *American Mathematical Monthly* 75, no. 6 (1968): 658–660.

45. P. D. Spanos, *Linearization Techniques for Non-Linear Dynamical Systems* (California Institute of Technology, 1977).

46. T. Barth, "Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms," in *25th AIAA Aerospace Sciences Meeting* (American Institute of Aeronautics and Astronautics (AIAA), 1987), 595.

47. H. Sithole, H. Mondal, and P. Sibanda, "Entropy Generation in a Second Grade Magnetohydrodynamic Nanofluid Flow Over a Convectively Heated Stretching Sheet With Nonlinear Thermal Radiation and Viscous Dissipation," *Results in Physics* 9 (2018): 1077–1085.

48. M. Revers, "On the Approximation of Certain Functions by Interpolating Polynomials," *Bulletin of the Australian Mathematical Society* 58, no. 3 (1998): 505–512.

49. J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed. (Dover Publications, 2001).

50. J. P. Boyd, "The Optimization of Convergence for Chebyshev Polynomial Methods in an Unbounded Domain," *Journal of Computational Physics* 45, no. 1 (1982): 43–79.

51. P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations* (Wiley, 1962).

52. J. C. Butcher, "Implicit Runge-Kutta Processes," *Mathematics of Computation* 18, no. 85 (1964): 50–64.

53. E. Areo and M. Rufai, "A New Uniform Fourth Order One-Third Step Continuous Block Method for the Direct Solutions of y= f (x, y, y)," *British Journal of Mathematics & Computer Science* 15, no. 4 (2016): 1–12.

54. E. A. Areo and M. T. Omojola, "A New One-Twelfth Step Continuous Block Method for the Solution of Modeled Problems of Ordinary Differential Equations," *American Journal of Computational Mathematics* 5, no. 4 (2015): 447–450.

55. G. Dahlquist, "Convergence and Stability in the Numerical Integration of Ordinary Differential Equations," *Mathematica Scandinavica* 4 (1956): 33–53.

56. H. A. Watts and L. Shampine, "A-Stable Block Implicit One-Step Methods," *BIT Numerical Mathematics* 12, no. 2 (1972): 252–266.

57. G. Wanner and E. Hairer, *Solving Ordinary Differential Equations II*, vol. 375 (Springer, 1996).

58. G. Wanner, E. Hairer, and S. P. Nørsett, "Order Stars and Stability Theorems," *BIT Numerical Mathematics* 18 (1978): 475–489.

59. J. Ferreira, S. Meira, M. Meneguette, Jr., and J. Nogueira, "Linear Multistep Methods and Order Stars: Some Properties," *Trends in Computational and Applied Mathematics* 9, no. 2 (2008): 233–242.

60. S. Qureshi, A. Soomro, F. S. Emmanuel, and E. Hincal, "A New Family of L-Stable Block Methods With Relative Measure of Stability," *International Journal of Applied Nonlinear Science* 3, no. 3 (2022): 197–222.

61. J. D. Lambert, *Computational Methods in Ordinary Differential Equations* (Wiley, 1973).

62. S. Hu, C. Fan, C. Chen, and D. Young, "Method of Fundamental Solutions for Stokes' First and Second Problems," *Journal of Mechanics* 21, no. 1 (2005): 25–31.

63. T. Sharma, S. Pathak, and G. Trivedi, "Comparative Study of Crank-Nicolson and Modified Crank-Nicolson Numerical Methods to Solve Linear Partial Differential Equations," *Indian Journal of Science and Technology* 17, no. 10 (2024): 924–931.

64. A. M. Wazwaz, "Analytic Study on Burgers, Fisher, Huxley Equations and Combined Forms of These Equations," *Applied Mathematics and Computation* 195, no. 2 (2008): 754–761.

65. M. J. Ablowitz and A. Zeppetella, "Explicit Solutions of Fisher's Equation for a Special Wave Speed," *Bulletin of Mathematical Biology* 41, no. 6 (1979): 835–840.

66. A. Golbabai and M. Javidi, "A Spectral Domain Decomposition Approach for the Generalized Burger s–Fisher Equation," *Chaos, Solitons and Fractals* 39, no. 1 (2009): 385–392.

67. H. N. Ismail, K. Raslan, and A. A. Abd Rabboh, "Adomian Decomposition Method for Burger's–Huxley and Burger's–Fisher Equations," *Applied Mmathematics and Ccomputation* 159, no. 1 (2004): 291–301.

68. A. R. Seadawy, S. T. Rizvi, and S. Ahmed, "Multiple Lump, Generalized Breathers, Akhmediev Breather, Manifold Periodic and Rogue Wave Solutions for Generalized Fitzhugh-Nagumo Equation: Applications in Nuclear Reactor Theory," *Chaos, Solitons and Fractals* 161 (2022): 112326.

69. A. H. Bhrawy, "A Jacobi–Gauss–Lobatto Collocation Method for Solving Generalized Fitzhugh–Nagumo Equation With Time-Dependent Coefficients," *Applied Mathematics and Computation* 222 (2013): 255–264.

70. H. Triki, A. Kara, A. Bhrawy, and A. Biswas, "Soliton Solution and Conservation Law of Gear-Grimshaw Model for Shallow Water Waves," *Acta Physica Polonica A* 125, no. 5 (2014): 1099–1106.

71. S. Bak, P. Kim, and D. Kim, "A Semi-Lagrangian Approach for Numerical Simulation of Coupled Burgers Equations," *Communications in Nonlinear Science and Numerical Simulation* 69 (2019): 31–44.

72. R. Mittal and A. Tripathi, "Numerical Solutions of Two-Dimensional Burgers Equations Using Modified bi-Cubic B-Spline Finite Elements," *Engineering Computations* 32, no. 5 (2015): 1275–1306.

73. M. Mkhatshwa, M. Khumalo, and P. Dlamini, "Multi-Domain Multivariate Spectral Collocation Method for (2+ 1) Dimensional Nonlinear Partial Differential Equations," *Partial Differential Equations in Applied Mathematics* 6 (2022): 100440.

74. P. G. Drazin and R. S. Johnson, *Solitons: An Introduction*, vol. 2 (Cambridge University Press, 1989).

75. S. Motsa, S. Ahmedai, M. Nefale, and O. Otegbeye, "A Rational Optimal Block Hybrid Method for Enhanced Accuracy in Solving Lane–Emden Equations," *Partial Differential Equations in Applied Mathematics* 12 (2024): 101003.

76. M. Mkhatshwa, S. Motsa, and P. Sibanda, "Overlapping Multi-Domain Bivariate Spectral Method for Systems of Nonlinear PDEs With Fluid Mechanics Applications," in *Advances in Fluid Dynamics: Selected Proceedings of ICAFD* (Springer Singapore, 2020), 685–699.

77. S. Motsa, "Overlapping Grid-Based Optimized Single-Step Hybrid Block Method for Solving First-Order Initial Value Problems," *Algorithms* 15, no. 11 (2022): 427.

78. Y. O. Tijani, O. Otegbeye, and S. D. Oloniiju, "Adaptive Multidomain Numerical Solution for Singularly Perturbed Fractional Differential Equation: Chebyshev Pseudospectral Method," *Journal of Nonlinear Science* 35, no. 5 (2025): 100.